



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE INGENIERÍA

**DESARROLLO DE UN SISTEMA WEB PARA
LA CONSULTA DE INFORMACIÓN DE UNA
BASE DE DATOS ESPACIAL DE
VARIABLES OCEANOGRÁFICAS**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERA EN COMPUTACIÓN
P R E S E N T A :
SYLVIA NALLELY SANCHEZ PEREZ

DIRECTOR DE TESIS:

M. en C. RANULFO RODRIGUEZ SOBREYRA

MÉXICO, D. F. SEPTIEMBRE DE 2014



AGRADECIMIENTOS

Gracias a mis padres Silvia, Rosa María y Carlos, todo lo que soy y todo lo que he logrado, ha sido por ustedes, gracias por todo el amor, el apoyo, la comprensión y la felicidad que me han brindado cada día de mi vida. Gracias por ser mi principal ejemplo y motor. Los amo con todas mis fuerzas.

A mi familia que siempre me ha apoyado, ha creído en mí y me ha brindado tanto cariño, gracias Oscar, Alma, Liliana, Lilia, Silvia, Triny, América, Gaby, Toño, Ana, Morelia, Cuauhtémoc, Claudia, Laya, Sofy, gracias por ser tan increíbles conmigo.

Gladys, gracias por todo lo que has hecho por mí, por compartir conmigo tus ganas de vivir al máximo y disfrutar cada momento, por todo lo que me has enseñado, por estar a mi lado siempre y brindarme lo mejor de ti, eres la mejor amiga y compañía que puedo tener, sabes lo mucho que te quiero, gracias también por permitirme conocer y convivir con tu familia, los aprecio mucho.

Areli, Michelle, Aldo, Jebús, gracias por todas las experiencias, alegrías, fiestas, viajes, risas y lágrimas que hemos compartido y que nos han unido como la familia que somos. Gracias a mis amigos Eder, Armando, Fer, Mariana, Yajhaira, Aída, Mireya, y todas las personas que he tenido el gusto de conocer y han aportado algo bueno a mi vida.

A mi amada Facultad de Ingeniería y a la Universidad Nacional Autónoma de México, que me ha formado como una profesionalista y como un ser humano, todo lo que aprendí, lo que viví y las personas que conocí, contribuyeron a que una de las mejores etapas de mi vida fuera en esos años de carrera.

Gracias a mi director de tesis M. en C. Ranulfo Rodríguez, por darme la oportunidad, confianza y apoyo para realizar este proyecto, gracias al Instituto de Ciencias del Mar y Limnología y las personas que me apoyaron el tiempo que estuve en el laboratorio.

A mis sinodales Ing. Betzabé Lizárraga, Ing. Jorge Rodríguez, Ing. Luciralia Hernández e Ing. Lucila Arellano, por tomarse el tiempo de leer y revisar mi trabajo, por las contribuciones que realizaron a esta tesis, por motivarme a ser mejor y enriquecerme. Gracias por su tiempo y consejos.

A los maestros que durante toda mi formación han aportado conocimiento, sabiduría, y me dejaron para siempre las ganas de querer seguir aprendiendo.

Gracias a Dios, a la Vida y al Universo que siempre me han acompañado y me han llevado por un hermoso camino.

RESUMEN

El Laboratorio de Oceanografía Física que pertenece al Instituto de Ciencias del Mar y Limnología de la UNAM, se ha dedicado a recolectar diariamente imágenes satelitales de la Temperatura de la Superficie del Mar (TSM) desde 1996, de un área de estudio limitada a la zona exclusiva de comercio de México. Cada imagen está compuesta por una cuadrícula de celdas muy pequeñas, en donde cada celda posee información de la TSM. Toda esta información se encuentra almacenada en discos en el laboratorio, y presenta diversas desventajas: en primer lugar, la cantidad de imágenes es muy grande, segundo, para visualizar esta información se necesita de software especializado y el formato de presentación no es el más óptimo y por último la información no está al alcance de las personas. Esta tesis se enfoca en desarrollar el sistema que dé solución a los problemas anteriores y aporte nuevos recursos para el beneficio de la comunidad científica y profesional.

Siguiendo el “Proceso de Desarrollo de Software con Metodología Orientada a Objetos y UML”, se construyó el sistema web, que permite a los usuarios interactuar con diferentes aplicaciones que les ayudarán a sacar el mayor provecho de la información de la TSM.

El principal subsistema es una aplicación WebMapping que está compuesta por una base de datos con extensión espacial (PostgreSQL/PostGIS) para la gestión de los datos que tienen información georreferenciada. Esta base de datos se conecta con un servidor de mapas (Geoserver) que se encarga de generar tanto las imágenes para el mapa (WMS) como los datos vectoriales (WFS) para las consultas de datos.

Este servidor de mapas pasa después por un caché de mapas (GeoWebCaché) que optimiza la visualización y manipulación del mapa a la hora de estar trabajando con él y finalmente se utiliza la API OpenLayers para acceder a la información y poder presentarla de forma atractiva y funcional, como un mapa en una página web. Este mapa cuenta con controles, que les permitirá a los usuarios interactuar con él y lo principal, consultar la temperatura de cada celda y su información asociada.

Gracias a la metodología empleada para desarrollar el sistema, se cubren todos los requerimientos solicitados, creando un sistema fácil de usar, eficiente, diseñado para seguir agregando nueva información de otras variables oceanográficas y más herramientas de análisis. La puesta en marcha de este sistema servirá a los especialistas como una herramienta de consulta y análisis, que les permitirá complementar su información para investigaciones relacionadas principalmente con los procesos Océano-Atmósfera-Continente y con el Cambio Climático Global.

ÍNDICE

Capítulo 1 Introducción	1
1.1 Introducción general	1
1.2 Objetivos.....	2
1.3 Contenido	3
Capítulo 2 Fundamentos de Cartografía	4
2.1 Introducción	4
2.2 Elementos cartográficos de la Tierra.....	5
2.2.1 Forma de la Tierra y su expresión cartográfica	5
2.2.2 Sistemas de coordenadas	6
2.2.3 Proyecciones cartográficas.....	7
2.2.4 Proyecciones cartográficas en México	9
2.2.5 La escala.....	10
2.3 Los mapas	10
2.3.1 Tipos de mapas.....	11
2.3.2 Elementos del mapa	11
2.3.3 El proceso cartográfico.....	12
2.4 Cartografía digital	13
2.4.1 La tecnología y la cartografía.....	13
2.4.2 Ventajas de los mapas digitales.....	14
2.4.3 Obtención de datos geográficos digitales	14
2.5 Sistemas de Información Geográfica (SIG)	16
2.5.1 Capas	17
2.5.2 Arquitectura de un SIG	17
Capítulo 3 Bases de Datos y Bases de Datos Espaciales	18
3.1 Introducción	18
3.2 Sistemas de Bases de Datos	19
3.2.1 Bases de Datos.....	19
3.2.2 Sistemas Manejadores de Bases de Datos	20
3.2.3 Gestión de transacciones	21
3.2.4 Arquitectura de tres niveles	21
3.2.5 Modelado de los datos.....	22

3.2.6	Lenguajes de Bases de Datos.....	24
3.2.7	Programas de aplicación e interfaces de usuario.....	24
3.3	Proceso de diseño de una Base de Datos.....	25
3.3.1	Diseño conceptual.....	26
3.3.2	Diseño lógico.....	29
3.3.3	Diseño físico.....	32
3.4	Bases de Datos Espaciales.....	32
3.4.1	Objetos geográficos.....	33
3.4.2	Representación de datos geográficos.....	34
3.4.3	Manipulación de datos geográficos.....	35
3.4.4	Modelos lógicos.....	35
Capítulo 4	Aplicaciones WebMapping.....	36
4.1	Introducción.....	36
4.2	Internet y la web.....	37
4.2.1	Internet.....	37
4.2.2	La web.....	38
4.2.3	Arquitectura cliente/servidor.....	39
4.3	Aplicaciones web.....	40
4.3.1	Entornos web.....	40
4.3.2	Ventajas y desventajas de las aplicaciones web.....	41
4.3.3	Arquitectura de las aplicaciones web.....	42
4.4	Aplicaciones WebMapping.....	43
4.4.1	Mapas web.....	43
4.4.2	WebMapping (Cartografía Web).....	45
4.4.3	Historia del WebMapping.....	45
4.4.4	Arquitectura WebMapping.....	46
4.4.5	Protocolos.....	48
4.4.6	Ventajas y desventajas WebMapping.....	48
4.4.7	Herramientas para el desarrollo de aplicaciones WebMapping.....	49
Capítulo 5	Desarrollo de Software Orientado a Objetos con UML.....	51
5.1	Introducción.....	51
5.2	Proceso de desarrollo de Software.....	52
5.2.1	Modelos del proceso de software.....	52

5.2.2	Metodologías para el desarrollo de software	53
5.2.3	Metodologías web.....	54
5.3	Desarrollo de software Orientado a Objetos	55
5.3.1	Comparación con el modelo tradicional	55
5.3.2	Ventajas del enfoque orientado a objetos.....	55
5.3.3	Características de la Orientación a Objetos	56
5.3.4	Conceptos Básicos de la Orientación a Objetos	57
5.4	Introducción a UML.....	58
5.4.1	Importancia del UML.....	59
5.4.2	Diagramas UML	59
5.4.3	Otras características del UML.....	66
5.5	Proceso de desarrollo de software con metodología orientada a objetos y UML.....	67
5.5.1	Visión del proyecto.....	68
5.5.2	Determinación de requerimientos	68
5.5.3	Análisis.....	68
5.5.4	Diseño.....	69
5.5.5	Arquitectura	69
5.5.6	Construcción (implementación).....	69
5.5.7	Pruebas.....	69
Capítulo 6	Desarrollo y Resultados	69
6.1	Desarrollo	70
6.1.1	Visión del proyecto (estudio del problema).....	70
6.1.2	Determinación de requerimientos.....	73
6.1.3	Análisis.....	81
6.1.4	Diseño.....	84
6.1.5	Arquitectura	86
6.1.6	Construcción.....	88
6.1.7	Pruebas.....	101
6.2	Resultados	102
Capítulo 7	Conclusiones	107
	Bibliografía.....	109
	Anexos	112

CAPÍTULO 1 INTRODUCCIÓN

1.1 INTRODUCCIÓN GENERAL

Las bases de datos son parte importante de la vida diaria de las personas y son, sin duda alguna una de las herramientas más poderosas e imprescindibles en casi todas las empresas de cualquier tipo. A esto, hay que sumarle la importante aparición y gran desarrollo que han tenido las bases de datos espaciales, haciendo aún más grande el abanico de soluciones que las bases de datos ya proporcionaban.

Las bases de datos espaciales contribuyen actualmente a tener una gran cantidad de información geográfica almacenada a través del uso de figuras geométricas con una ubicación espacial, que permite la representación de objetos del mundo real como calles, edificios, ríos, montañas, ciudades, con la posibilidad de ser visualizados con software especializado. También, permiten la gestión de la información y realizar consultas especializadas (análisis espacial) dedicadas a los datos geoespaciales.

Por otro lado, los mapas no sólo les han servido a los humanos para poder ubicar determinados lugares o regiones. En la actualidad los mapas pueden representar aspectos de gran importancia como los demográficos, socioeconómicos, recursos disponibles, fenómenos naturales y tanta información sea posible asociar a una ubicación geográfica.

En la actualidad la combinación de los mapas y las bases de datos espaciales constituyen una nueva rama que provee a la investigación científica una gran cantidad de ventajas. Los Sistemas de Información Geográfica (SIG) son un ejemplo de esto, sin embargo, debido a que es software que requiere de instalación y un conocimiento especializado, tienen el inconveniente de que sólo pueden ser usados por un sector reducido de la población.

Esta tecnología encontró en Internet y la web la mejor manera de promoverse y usarse, y es conocida como aplicaciones WebMapping. Éstas constituyen una nueva y mejor solución a muchos problemas actuales, una solución práctica, eficiente y de fácil acceso, a través de mapas interactivos en la web.

El desarrollo de este tipo de aplicaciones, como en general el desarrollo de cualquier software no es algo de rutina, se necesita el apoyo de otras disciplinas que se dedican a esto, que proporcionan la teoría, herramientas, metodologías y diagramas para poder desarrollar un software de calidad, eficiente, fácil de usar, altamente escalable y de fácil modificación y mantenimiento.

Esta tesis hace uso de lo anterior para dar solución al problema planteado por el Laboratorio de Oceanografía Física: un sistema web para la consulta de información de la Temperatura de la Superficie del Mar (TSM).

La propuesta para el desarrollo del sistema es la siguiente:

- a) Utilizar una metodología de desarrollo de software orientada a objetos y del Lenguaje Unificado de Modelado (UML), se desarrollará etapa por etapa, para poder completar el sistema.
- b) El problema de la consulta de información de TSM, se resolverá desarrollando una aplicación WebMapping, ya que permitirá presentar de forma más eficiente y muy gráfica la información a través de un mapa.
- c) Utilizar una base de datos con componente espacial para almacenar y manipular la información.
- d) Utilizar los conceptos de cartografía para dar una correcta representación a los mapas de la aplicación, así como todos los elementos fundamentales que todo mapa debe llevar.

1.2 OBJETIVOS

Aplicar una metodología de software para desarrollar un sistema web para la consulta de información de una base de datos espacial de variables oceanográficas.

Objetivos particulares:

- Realizar un estudio detallado de las diferentes partes que constituyen una aplicación web mapping, que abarque desde el aspecto físico hasta el lógico.
- Realizar el estudio de metodologías de desarrollo de software orientadas a objetos para poder elegir la más adecuada para llevar a cabo el proyecto.
- Desarrollar y poner en marcha un sistema web que sea capaz de cumplir con las necesidades que el Laboratorio de Oceanografía Física (LOF) expone, dándole un mayor énfasis a la aplicación web de mapas.

1.3 CONTENIDO

Para cumplir con estos objetivos se desarrollan los siguientes capítulos:

Capítulo 2: Fundamentos de Cartografía

En este capítulo se abordan los conceptos y las diferentes formas de representar a la Tierra en un mapa. Los elementos fundamentales de los mapas, la cartografía digital, y las herramientas que proporcionan en la actualidad.

Capítulo 3: Conceptos de Bases de Datos y Bases de Datos Espaciales

El capítulo define las principales características de las bases de datos y de los sistemas manejadores de bases de datos. El proceso de desarrollo de una base de datos, considerando los diseños conceptual, lógico y físico. También se incluyen los conceptos fundamentales de las bases de datos espaciales, su representación y manipulación.

Capítulo 4: Aplicaciones WebMapping

Este capítulo presenta una pequeña introducción sobre Internet y la web. Se estudia también la historia, arquitectura, funcionamiento y ventajas de las aplicaciones web, destacando las de WebMapping.

Capítulo 5: Desarrollo de Software Orientado a Objetos y UML

En este capítulo se abordan aspectos importantes de la Ingeniería de Software, sobre el proceso de desarrollo y algunas de las metodologías más importantes. Las que destacan son las orientadas a objetos, por lo cual se describen los conceptos más importantes. Se hace un breve estudio del Lenguaje Unificado de Modelado (UML) y a sus principales diagramas, que son las bases para poder llevar a cabo “El proceso de desarrollo de software con metodología orientada a objetos y UML”, que es la metodología elegida para el desarrollo del sistema.

Capítulo 6: Desarrollo y Resultados

En el último capítulo se presenta cada una de las etapas de la metodología empleada en el desarrollo del sistema para la consulta de TSM y los resultados obtenidos del sistema implementado.

Capítulo 7: Conclusiones

CAPÍTULO 2 FUNDAMENTOS DE CARTOGRAFÍA

2.1 INTRODUCCIÓN

Un mapa nos permite planear el desarrollo de una región; constituye la infraestructura de cualquier proyecto; permite mostrar gráficamente, de manera simplificada, una gran cantidad de información. Todo esto con el fin de poder realizar estudios, planeaciones, análisis, proyectos sobre determinadas regiones, que nos ayuden a dar solución a una gran variedad de problemas (Silva et al. 2010).

A lo largo de la historia de la humanidad, los mapas han sido muy importantes y han ayudado a las personas en gran medida para poder ubicar lugares, objetos y fenómenos de diversos tipos, información que es de suma importancia en sectores como la economía, el militar, la investigación y en la vida cotidiana de las personas; en la actualidad un 70% de la información que manejamos en cualquier disciplina está georreferenciada y cada día somos más conscientes de la importancia que esa componente geográfica tiene, y no sólo en el terreno científico, sino en el terreno mismo de la vida diaria (Olaya, 2011).

Trabajar con esta información georreferenciada requiere conocer una serie de conceptos previos necesarios para poder realizar correctamente todo tipo de operaciones. Estos datos tienen además una peculiaridad como datos espaciales, pues son datos que se sitúan sobre la superficie de la Tierra. Por ello, es necesario tener un conocimiento preciso de la forma de esta, para así tratar con exactitud y rigor la información con que se trabaja (Olaya, 2011).

La **geodesia** es la ciencia que se encarga del estudio de la forma de la Tierra, que es de suma importancia ya que debemos saber cómo es la Tierra para poder localizar puntos sobre su superficie (Olaya, 2011), mientras que la **cartografía** se encarga de reunir, analizar y procesar la información obtenida en las diversas regiones de la Tierra y representar éstas gráficamente a una escala reducida, cuidando que todos los elementos y detalles sean claramente visibles, facilitando la lectura e interpretación de los aspectos graficados (Caire,2002).

2.2 ELEMENTOS CARTOGRÁFICOS DE LA TIERRA

La necesidad del estudio geodésico surge por el hecho de que la Tierra no es plana, y cuando el territorio que pretendemos estudiar es lo suficientemente extenso, la curvatura de la Tierra no puede ser ignorada. Otro aspecto básico son las denominadas **proyecciones cartográficas**. Estas permiten transformar las coordenadas sobre la superficie curva de la Tierra en coordenadas sobre una superficie plana. Esto es necesario para poder representarlas en un soporte plano tal como puede ser un mapa o la pantalla de una computadora, así como para poder analizarlas de forma más simple (Olaya, 2011).

2.2.1 Forma de la Tierra y su expresión cartográfica

La Tierra no es una esfera perfecta, ya que su propia rotación ha modificado esa forma y ha provocado un achatamiento en los polos; esta forma es aproximadamente esférica, pero la verdadera forma de la Tierra es la del **Geoide**. Esta superficie ideal coincide con el nivel medio del mar, prolongado por debajo de los continentes, sin embargo esta figura irregular no es expresable matemáticamente (Silva et al. 2010).

El intento más básico de establecer un modelo de la forma de la Tierra es asimilar ésta a una figura geométrica simple, la cual pueda expresarse mediante una ecuación matemática, que permite la aplicación de conceptos geométricos, estableciendo así una base práctica para el trabajo con coordenadas y la definición de sistemas de referencia (Olaya, 2011).

Esta figura es el **elipsoide de revolución** (figura 2.1), es decir la engendrada por una elipse que gira alrededor de su eje menor (Silva et al. 2010), un elipsoide viene definido por dos parámetros: el semieje mayor y el semieje menor (Olaya, 2011).

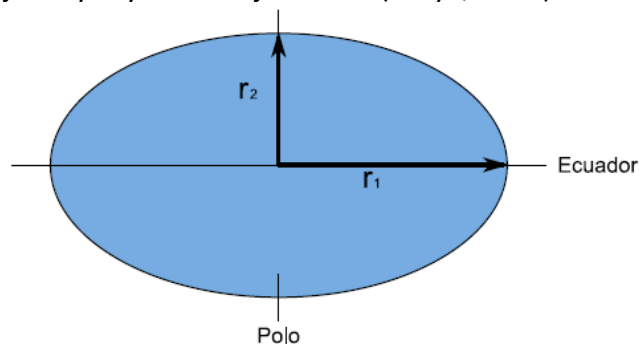


Figura 2.1 Representación de la Tierra con una figura geométrica simple, el Elipsoide de revolución, con sus parámetros, semieje mayor (r_1) y semieje menor (r_2).

Como no existe correspondencia 1:1 entre el Geoide y el elipsoide, las diferentes regiones del planeta se han referido a varios elipsoides de acuerdo al mejor ajuste con la curvatura de la Tierra en la región en cuestión y son conocidos como elipsoides de referencia (Figura 2.2), el elipsoide internacional es el de Hayford, 1910, mientras que en México se ha utilizado el elipsoide de Clarke de 1866 (Silva et al. 2010).

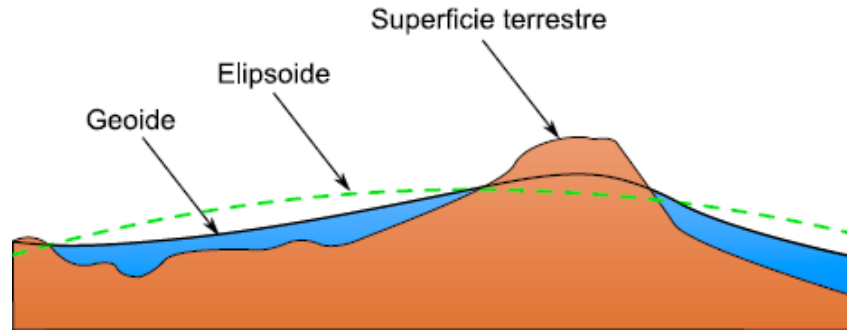


Figura 2.2 Representación del geoide y del elipsoide de referencia y los datums.

Los diferentes elipsoides se refieren a puntos en los cuales el Geoide y los elipsoides coinciden; a tales puntos se les conoce como datums y están caracterizados por el conocimiento de su posición geográfica precisa, de tal forma que sirven como origen. Junto con el elipsoide de Clarke 1866, se utilizaba el Datum Norteamericano de 1927 (Meades Ranch), mejor conocido como NAD27 (Silva et al. 2010).

A partir de que los satélites artificiales han permitido efectuar mediciones a escala mundial, ha sido preciso establecer un elipsoide globalmente válido. Los parámetros, definidos por los primeros elipsoides, no tardaron en ser ajustados para definir el elipsoide del sistema geodésico mundial **WGS 84**, que es muy empleado en la actualidad, pues es utilizado por el sistema GPS (Correia, 2000), y también se pudo observar que el elipsoide Clarke 1866 y del datum NAD 27 no eran tan exactos y se ha pasado a utilizar en México como referencia el sistema ITRF92 con el elipsoide GRS80 y el WGS84 con elipsoide del mismo nombre (Silva et al. 2010).

2.2.2 Sistemas de Coordenadas

Disponiendo de un modelo preciso para definir la forma de la Tierra, podemos establecer ya un sistema de codificar cada una de las posiciones sobre su superficie y asignar a estas las correspondientes coordenadas. Puesto que la superficie de referencia que consideramos es un elipsoide, lo más lógico es recurrir a los elementos de la geometría esférica y utilizar estos para definir el sistema de referencia. De ellos se derivan los conceptos de latitud y longitud, empleados para establecer las **coordenadas geográficas** de un punto (Olaya, 2011).

Las coordenadas geográficas consisten en dos familias de círculos imaginarios, una que pasa por el eje de rotación de la Tierra: los **meridianos** y otra normal a dicho eje: los **paralelos** (figura 2.3). La situación de un punto sobre la superficie terrestre queda determinada por la intersección de un meridiano y un paralelo constituyendo sus coordenadas geográficas la **longitud y la latitud** (Silva et al. 2010).

La latitud de un lugar se mide de 0° a 90° sexagesimales de una parte y de otra del Ecuador es norte o sur según el hemisferio donde se localice el punto. La distancia correspondiente a un grado de latitud sexagesimal varía de 110.56 kilómetros en la vecindad del Ecuador a 111.70 kilómetros en la vecindad de los polos (Caire, 2002).

La longitud se mide de 0° a 180° sexagesimales, de una y de otra parte del meridiano de origen, este u oeste, el meridiano de origen es Greenwich; La distancia correspondiente a un grado de longitud de un arco decrece de 111.3 kilómetros sobre el Ecuador a cero kilómetros en los polos (Caire, 2002).

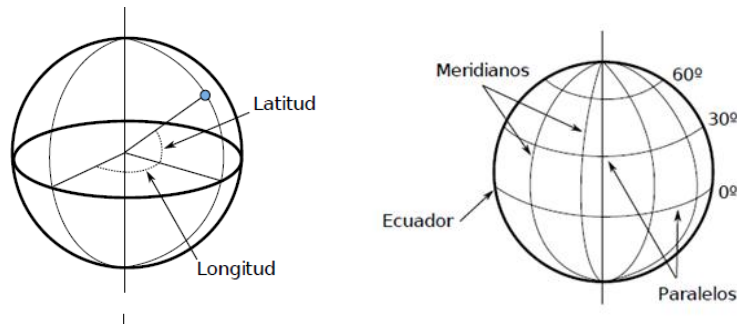


Figura 2.3 Elementos del sistema de coordenadas geográficas

Las coordenadas geográficas resultan de gran utilidad, especialmente cuando se trabaja con grandes regiones. No obstante, no se trata de un sistema cartesiano, y tareas como la medición de áreas o distancias es mucho más complicada.

2.2.3 Proyecciones Cartográficas

La geometría plana resulta mucho más intuitiva y práctica que la geometría esférica para realizar ciertas tareas, y a raíz de esto surgen las proyecciones cartográficas, que tratan de situar los elementos de la superficie del elipsoide sobre una superficie plana, y que son los que se emplean para la creación de cartografía. Al aplicar una proyección cartográfica, las coordenadas resultantes son ya **coordenadas cartesianas**.

Estamos más acostumbrados a la utilización de sistemas cartesianos en los cuales la posición de un punto se define mediante un par de medidas de distancia x e y . Por otro lado, si necesitamos crear una representación visual de la información cartográfica, lo habitual es hacerlo en una superficie plana, ya sea a la manera clásica en un pliego de papel o, usando las tecnologías actuales, en un dispositivo tal como una pantalla (Olaya, 2011). El proceso de asignar una coordenada plana a cada punto de la superficie de la Tierra (que no es plana) se conoce como proyección cartográfica (Olaya, 2011).

Es muy fácil trazar sobre una esfera un sistema de paralelos y meridianos, pero su representación en un plano requiere un estudio especial, ya que la superficie esférica no puede desarrollarse sobre un plano sin que se deforme o se rompa.

El método más sencillo para resolver este problema, consiste en rodear la esfera con un cilindro, o con un cono, o en colocar la esfera tangencialmente a un plano y proyectar una parte de la red de meridianos y paralelos desde el centro de la esfera (Figura 2.4). Cortando después el cilindro o el cono a lo largo de una generatriz y extendiéndolo sobre un plano se tiene un sistema de meridianos y paralelos resultado de una verdadera proyección. Así pues, podemos definir una proyección diciendo que es un sistema plano de meridianos y paralelos sobre el cual puede dibujarse un mapa (Raisz, 1985).

Las principales proyecciones cartográficas son las cilíndricas, las planas o azimutales y las cónicas y dependiendo de sus características, representan de mejor manera ciertas regiones de la superficie de la Tierra.

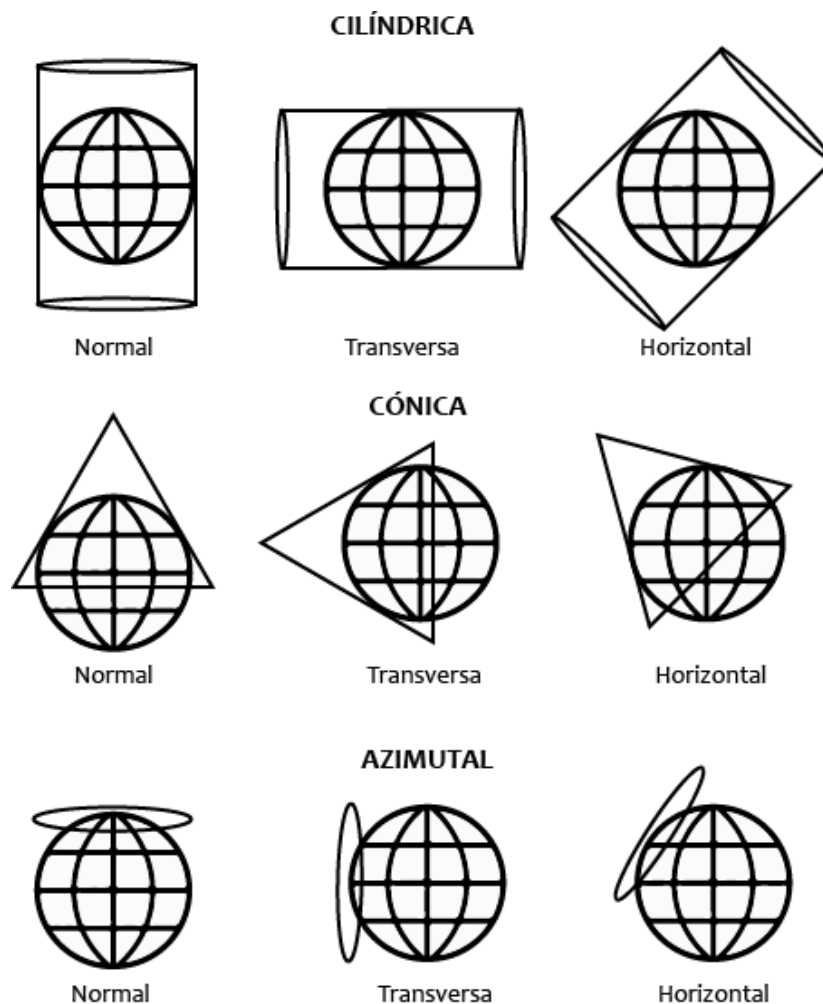


Figura 2.4 Proyecciones cartográficas tangentes, que se generan con las tres figuras geométricas, el cilindro, el cono y el plano, y de la posición en la que se encuentran.

2.2.4 Proyecciones Cartográficas en México

En México se utiliza la Proyección Cónica de Lambert en mapas a escalas igual y menores a 1:1000000. En tanto que, la cartografía regional y de detalle se realiza por lo general con base en la Proyección Universal de Mercator (UTM) (Silva, 2010).

La proyección **Cónica Conforme De Lambert** tiene la característica de ser ortofórmica. La condición de ortomorfismo establece la igualdad de las formas entre pequeñas extensiones de la Tierra y sus representaciones en el mapa. Esto se consigue haciendo que los meridianos y paralelos en la gradícula se corten a 90° y que los factores de escala en dos direcciones cualesquiera, trazadas desde un punto de vista sean iguales (Silva et al. 2010).

En la actualidad, una de las proyecciones más extendidas en todos los ámbitos es la proyección **Universal Transversa de Mercator**, la cual da lugar al sistema de coordenadas UTM. Este sistema, no es simplemente una proyección sino que se trata de un sistema completo para cartografiar la totalidad de la Tierra.

Utiliza un cilindro en posición transversa (el eje del cilindro se encuentra contenido en el plano del Ecuador) como superficie de proyección (figura 2.5), en condición secante con secciones elípticas, es conforme (ortomórfica) conserva los ángulos y casi no distorsiona las formas.

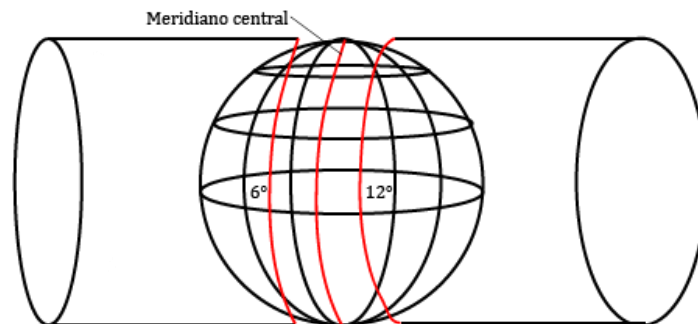


Figura 2.5 Cilindro en posición transversa de la UTM

Esta proyección divide a la Tierra en una serie de zonas rectangulares mediante una cuadrícula y se aplica una proyección y unos parámetros geodésicos concretos a cada una de dichas zonas (en la actualidad se emplea un único elipsoide WGS-84).

La cuadrícula UTM tiene un total de 60 husos numerados entre el 1 y 60, cada uno de los cuales abarca una amplitud de 6° de longitud. El huso 1 se sitúa entre los 180° y 174° O, y la numeración avanza hacia el Este. En la latitud, cada huso se divide en 20 zonas, que van desde los 80° S hasta los 84° N. Estas se codifican con las letras desde la C a la X, no utilizándose las letras I y O. Cada zona abarca 8 grados de longitud, excepto la X que se prolonga unos 4 grados adicionales (Olaya, 2011).

De esta manera a la República Mexicana le corresponden los husos 11, 12, 13, 14, 15 y 16 y las fajas P, Q, R y S.

2.2.5 La escala

Con los elementos de la geodesia y las proyecciones cartográficas ya se puede elaborar cartografía y trabajar con información georreferenciada. No obstante, existen ciertos conceptos relativos a esa cartografía que resultan de suma importancia y deben conocerse antes de abordar esas tareas. El más importante de ellos es la escala (Olaya, 2011).

El concepto de escala es fundamental a la hora de trabajar con cartografía, y es uno de los valores básicos que definen toda representación cartográfica (Olaya, 2011), está directamente relacionada con el contenido, propósito, objetivos, dimensiones y precisión del mapa. La escala caracteriza al mapa, y su correcta elección es determinante para representar con éxito la información deseada (Caire, 2002).

Se denomina escala a la relación entre la magnitud dibujada y la real; es la relación de proporción o similitud entre el mapa y el terreno. Se expresa adimensionalmente de tal modo que una unidad en el mapa corresponde con “n” unidades en el terreno (Silva et al. 2010).

Existen diversas formas en las que se puede representar a la escala: declarada (se representa a la escala mediante una frase en la que se mezclan unidades: *Un centímetro es igual a dos kilómetros*), numérica (puede expresarse mediante una fracción adimensional en la cual el numerador es la unidad, mientras que en el denominador aparece el número de unidades representadas, o también se expresa como una razón o proporción: *1:10,000*) y gráfica (se expresa como un segmento de recta subdividido de acuerdo a las unidades dibujadas y con los valores correspondientes a las magnitudes reales (figura 2.6)).

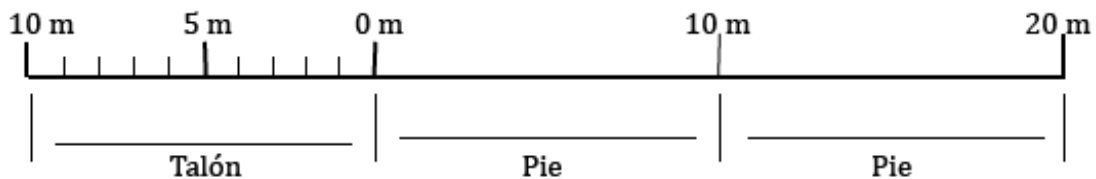


Figura 2.6 Presentación de la escala gráfica en los mapas.

2.3 LOS MAPAS

Según la Asociación Cartográfica Internacional un mapa es “la representación convencional gráfica de fenómenos concretos o abstractos, localizados en la Tierra o en cualquier parte del Universo” y de acuerdo a la S.P.P (1979), los mapas deben tener las siguientes cualidades:

- ✓ *Exactos*: La reproducción de los rasgos o elementos debe corresponder con la realidad con precisión de acuerdo a la escala de representación.
- ✓ *Completos*: La información debe ser la necesaria y suficiente.
- ✓ *Adecuados a su propósito*: Su contenido debe estar encaminado a satisfacer las necesidades de información que los motivó.
- ✓ *Claros*: Deben expresar de manera unívoca los rasgos que se pretenden representar.
- ✓ *Legibles*: La disposición de los elementos gráficos debe ser la adecuada y la impresión limpia y nítida.
- ✓ *Estéticos*: La distribución de la simbología y de los nombres debe ser armónica, deben tener una discreta graduación del objeto por describir y buen gusto en el uso de los colores.

2.3.1 Tipos de mapas

Los mapas se pueden dividir en dos grupos dependiendo de sus características, el primero es de acuerdo a *su escala de trabajo* y el segundo de acuerdo *al propósito para el cual han sido creados*.

En el primero grupo están los mapas de pequeña escala que son aquellos que representan grandes porciones de superficie terrestre y se debe de tener muy en cuenta la forma esférica de la Tierra. Por otro lado están los mapas de gran escala, que por el contrario representan regiones más pequeñas y representan con más detalle sus elementos.

En el segundo grupo encontramos a los mapas de referencia y los mapas temáticos, los primeros, también conocidos como de propósito general, se encargan de representar los principales elementos que hay en la superficie terrestre, como hidrografía, relieves, vías de comunicación, entidades de población, entre otras, mientras que los mapas temáticos se enfocan a representar determinados fenómenos o características asociados a ciertas regiones o a toda la Tierra, y por lo tanto puede abarcar desde aspectos políticos, sociales, históricos, hasta fenómenos climáticos, geológicos, entre otros.

2.3.2 Elementos del mapa

Los mapas deben contar con al menos dos elementos, que son indispensables para la correcta interpretación y lectura de los mismos, como lo son la escala y la leyenda; con la escala como ya se mencionó se puede saber la relación que existe entre lo dibujado en el mapa y la realidad, mientras que la leyenda, se encarga de explicar toda la simbología presente en el mapa.

El arreglo de las partes de la leyenda debe hacerse procurando agrupar simbologías afines (vías de comunicación, poblaciones, tipos de roca, climas, etc.) (Silva, 2010).

Sería deseable que en el mapa también se incluyan los siguientes elementos:

- *Norte*: indicar hacia donde se encuentra el polo norte.
- *Sistema de Coordenadas*: se muestran las coordenadas geográficas y el sistema de proyección que se utiliza.
- *Marco*: Delimita el área geográfica que se está representando en el mapa.
- *Título*: Es la presentación del mapa, debe proporcionar información clara y precisa acerca del contenido del mapa y/o región geográfica representada.

(Silva et al. 2010).

2.3.3 El proceso cartográfico

La concepción del mapa: Es la primera visión del mapa, dejar en claro cuál es su propósito para el que será creado, en base a eso, concebirlo, hacer una abstracción de la realidad. Partiendo de este punto se deberá hacer un primer análisis para poder elegir el tipo de mapa y su tamaño, el sistema de proyección y la escala.

La preparación del mapa: los mapas representan muchísimos fenómenos de todo tipo que puedan estar asociados a una ubicación geográfica y es necesario representar esta información de manera clara, correcta y que vaya acorde con el mapa, se necesita saber con qué tipos de datos se está trabajando y si son distribuciones continuas (se presentan en todos los puntos de la superficie) o discretas (puntos específicos).

La recopilación de datos: Ya que los mapas son una poderosa herramienta que es capaz de mostrarnos mucha información, es necesario hacer la elección y recolección de dicha información específica para cada mapa.

El diseño del mapa: Esta etapa podría ser considerada como la parte de diseño gráfico, aquí es donde se decide que símbolos se utilizarán para representar a la información y todas las indicaciones necesarias contenidas en el mapa, también se eligen los colores que se utilizarán para las regiones y símbolos, (Silva et al. 2010).

Todas estas características se tienen que visualizar en conjunto para poder lograr una armonía y una clara representación del objetivo para el cual el mapa fue hecho, para que el usuario quien consultará el mapa, pueda hacerlo de una forma agradable y eficiente.

2.4 CARTOGRAFÍA DIGITAL

Desde la mitad del siglo XX, la ciencia geográfica logra sobrepasar el enfoque descriptivo e histórico con el que se le había identificado, y busca nuevas opciones para llevar a cabo sus estudios; la cartografía y la información geográfica han sido fuertemente impactadas por el desarrollo de la informática y por el entorno digital consecuente (Reyes, 2003).

Con más y más soluciones a los problemas geográficos diarios, solucionados por procedimientos algorítmicos (las rutas más cortas, combinadas con redes de calles digitales almacenados en automóviles, son un ejemplo actualmente aplicable), las personas ejecutan estas tareas más rápidamente. Así la cartografía moderna requiere tanto de la parte artística para la ilustración geográfica, como de la analítica para resolver los problemas geográficos (Raisz, 1985).

2.4.1 La tecnología y la cartografía

El avance científico y el desarrollo tecnológico alcanzado en el siglo pasado, sobre todo en las tres últimas décadas, ha modificado la forma tradicional de abordar y realizar las actividades humanas. Esta revolución tecnológica que dio paso a la era de la computación, trajo consigo la rápida evolución de la informática. Con ello se lograron reducir los tiempos para procesar, archivar y recuperar grandes volúmenes de datos, así como el estudio y manipulación de situaciones hipotéticas que, sin el uso de las computadoras, serían muy difíciles de efectuar (Reyes, 2003).

Así, gradualmente, en las últimas décadas del siglo pasado, se comenzaron a utilizar las nuevas tecnologías para generar **información geográfica**. Entre estas tecnologías destacan la teledetección, el sistema de posicionamiento global (GPS) y los sistemas de información geográfica (Reyes, 2003).

Actualmente estamos presenciando la consolidación del cambio de paradigma de la producción de cartografía hacia la generación de **datos geográficos digitales (datos espaciales)**, dado fundamentalmente por el desarrollo de las Tecnologías de la Información (hardware, software, bases de datos, redes), el avance de la percepción remota y el sistema de posicionamiento global. Esta transformación conceptual y operacional ha sido apoyada y exigida por el diseño, formación y expansión de la autopista de la información: Internet. Con la red mundial y los servicios que ofrece la producción de datos geográficos digitales y su integración a bases de datos se engancha a modelos de acceso (consulta y explotación) compartición, distribución y comercialización de información geográfica impensables años atrás (Reyes, 2003).

2.4.2 Ventajas de los mapas digitales

La evolución tecnológica ha permitido alcanzar mayores precisiones en la recolección de los datos, agilizar su captura y acelerar los procesos de ajuste y tratamiento de la información (Reyes, 2003).

Los mapas digitales tienen una serie de ventajas frente a los analógicos, por ejemplo, la cartografía digital es editable, y esto simplifica enormemente la introducción de cambios, esta edición y actualización de datos pueden hacerla varias personas de modo concurrente. En un mapa analógico habría que rehacer todo el mapa y volver a imprimirse. Por otro lado, resulta más sencillo y menos costoso distribuir cartografía digital que analógica, ya que esto se puede hacer rápidamente por Internet.

Otra ventaja es en el campo del análisis, se pueden hacer con los datos geográficos digitales cosas que no eran posibles con los analógicos y se pueden automatizar estos análisis.

Asimismo, la precisión es mayor, ya que depende únicamente de los datos y la precisión exclusiva de estos, y no interviene el hombre en ellas, por lo que son representaciones precisas e imparciales y hasta se capturan detalles que tal vez una persona podría llegar a omitir (Olaya, 2011).

La facilidad de mantenimiento es otra ventaja, ya que aunque no se introduzcan modificaciones y no se actualicen los datos, el formato digital hace más fácil su conservación. La degradación del soporte no degrada directamente el dato en sí, haciéndolo perder su calidad. La degradación del soporte analógico (el papel), sí que lo hace. Además los datos digitales pueden replicarse con suma facilidad, por lo que su persistencia está garantizada en mayor medida y a un menor coste (Olaya, 2011).

Los mapas web pueden ofrecer fácilmente la información actualizada. Si los mapas se generan automáticamente a partir de bases de datos, que pueden mostrar la información en tiempo real (Neumann, 2008).

Anteriormente, la cartografía era restringida a unas pocas empresas, institutos y organismos cartográficos, requiriendo hardware y software caro y complejo, así como cartógrafos e ingenieros especializados para su manejo (Neumann, 2008), con la llegada de los mapas digitales, estos obstáculos se han superado.

2.4.3 Obtención de datos geográficos digitales

Gran parte de los datos geográficos que se producen actualmente son en formato digital. Otros, a pesar de producirse hoy en día, no lo son directamente, y junto a estos están, todos los datos (que no son pocos) generados con anterioridad y que se presentan en diversas formas, principalmente mapas de papel.

La teledetección es el estudio y medida de las características de una serie de objetos (en nuestro caso elementos de la superficie terrestre) sin que exista contacto físico. Para ello, se miden las perturbaciones que el objeto provoca en su entorno, principalmente las de tipo electromagnético. La fotografía aérea y las imágenes satelitales son las técnicas de teledetección más empleadas a la hora de obtener datos geográficos (Olaya, 2011).

Los satélites vinieron a darle una nueva cara a la cartografía, proporcionando muchos más elementos de los que se podían observar a simple vista o incluso con las mismas fotografías aéreas. Estas imágenes se obtienen desde algunos satélites que se dedican a cumplir esta función y que orbitan nuestra Tierra. Éstos cuentan con sensores que registran la *radiación electromagnética* (energía) que es reflejada de la superficie de la Tierra y cada valor medido toma un valor digital, esta información es la que es devuelta a estaciones en nuestro planeta y es procesada con computadoras que convierten estos números en colores entregando como resultado final una imagen de la superficie de la Tierra (figura 2.7).

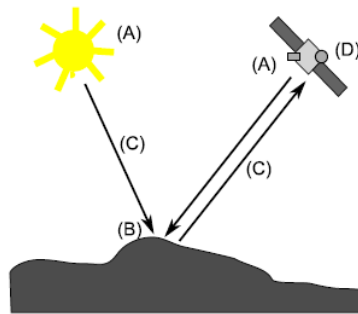


Figura 2.7 Proceso para la obtención de imágenes satelitales

Gracias al tipo de información que capturan los sensores de los satélites, las imágenes satelitales aparte de mostrarnos la superficie de la Tierra, nos da información sobre fenómenos como el clima, la temperatura de la superficie del mar, deforestación, corrientes globales, entre otros; esta información al ser transmitida en tiempo real, nos permiten observar la evolución de fenómenos naturales. Entre los sistemas de teledetección principales encontramos al LANDSAT, NOAA, IKONOS, SPOT, QuickBird, Aqua y Terra (Olaya, 2011).

La serie de satélites NOAA (National Oceanic and Atmospheric Administration) que se han lanzado desde 1960, son satélites que proveen de una gran cantidad de información de la superficie de la Tierra y son estos satélites de los cuales nosotros obtenemos la información de la Temperatura de la Superficie del Mar (TSM) (figura 2.8).

Estas imágenes satelitales, a veces necesitarán ser procesadas para que queden adecuadas al propósito para el que serán ocupadas, esto se realiza de igual forma con una computadora, a este proceso se le conoce como **procesamiento digital de imágenes**, el cual

mediante software especializado que utiliza procesos y algoritmos avanzados, es capaz de alterar una imagen, ya sea modificando el color, aplicando filtros, efectos, algoritmos.

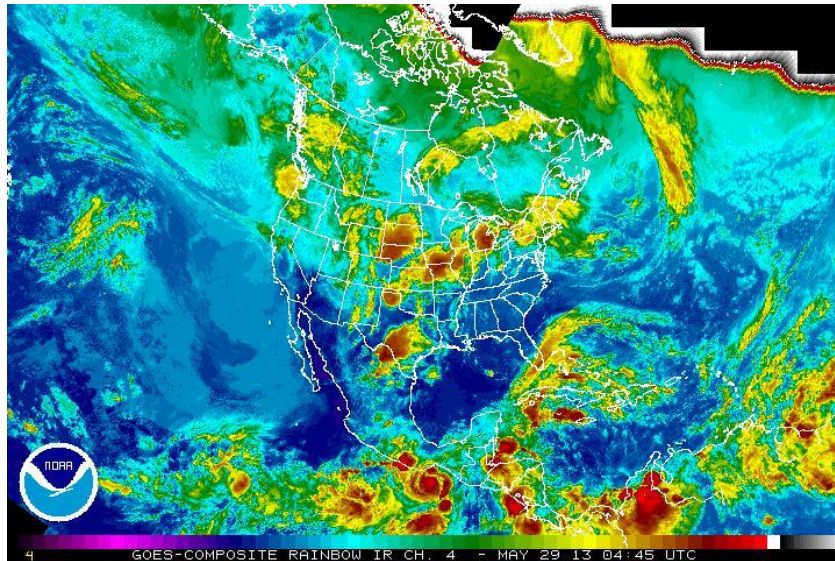


Figura 2.8 Imagen satelital obtenida con los satélites NOAA.

2.5 SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Un SIG, se puede entender como un sistema computarizado compuesto por hardware, software, datos y aplicaciones que es usado para registrar digitalmente, editar, modelizar y analizar datos geoespaciales, y presentarlos en forma alfanumérica y gráfica (Hewlett Packard, 1993). Los SIG se basan en los avances realizados en campos como la informática, las bases de datos, la estadística y la inteligencia artificial (Elmasri & Navathe, 2007).

Existen muchas definiciones de SIG y la mayoría serán muy acertadas, pero como se expone en el libro Geographic Information Systems and Science (Paul A. Longley et. Al), existirán muchas definiciones dependiendo del objetivo al que se esté mirando.

Básicamente, un SIG ha de permitir la realización de las siguientes operaciones:

- Lectura, edición, almacenamiento y, en términos generales, gestión de datos geoespaciales.
- Análisis de dichos datos. Esto puede incluir desde consultas sencillas a la elaboración de complejos modelos, y puede llevarse a cabo tanto sobre la componente espacial de los datos (la localización de cada valor o elemento) como sobre la componente temática (el valor o el elemento en sí).
- Generación de resultados tales como mapas, informes, gráficos, etc.

2.5.1 Capas

Uno de los grandes éxitos de los SIG es su estructura de manejo de información geográfica, que facilita las operaciones que se llevan a cabo con esta. El concepto de capa, es una de las grandes virtudes inherentes a los SIG, en cuanto que favorece la correcta estructuración de la información y el trabajo con ella. La capa permite dividir la información espacial referida a una zona de estudio en varios niveles, de tal forma que, pese a coincidir sobre una misma zona, información sobre distintas variables se encuentra recogida de manera independiente (figura 2.9), lo que permite trabajar con varios mapas simultáneamente, y combinar estos para la realización de operaciones en las que intervengan todos ellos.

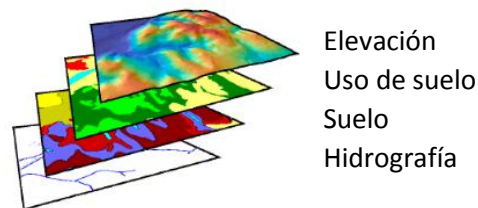


Figura 2.9 Ejemplos de diferentes capas para una misma zona de estudio

2.5.2 Arquitectura de un SIG

Anteriormente se habló de los SIG como un sistema, en el cual intervienen diferentes componentes, uno de ellos es el software como tal, este programa está compuesto por la siguiente arquitectura:

- *Interfaz de usuario (graphical user interface, GUI)*, mediante ésta, es posible la interacción entre el usuario y el GIS, ya que muestra gráficamente todas las herramientas, barras, menús y los controles que permiten la manipulación de los datos geoespaciales.
- *Herramientas*, son las que dan la funcionalidad al SIG, son las que permiten la visualización, el análisis, edición y transformación de los datos geoespaciales. Las herramientas más comunes en todos los SIG son el zoom, visualización de diferentes capas, operaciones entre capas, consulta de datos alfanuméricos, impresión de mapas y exportación de éstos a diferentes formatos.
- *Sistema manejador de datos*, finalmente están los datos geoespaciales, los mapas que son el objeto de estudio, están almacenados generalmente en bases de datos, que son controladas por el Sistema Manejador de Bases de Datos (SMBD). Los SMBD son los encargados de gestionar la información e interactuar con el SIG, para poder responder a todas las consultas que hagan los usuarios.

Todos estos elementos trabajan en conjunto para hacer funcional a un SIG, para darle el poder que realmente tiene, un sistema, una herramienta que nos permite afrontar y resolver muchos problemas (Longley, et al, 2005).

CAPÍTULO 3 BASES DE DATOS Y BASES DE DATOS ESPACIALES

3.1 INTRODUCCIÓN

Uno de los pilares de cualquier organización es la información que necesita para su funcionamiento; asimismo, una de sus actividades principales es su procesamiento, que tiene como objetivo proporcionar a las personas autorizadas la información que necesitan en el momento y el lugar adecuados. Por ello, uno de los componentes básicos de cualquier organización es su sistema de información.

Estos sistemas son un conjunto de elementos ordenadamente relacionados entre sí de acuerdo a ciertas reglas, que aportan a la organización a la que sirven la información necesaria para el cumplimiento de sus fines (Celma et al. 2003).

El mejor ejemplo de estos sistemas de información son las **bases de datos**, y son un componente esencial de la vida cotidiana en la sociedad moderna. Actualmente, la mayoría de nosotros nos enfrentamos a diversas actividades que implican cierta interacción con una base de datos, por ejemplo, ir al banco a depositar fondos, realizar una reservación en un hotel, acceder al catálogo computarizado de una biblioteca para buscar un libro, son actividades que implican que alguien o algún programa de computadora acceda a una base de datos (Elmasri & Navathe, 2007).

Sin embargo existe otro tipo de información que en la actualidad es muy importante, los datos espaciales. Los datos espaciales representan información sobre la ubicación física y la forma de objetos geométricos. Estos objetos pueden representar países, carreteras o lagos.

Una **base de datos espacial** es capaz de modelar, almacenar y consultar datos espaciales, de la misma forma en que puede hacerlo con datos no espaciales (alfanuméricos). Este tipo de sistemas permite la representación de objetos geométricos ubicados en el espacio, y sus aplicaciones van desde modelar ciudades, bosques, ríos, modelar el uso de la tierra o la división política de un país, hasta la representación de fenómenos naturales (Dunning, 2013).

3.2 SISTEMAS DE BASES DE DATOS

Un sistema de bases de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos (Silberschatz et al, 2002), hardware y usuarios (Date, 2001).

- La colección de datos, normalmente denominada *base de datos*, contiene información relevante para una empresa (Silberschatz et al, 2002).
- El Sistema Manejador de Base de Datos (SMBD) es una colección de programas que permite a los usuarios crear y mantener la base de datos (Elmasri & Navathe, 2007).
- Los componentes de hardware del sistema constan de los volúmenes de almacenamiento secundario que se emplean para contener los datos almacenados, junto con los dispositivos asociados de E/S, los controladores de dispositivos y los procesadores de hardware y la memoria principal asociada, para apoyar la ejecución del software del sistema de base de datos (Date, 2001).
- Las personas que trabajan con una base de datos se pueden catalogar como usuarios de bases de datos o como administradores de base de datos (Silberschatz et al, 2002).

Los sistemas de base datos se diseñan para gestionar grandes cantidades de información; la gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información en la base de datos de manera que sea tanto práctica como eficiente (Silberschatz et al, 2002).

Esto sistemas permiten a los usuarios recuperar, insertar, modificar y eliminar datos de los archivos existentes (Date, 2001). Además deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o de los intentos de acceso sin autorización.

Los sistemas de bases de datos son ampliamente usados, por ejemplo, en bancos, líneas aéreas, universidades, finanzas, ventas, producción, entre otras; los sistemas de bases de datos forman una parte esencial de casi todas las empresas actuales (Silberschatz et al, 2002).

3.2.1 Bases de Datos

Una base de datos (BD) es una colección de datos relacionados, en la cual los datos deben estar estructurados de forma que reflejen fielmente los objetos, las relaciones y las restricciones existentes en la parcela del mundo real representada por la base de datos.

Asimismo, y para que esta representación sea fiable, la base de datos debe ser sensible a los sucesos del mundo real y debe evolucionar para reflejar los cambios que estos sucesos puedan provocar a la parcela del mundo representada (Celma et al, 2003).

Aunque una base de datos es una colección estructurada de datos, no cualquier conjunto de estructuras de datos se puede considerar una base de datos. La tecnología de bases de datos se ha desarrollado intentando dar respuesta a las crecientes exigencias de funcionalidad y eficiencia que los usuarios plantean a los sistemas de información. Las características que definen esta tecnología, y que la diferencian de otras técnicas de gestión de datos anteriores, son las siguientes (Celma et al, 2003):

- **Seguridad:** permite proteger los datos frente a la pérdida total o parcial de información, por fallos del sistema (hardware o software) de forma que los datos puedan reconstruirse, por destrucción causada por fuego, robo, inundaciones, etc., por accesos no autorizados o indebidos.
- **Integridad de los datos:** se refiere a las medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), de operación (introducción de datos incoherentes) o de redundancia.
- **Independencia lógica y física de los datos:** es la propiedad que asegura que los programas escritos por los usuarios son independientes de los cambios realizados en los datos que no usan o en los detalles de representación física de los datos a los que acceden (Celma et al, 2003).
- **Acceso concurrente** por parte de múltiples usuarios: control de concurrencia mediante técnicas de bloqueo.

Gracias al cumplimiento de estos objetivos es que las bases de datos tienen muchas ventajas sobre los sistemas de archivos.

3.2.2 Sistemas Manejadores de Bases de Datos

Un sistema Manejador de Bases de Datos (SMBD) es una herramienta de software que permite la creación y manipulación de bases de datos (Celma et al, 2003). Todas las solicitudes de acceso a la base de datos son manejadas por el SMBD, agregar o eliminar archivos (o tablas), recuperar y almacenar datos desde y en dichos archivos, etc., son características que proporciona el SMBD (Date, 2001).

Los SMBD deben asegurar la independencia, la integridad y la seguridad de los datos. Para cumplir con estos objetivos, los sistemas de bases de datos, independientemente de la familia a la que pertenecen y del fabricante responsable, disponen de componentes con funciones bien definidas, y tienen una arquitectura estándar conocida como arquitectura de tres niveles (Celma, et.al, 2003).

3.2.3 Gestión de transacciones

Varias operaciones sobre la base de datos forman a menudo una única unidad lógica de trabajo. Una **transacción** es una colección de operaciones que se lleva a cabo como una única función lógica en una aplicación de bases de datos (Silberschatz et al, 2002). Es una unidad de programa que consulta y actualiza datos sin violar ninguna de las restricciones de consistencia de la base de datos. Es decir, si la base de datos era consistente antes de ejecutarse la transacción, debe seguir siéndolo tras su ejecución (Pons et al, 2008).

Las propiedades **ACID** es un conjunto de características que se deben garantizar durante la ejecución de las transacciones en una base de datos:

- **Atomicidad (Atomicity):** Todas las operaciones asociadas a una transacción deben ejecutarse por completo o no ejecutarse ninguna de ellas (Pons et al, 2008).
- **Consistencia (Consistency):** una transacción es una unidad de integridad porque mantiene la consistencia de los datos, transformando un estado coherente de datos en otro estado de datos igualmente consistente.
- **Aislamiento (Isolation):** aunque se ejecuten varias transacciones concurrentemente, la ejecución de individual de cada una de ellas no debe interferir en la ejecución de las otras (Pons et al, 2008). El aislamiento requiere que parezca que cada transacción sea la única que manipule el almacén de datos, aunque se puedan estar ejecutando otras transacciones al mismo tiempo. Una transacción nunca debe ver las fases intermedias de otra transacción.
- **Persistencia (Durability):** tras la ejecución con éxito de una transacción, los cambios realizados sobre los datos deben permanecer, incluso si se produce un fallo después (Pons et al, 2008).

3.2.4 Arquitectura de tres niveles

Uno de los propósitos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadoras, los desarrolladores esconden esta complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema. (Silberschatz et al. 2002).

La arquitectura propuesta por el Grupo de Estudio en Sistemas de Administración de Bases de Datos de ANSI/SPARC proporciona estos niveles de abstracción y están divididos en tres niveles como se muestra en la figura 3.1 conocidos como **interno** (nivel físico), **conceptual** (nivel lógico) y **externo** (vistas) (Date, 2001).

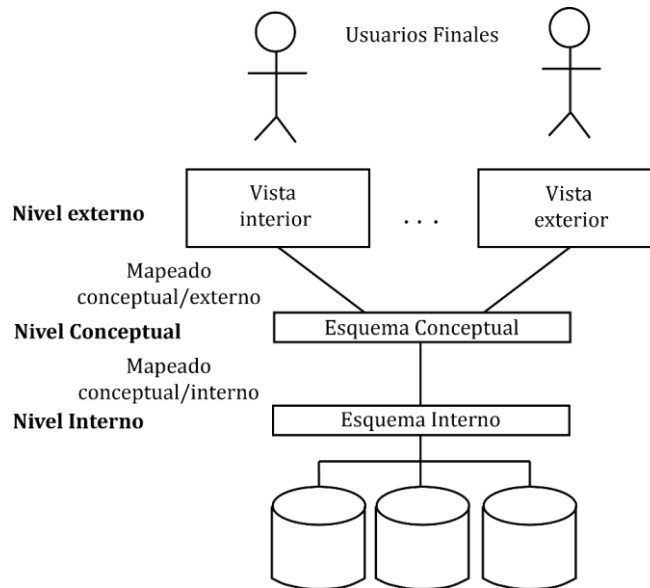


Figura 3.1 *Arquitectura de tres niveles*

- Nivel interno: tiene un esquema interno, que describe la estructura de almacenamiento físico de la base de datos. El esquema interno utiliza un modelo de datos físico y describe todos los detalles de almacenamiento de datos y las rutas de acceso a la base de datos (Elmasri & Navathe, 2007).
- Nivel conceptual: tiene un esquema conceptual que describe la estructura de toda la base de datos para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras de almacenamiento físico y se concentra en describir las entidades, los tipos de datos, las relaciones, las operaciones de los usuarios y las restricciones (Elmasri & Navathe, 2007).
- Nivel externo o de vista: incluye una cierta cantidad de esquemas externos o vistas de usuario. Un esquema externo describe la parte de la base de datos en la que un grupo de usuarios en particular está interesado y le oculta el resto de la base de datos (Elmasri & Navathe, 2007).

3.2.5 Modelado de los datos

La construcción de un sistema de información tiene como objetivo básico la obtención, por medio de una abstracción del mundo real, de un conjunto estructurado de datos y un conjunto de operaciones definidas sobre ellos que permitan satisfacer las necesidades de información de una organización de la forma más eficiente posible (Celma et al, 2003).

El modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia (Silberschatz et al, 2002), que permiten realizar el proceso de abstracción que conduce del mundo real al mundo de los datos (Celma et al, 2003).

Independientemente de las particularidades de cada modelo de datos, todos ellos deben proporcionar conceptos que permitan representar las siguientes propiedades (Celma et al, 2003):

- Propiedades estáticas: objetos de información (entidades), propiedades de los objetos (atributos), relaciones entre objetos, y restricciones sobre los objetos o sus relaciones.
- Propiedades dinámicas: operaciones sobre los objetos o sus relaciones, relaciones entre operaciones (transacciones), y restricciones sobre la evolución de los objetos y sus relaciones.

A lo largo del tiempo se han ido desarrollando diversos modelos de datos entre los que destacan:

- Modelos de datos primitivos: coinciden con los modelos de sistemas de registros, base de los sistemas de administración de archivos, que constituyeron el soporte de los sistemas de información durante la década de los años sesenta. Las operaciones básicas que proporcionan para manipular los objetos son la lectura y escritura de registros (Celma et al, 2003).
- Modelo jerárquico y red: aparecieron como extensiones de los modelos de sistemas de archivos, buscando una mayor eficiencia en la manipulación de las relaciones entre objetos. En ellos, los objetos se siguen representando como registros organizados en archivos, pero sin embargo proporcionan estructuras de datos más complejas, que permiten expresar directamente las relaciones entre objetos (Celma et al, 2003).
- Modelo Relacional: en el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Los diseños de base de datos a menudo se realizan en el **modelo Entidad-Relación**, y después se traducen al modelo relacional (Silberschatz et al, 2002).
- Modelos de datos semánticos: surgen con la intención de aumentar la capacidad expresiva de los modelos clásicos (jerárquico, red y relacional); para ello, incorporan conceptos y mecanismos de abstracción que permiten modelar la realidad de una forma más natural. Hasta el momento han sido utilizados fundamentalmente como herramientas para el diseño de bases de datos. El modelo más utilizado es el modelo

Entidad-Relación, este modelo está basado en una percepción del mundo real (Celma et al, 2003).

- Modelo de datos orientado a objetos y el modelo relacional orientado a objetos: con modelos de datos semiestructurados, permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos (Silberschatz et al, 2002).

3.2.6 Lenguajes de Bases de Datos

Un sistema de base de datos proporciona un lenguaje de definición de datos (LDD) para especificar el esquema de la base de datos, un lenguaje de manipulación de datos (LMD) para expresar las consultas a la base de datos y las modificaciones (Silberschatz et al, 2002) y un lenguaje de control de datos (LCD) que está conformado por sentencias que controlan la integridad, la atomicidad y en general la seguridad de los datos. Con el lenguaje de control de datos, se crean perfiles de usuario en donde se otorgan o quitan permisos a los usuarios para la creación, lectura, borrado o modificación de los datos, así como la ejecución de otros programas almacenados en la base de datos (del Pilar, 2013).

En la práctica, estos lenguajes forman parte de un único lenguaje de bases de datos, tal como el **SQL**, que es ampliamente usado (Silberschatz et al, 2002).

3.2.7 Programas de aplicación e interfaces de usuario

Los programas de aplicación son programas que se usan para interactuar con la base de datos. Los programas de aplicación se escriben usualmente en un lenguaje anfitrión, tal como C, C++ o Java.

Para acceder a la base de datos, las instrucciones LMD necesitan ser ejecutadas desde el lenguaje anfitrión. Una manera de hacerlo es proporcionando una interfaz de programas de aplicación que se pueden usar para enviar instrucciones LMD y LDD a la base de datos, y recuperar resultados. El estándar de conectividad abierta de base de datos (ODBC, Open Data Base Connectivity) se usa con el lenguaje C y para Java se usa el estándar de conectividad de Java con base de datos (JDBC, Java Data Base Connectivity).

La mayoría de los usuarios de un sistema de bases de datos no están situados actualmente junto al sistema de bases de datos, sino que se conectan a él a través de una red (Silberschatz et al, 2002).

3.3 PROCESO DE DISEÑO DE UNA BASE DE DATOS

El primer paso, es la recopilación de requisitos y el análisis. Durante este paso, los diseñadores de bases de datos entrevistan a los potenciales usuarios de la base de datos para comprender y documentar sus requisitos en cuanto a datos. En paralelo al estudio de estos requisitos, resulta útil especificar los requisitos funcionales de la aplicación, que consisten en las operaciones (o transacciones) definidas por el usuario que se aplicarán a la base de datos, incluyendo las recuperaciones y las actualizaciones (figura 3.2).

Una vez recopilados y analizados todos los requisitos, el siguiente paso es crear un esquema conceptual para la base de datos, mediante un modelo de datos conceptual de alto nivel. Este paso se denomina **diseño conceptual**. El esquema conceptual es una descripción concisa de los requisitos de datos por parte de los usuarios e incluye descripciones detalladas de los tipos de entidades, relaciones y restricciones (Elmasri & Navathe, 2007).

El siguiente paso es la implementación real de la misma mediante un DBMS comercial. La mayoría de los DBMS comerciales actuales utilizan un modelo de datos de implementación (como el modelo de base de datos relacional), de modo que el esquema conceptual se transforma de modelo de datos de alto nivel en modelo de datos de implementación. Este paso se conoce como **diseño lógico**; su resultado es un esquema de base de datos en el modelo de datos de implementación del DBMS.

El último paso es la fase de **diseño físico**, durante la cual se especifican las estructuras de almacenamiento interno, los índices, las rutas de acceso y la organización de los archivos de la base de datos. En paralelo a estas actividades, se diseñan e implementan los programas de aplicación (Elmasri & Navathe, 2007).

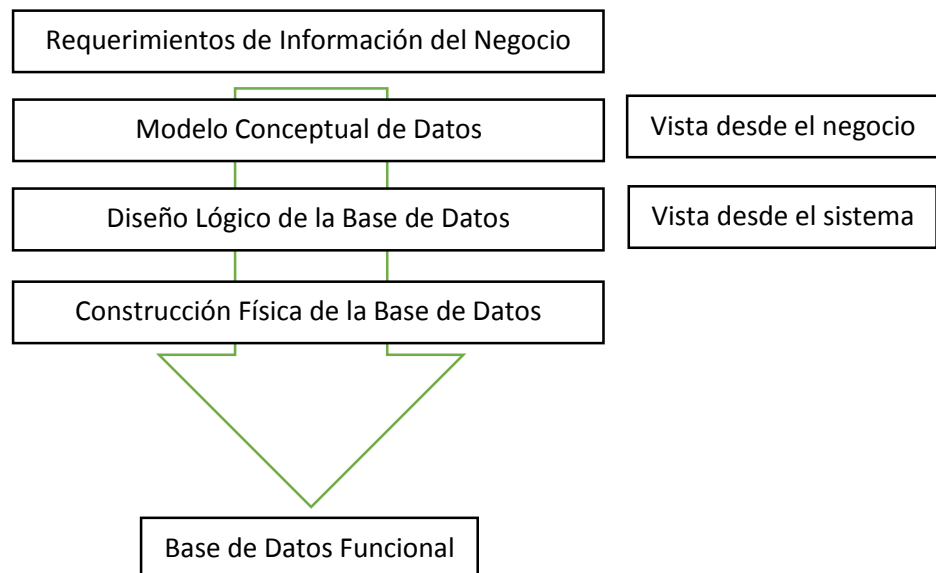


Figura 3.2 Proceso de desarrollo de una Base de Datos

3.3.1 Diseño Conceptual

Se alimenta del análisis de requerimientos y produce un diseño que trata de reflejar como son los datos. El diseño conceptual trata de crear un modelo parcial del Universo donde se trata de capturar lo suficiente para poder soportar todas las funciones a las que servirá el sistema final. El resultado final de esta fase es un Esquema de la Base de Datos. Dentro de esta fase es común el uso del **modelo Entidad-Relación** (del Pilar, 2013).

El diseño conceptual de datos debe ser independiente del hardware y software que son eventualmente utilizados para la implantación. Esto permite tomar una vista objetiva de las necesidades del negocio sin las restricciones de un ambiente específico (Oracle, 2001).

3.3.1.1 Modelo Entidad-Relación

El modelo de datos entidad-relación (MER) es una herramienta para el modelado de datos de un sistema de información. Fue propuesto por Peter Chen en 1976, está basado en una percepción del mundo real consistente en objetos básicos llamados **entidades** y de **relaciones** entre estos objetos y sus **propiedades**. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual (Silberschatz et al, 2002). El MER consiste en:

1. Partir de una descripción textual del problema o sistema de información (requerimientos).
2. Se hace una lista de sustantivos y verbos (los sustantivos son posibles entidades y los verbos posibles relaciones).
3. Analizando las frases y las reglas de negocio se determina la cardinalidad de las relaciones y demás detalles.
4. Se elabora el diagrama entidad-relación.
5. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se puedan reflejar en el diagrama.

Hay tres nociones básicas que emplea el modelo de datos E-R: conjuntos de entidades, conjuntos de relaciones y atributos.

- **Entidades:** Una entidad es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos, puede ser pensada como una cosa de importancia que necesita el negocio para obtener información. Todas las entidades son sustantivos, pero no todos los sustantivos son entidades (Oracle, 2001).

- **Atributos:** Los atributos son los descriptores de una entidad. Todas las entidades deben tener atributos o no son entidades. Los atributos son la forma de obtener información de las entidades almacenadas (Oracle, 2001).
- **Relaciones:** Una relación es la forma en que una entidad se relaciona con otra, es lo que tiene que ver una entidad con otra. Las relaciones son las reglas del negocio para ligar la información que el negocio requiere. Una relación es bidireccional, se debe tomar en cuenta el flujo de esta en las dos direcciones (Oracle, 2001). Existen diferentes tipos de relaciones (cardinalidad):
 - ✓ *Uno a Uno:* una entidad en A se asocia con a lo sumo una entidad B, y una entidad B se asocia con a lo sumo una entidad A (Figura 3.3a).
 - ✓ *Muchos a Uno:* una entidad en A se asocia con a lo sumo una entidad en B. Una entidad en B, sin embargo, se puede asociar con cualquier número de entidades (ninguna o varias) en A (Figura 3.3b).
 - ✓ *Muchos a Muchos:* Una entidad en A se asocia con cualquier número de entidades (ninguna o varias) en B, y una entidad en B se asocia con cualquier número de entidades (ninguna o varias) en A (Figura 3.3c).

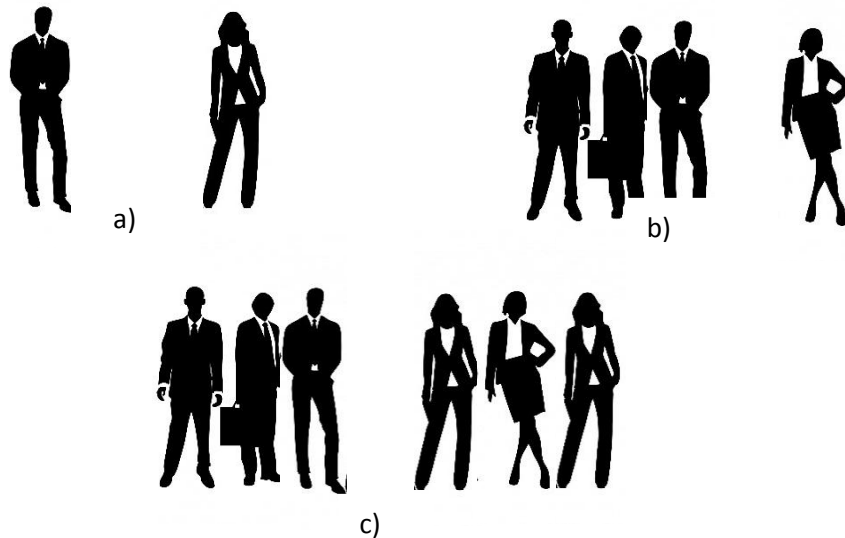


Figura 3.3 Tipos de relaciones que se pueden presentar en un modelo E-R.

Aparte de las entidades, las relaciones y los atributos, se deben tomar en cuenta los siguientes conceptos para poder realizar el modelo E-R:

- *Detalles de los atributos:* Cuando se ha identificado que un atributo existe para una entidad, y que podrá tener un valor para cada instancia de la entidad, se pueden examinar los detalles del atributo. Esto permite tener las reglas de validación para el atributo (Oracle, 2001).
- *Identificadores únicos (claves):* los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos. Una **clave** permite identificar un conjunto de atributos suficientemente para distinguir las entidades entre sí y también a las relaciones. Una clave candidata es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades. Una **clave primaria** es una clave candidata que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades (Silberschatz et al, 2002).
- *Descripción de entidad y atributos:* documentar las entidades y atributos y su significado claramente.
- *Formatos de atributos:* todo atributo debe tener asignado un formato de información (tipo de dato, longitud máxima, unidad de medida).
- *Valores permitidos en los atributos:* valores por default, rango de valores, lista de valores.
- *Dominios:* Un dominio es un conjunto de reglas de validación del negocio, formato de restricciones y otras propiedades que se aplican a más de un atributo. Los dominios son utilizados para cosas que raramente cambian, por ejemplo, días de la semana o unidades monetarias (Oracle, 2001).

3.3.1.2 Diagrama Entidad-Relación

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un Diagrama E-R (DER), es la representación gráfica del MER.

Como ya se mencionó los tres elementos principales del modelo entidad relación son las entidades, los atributos y las relaciones, así como los identificadores únicos, y estos elementos de igual manera están presentes en el DER (figura 3.4). Los otros elementos que forman parte del DER son los siguientes:

- *Supertipos:* es una entidad que puede ser dividida en pequeños grupos mutuamente exclusivos (una instancia de un subtipo no puede ser instancia de otro subtipo). Un supertipo puede tener atributos o sólo ser utilizado como el nombre de un grupo.
- *Subtipos:* es una entidad que representa la composición de un grupo en un supertipo. Cada subtipo puede tener sus propios atributos, pero inherentemente tendrá todos los atributos asignados al supertipo.

- Arcos: representa a las relaciones exclusivas; al crear el modelo, se puede encontrar que una entidad tiene una misma relación con una entidad A o con una entidad B. Ambas relaciones pueden ser válidas, pero no al mismo tiempo: cualquiera de A o B, pero no ambas.
- Relaciones Recursivas: es una relación entre una instancia de una entidad y otra instancia de la misma entidad.
(Oracle, 2001).

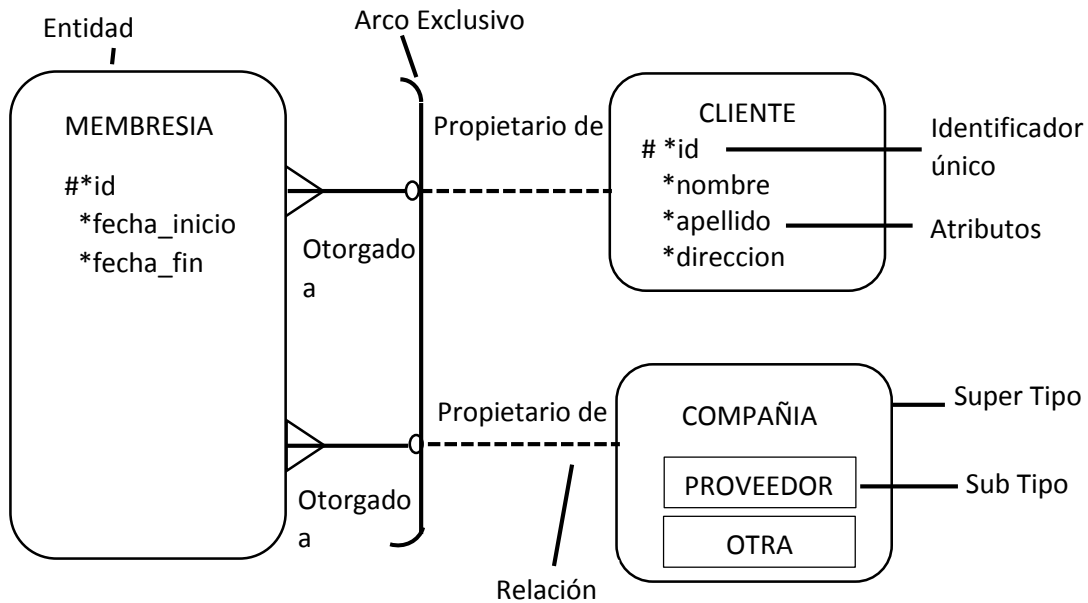


Figura 3.4 Elementos del diagrama Entidad-Relación

3.3.2 Diseño lógico

El diseño lógico toma el Modelo Entidad-Relación de la fase de Diseño Conceptual, y produce un modelo que se acerca más a la implementación en un DBMS. En esencia esta fase transforma el modelo Entidad-Relación en **tablas** que podrán ser implementadas en un DBMS particular.

El modelo de datos que se usa para esta etapa es el **Modelo Relacional**. Posteriormente este modelo es transformado para eliminar ciertas anomalías, debidas principalmente a la redundancia, el proceso a través del cual se da esto se conoce como Normalización (del Pilar, 2013).

3.3.2.1 Modelo Relacional

El modelo relacional fue presentado por primera vez por Ted Codd, de IBM Research, en 1970 (Elmasri & Navathe, 2007). Se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos, y ha conseguido esta posición

debido a su simplicidad, que facilita el trabajo del programador en comparación con otros modelos anteriores como el de red y el jerárquico (Silberschatz et al, 2002).

El modelo relacional está compuesto por **tuplas** (filas), **atributos** (columnas) y **relaciones** (tablas) y representa la base de datos como una colección de relaciones (figura 3.5). Cuando una relación está pensada como una tabla de valores, cada fila representa una colección de valores relacionados. El nombre de la tabla y de las columnas se utiliza para ayudar a interpretar el significado de cada uno de los valores de las filas.

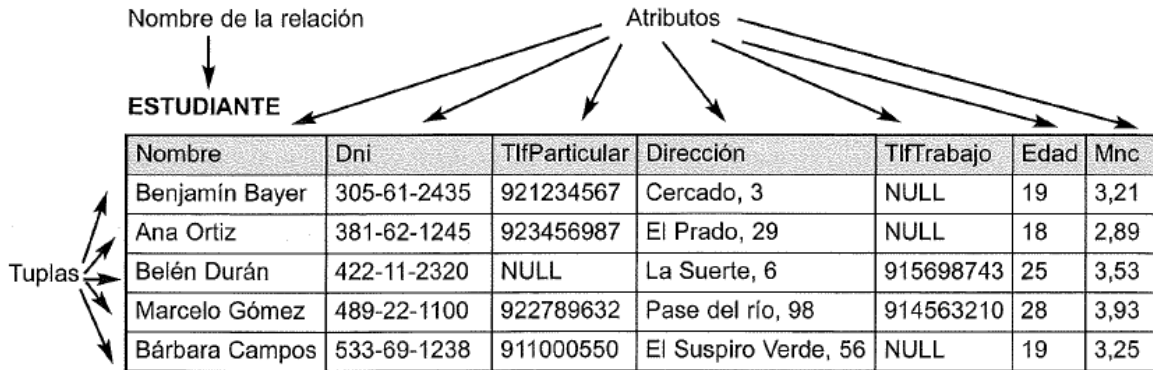


Figura 3.5 Elementos del Modelo Relacional

Características de las relaciones

- Todos los registros de una relación deben tener el mismo número de atributos, aunque alguno esté vacío (se admite el valor NULL).
- Los valores de los atributos son atómicos: fijado un registro, cada atributo toma un único valor (no se admiten atributos multi-valorados).
- Restricción de unicidad: no se admiten registros duplicados. Esto es, dos registros de una misma relación deben diferir, al menos en el valor de un atributo.
- Dos atributos de la misma relación no pueden tener el mismo nombre.
- Dos atributos de relaciones distintas pueden tener el mismo nombre.
- El orden de los atributos no importa.
- El orden de los registros no importa

Estas características son las que diferencian a las relaciones de los sistemas de registros.

3.3.2.2 Restricciones de integridad

Las restricciones de integridad proporcionan un medio de asegurar que las modificaciones hechas a la base de datos por los usuarios autorizados no provoquen la pérdida de la consistencia de los datos. Por lo tanto, las restricciones de integridad protegen a la base de datos contra daños accidentales (Silberschatz et al, 2002).

Generalmente, existen muchas restricciones, en los valores de un estado de base de datos. Estas restricciones están derivadas de las reglas de negocio que dicha base de datos representa (Elmasri & Navathe, 2007).

En el modelo relacional se contemplan los siguientes tipos de restricciones:

- Valores de atributos: son los predicados definidos por el administrador de la base de datos sobre los valores de los atributos.
- Integridad de entidad: el valor de una tabla debe tener un valor único para cada fila de la tabla.
- Integridad referencial: asegura la integridad entre las llaves foráneas y primarias.

Para cumplir con las restricciones anteriores, existen dos métodos:

1. **Constraints** (a nivel de tabla o columna): unique, clustered, primary key, references, NULL, NOT NULL, check y foreign key.
2. **Objetos** (asociados a diversas tablas y columnas): reglas, índices, procedimientos almacenados y triggers.

3.3.2.3 Mapeo del DER al MR

El mapeo del diagrama Entidad-Relación al modelo relacional, se refiere a diseñar el esquema de una base de datos relacional basándose en un diseño de esquema conceptual (Elmasri & Navathe, 2007).

Muchas herramientas de ingeniería de software asistidas por computadora (CASE) están basadas en los modelos ER, o en otros modelos similares. Los diseñadores de bases de datos utilizan interactivamente estas herramientas computarizadas para desarrollar gráficamente el diagrama Entidad Relación, y después lo convierten automáticamente en un esquema de base de datos relacional en el LDD de un DBMS relacional específico.

3.3.2.4 Normalización

La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por tanto más manejables. Las bases de datos se normalizan para evitar redundancia de los datos, evitar problemas de actualización de los datos en las tablas y proteger la integridad de los datos.

3.3.2.5 Lenguajes del modelo relacional

Un modelo de datos debe incluir un conjunto de operaciones para manipular la base de datos junto con los conceptos necesarios para la definición de su estructura y restricciones. El conjunto de operaciones básicas del modelo relacional es el álgebra relacional. Aunque ninguno de los Sistemas Manejadores de Bases de Datos Relacionales (SMBDR) comerciales

actuales proporcionan una interfaz para las consultas de álgebra relacional, las funciones y operaciones centrales de cualquier sistema relacional están basadas en estas operaciones, también es importante, porque algunos de sus conceptos se han incorporado al lenguaje estándar de consultas SQL (Lenguaje de consulta estándar, *Standard Query Language*) para los SMBDR.

El álgebra relacional permite al usuario especificar las peticiones fundamentales de recuperación. El resultado de una recuperación es una nueva relación, la cual puede ser constituida por una o más relaciones.

Por otro lado el lenguaje SQL aunque se considera un lenguaje de consultas, contiene muchas otras capacidades además de la consulta en bases de datos. Incluye características para definir la estructura de datos, para la modificación de los datos en la base de datos y para la especificación de restricciones de seguridad (Silberschatz et al, 2002).

3.3.3 Diseño físico

Una vez que se tienen las tablas resultantes del diseño lógico es importante el decidir tanto la estructura de almacenamiento y las estrategias de acceso.

Cada vez es más común que los DBMS tengan ya predefinida la estructura de almacenamiento y como estrategia de acceso sólo tengan dos, el acceso secuencial y acceso usando B-Trees. Entonces esta etapa se reduce en términos simples a la selección de los índices para acelerar el acceso. En esta etapa, el diseño de base de datos se implementa y rellena con datos reales.

3.4 BASES DE DATOS ESPACIALES

Hace algunas décadas, los mapas de papel eran el medio principal para sintetizar y representar información geográfica. Manipular esta información estaba limitado a lo manual, sin procesos interactivos. Desde entonces, el rápido desarrollo de nuevas tecnologías para reunir y digitalizar datos geoespaciales, junto con una creciente demanda tanto para la manipulación interactiva como para el análisis de estos datos, se ha generado una necesidad de software dedicado, como los Sistemas de Información Geográfica (SIG) (Rigaux et al, 2002).

Debido al crecimiento constante de volumen de datos geográficos, una de las mayores tareas de un SIG es administrar eficientemente grandes bases de datos de información compleja. En una arquitectura clásica de software, esta tarea es generalmente hecha por un SMBD. La mayoría de los SMBDs usados en la práctica son relacionales. Sin embargo los SMBD relacionales están fuertemente dirigidos hacia las aplicaciones de negocios, las cuales manipulan grandes pero simples conjuntos de datos. Esto resulta en que estos sistemas son

incapaces de manipular datos geográficos, en particular debido a su **componente espacial** inherente. Ha habido grandes esfuerzos para extender y adaptar la tecnología de los SMBD a la información espacial (Rigaux et al, 2002), como IBM, DB2 Spatial Extender, Informix Spatial Datablade u Oracle Spatial (Silberschatz et al, 2002).

Las bases de datos espaciales incluyen un conjunto de tipos de datos especiales, como puntos, líneas y polígonos, para modelar entidades geométricas en el espacio (Dunning, 2013). Por ejemplo, el objeto espacial asociado con un río es una línea, mientras que el objeto asociado con una ciudad es una región (polígono) (Rigaux et al, 2002). Este tipo de datos de datos más complejos son denominados Tipos de Datos Abstractos (**ADT**, Abstract Data Type) que admiten objetos geométricos (espaciales).

3.4.1 Objetos geográficos

Los objetos geográficos corresponden a una entidad del mundo real y tiene dos componentes:

- ✓ *Una descripción.* El objeto es descrito por un conjunto de atributos descriptivos. Por ejemplo, el nombre y los habitantes de una ciudad. Estos también son conocidos como atributos alfanuméricos.
- ✓ *Una componente espacial* que puede incorporar tanto la **geometría** (ubicación en el espacio geográfico y la forma) y la **topología** (relaciones espaciales existentes entre los objetos, tales como proximidad). Por ejemplo una ciudad podría tener como valor geométrico un polígono en un espacio 2D. El componente espacial aislado de un objeto geográfico es lo que llamamos **objeto espacial**.

El atributo espacial del objeto geográfico no corresponde a algún tipo de datos estándar, como un string o un entero, en un ambiente de programación computacional. La representación de la geometría y topología, requiere un modelado poderoso a nivel capa u objeto (Rigaux et al, 2002). También se necesita de un sistema que permita ejecutar tareas como entrada, almacenamiento, recuperación y análisis tanto de datos espaciales como de alfanuméricos.

Una tendencia común es extender el SMBD para incluir nuevos tipos y operaciones al sistema relacional. El lenguaje de consulta SQL es extendido para manipular datos así como los datos descriptivos. Nuevos tipos espaciales (punto, línea, región) son manejados como tipos alfanuméricos. Muchas otras funciones de DBMS, tal como optimización de consultas, son adaptados a fin de que los datos geoespaciales puedan ser manejados eficientemente (Rigaux et al, 2002).

3.4.2 Representación de Datos geográficos

La realidad geográfica debe recogerse en un formato que pueda ser entendido por las computadoras y así susceptible de emplearse dentro de un software especializado para procesar y manipular esta información, lo cual no resulta una tarea sencilla. Una de las preocupaciones principales ha sido la de representar de la mejor manera posible toda la información que podemos extraer de una zona geográfica dada.

El problema principal reside en el hecho de que el detalle real que encontramos en la naturaleza es prácticamente infinito, mientras que la representación y almacenamiento de esa realidad son finitos (Olaya, 2011). El camino que lleva desde la realidad hasta ese conjunto de meros valores numéricos pasa por tres niveles:

Establecimiento de un modelo geográfico. Es decir, un modelo conceptual de la realidad geográfica y su comportamiento. Es un esquema mental que constituye una forma particular de entender el hecho geográfico en sí, pero que todavía no incorpora elementos relativos a su representación o almacenamiento. Existen muchos modelos geográficos distintos, entre los cuales cabe destacar el de **Campos** y el de **Entidades Discretas**.

- *Modelo basado en campos:* cada punto en el espacio es asociado uno o muchos valores de atributos, definidos como funciones continuas en X e Y. La altitud sobre el nivel del mar es un ejemplo de función definida sobre X e Y, cuyo resultado es el valor de la variable H para cualquier punto en el espacio 2D (Rigaux et al, 2002).
- *Modelo basado en Entidades:* concibe un entorno geográfico como un espacio vacío sobre el que se sitúan distintos elementos (entidades) que lo van rellenando, pueden ser elementos geométricos como puntos, líneas y polígonos (Olaya, 2011).

Establecimiento de un modelo de representación. Es decir, una forma de recoger el anterior modelo conceptual y sus características propias reduciéndolo a una serie finita de elementos. Existen dos modelos principales: modelo de representación **vectorial** y modelo de representación **mosaico (ráster)**.

- *Modelo vector:* el dato vectorial es la representación discreta de un objeto, estos se construyen de puntos y bordes (figura 3.6a). Un punto es representado por un par de coordenadas, mientras más complejos sean los objetos, serán representados por estructuras (listas, conjuntos) (Rigaux et al, 2002).
- *Modo Mosaico:* en este modelo el dato ráster es un arreglo de celdas, donde cada una de ellas representa un atributo. Por ejemplo, una ciudad es representada como un conjunto de celdas que cubren el interior de la ciudad (figura 3.6b).

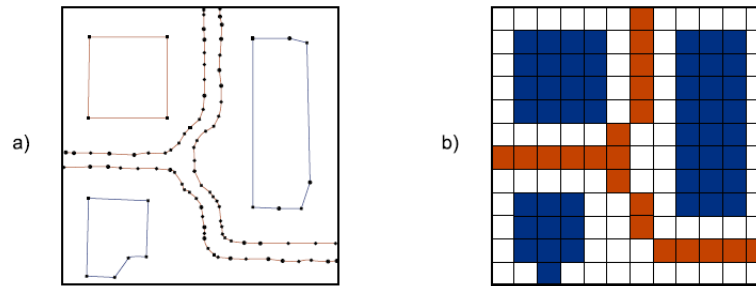


Figura 3.6 Diferencias entre el modelo vectorial (a) y el modelo ráster (b)

Combinando los modelos del espacio geográfico (modelo basado en entidades y modelo basado en campos) con los modos de representación (modo mosaico y modo vector), se tienen los modelos para representar los objetos espaciales: (Rigaux et al, 2002).

Establecimiento de un modelo de almacenamiento. Es decir, un esquema de cómo almacenar los distintos elementos del modelo de representación

3.4.3 Manipulación de datos geográficos

En el álgebra de capas se realizan operaciones simples que toman una o muchas capas como entrada y dan como resultado otra capa. Entre estas operaciones se encuentran la proyección descriptiva, la selección descriptiva, la unión de tópicos y la superposición.

Sin embargo, no todas las operaciones sobre capas van a resultar en otra capa. Por ejemplo, algunas operaciones basadas en mediciones regresarán un número (por ejemplo, la operación distancia). Otras operaciones sobre capas son más complicadas: operaciones topológicas, que se refieren a las relaciones existentes entre datos. Una consulta que usa tales operaciones es, por ejemplo, ¿Qué países colindan con Bélgica?

3.4.4 Modelos lógicos

Mientras que la representación de objetos espaciales se ocupa de su representación interna, en los modelos lógicos se trata con la representación pero de los objetos geográficos. Por lo tanto, las características descriptivas y espaciales de un objeto son tomadas en cuenta simultáneamente. Tomando el ejemplo del modelo relacional, para servir como soporte para aplicaciones geoespaciales, necesita ser extendido tomando en cuenta los tipos no incorporados (tal como enteros o reales). La introducción de estos tipos en el DBMS requiere no sólo una definición de su estructura sino también las operaciones que pueden ser ejecutadas sobre él. Por ejemplo, un tipo polígono, puede ser introducido con la operación PolygonArea (que calcula el área del polígono). Combinaciones de tales tipos de datos y sus operaciones son conocidas como tipos de datos abstractos (**ADTs**) (Rigaux et al, 2002).

CAPÍTULO 4 APLICACIONES WEBMAPPING

4.1 INTRODUCCIÓN

Internet y la Web han influido enormemente tanto en el mundo de la informática como en la sociedad en general. Si nos centramos en la Web, en poco menos de 10 años ha transformado los sistemas informáticos: ha roto las barreras físicas (debido a la distancia), económicas y lógicas (debido al empleo de distintos sistemas operativos, protocolos, etc.) y ha abierto todo un abanico de nuevas posibilidades. Una de las áreas que más expansión está teniendo en la Web en los últimos años son las aplicaciones web. Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página web (Luján, 2002).

Actualmente las empresas, instituciones, administraciones y demás están migrando rápidamente todos sus servicios, aplicaciones, tiendas, etc., a un entorno web que permita a sus clientes y usuarios acceder a todo ello por Internet. Esta red está destinada a convertirse en una suerte de servicio universal de comunicaciones, permitiendo una comunicación universal (Mateu, 2004).

Por otro lado el uso de la web como medio de difusión para los mapas puede ser considerado como una gran avance para la cartografía y abre muchas nuevas oportunidades, tales como mapas en tiempo real, difusión más barata, actualizaciones más frecuentes y baratas tanto de los datos como del software, el contenido de mapas personalizados, fuentes de datos distribuidos y el intercambio de información geográfica (Neumann, 2008).

Las aplicaciones Web relacionadas con la información geográfica, han evolucionado, habiendo ganado día tras día en riqueza hasta el estado actual donde pueden llegar a ofrecer casi tantas funcionalidades como un SIG de escritorio (Olaya, 2011).

En los últimos diez años, la tecnología WebMapping se ha convertido en una opción realista para muchas organizaciones medianas y grandes (Batzli, 2010).

4.2 INTERNET Y LA WEB

Aunque los inicios de Internet se remontan a los años sesenta, no ha sido hasta los años noventa cuando, gracias a la Web, se ha extendido su uso por todo el mundo. En pocos años la Web ha evolucionado enormemente: se ha pasado de páginas sencillas, con pocas imágenes y contenidos estáticos a páginas complejas con contenidos dinámicos que provienen de bases de datos, que da lugar a la creación de aplicaciones web (Luján, 2002), estas nos permite el acceso a multitud de prestaciones y funciones, así como a infinidad de servicios, programas, tiendas, etc. (Mateu, 2004).

4.2.1 Internet

Internet es una gran red mundial de computadoras conectadas que, como todas las redes, permite compartir recursos e intercambiar información. En esta gran red mundial, las computadoras están unidas a través de conexiones de varios tipos y para comunicarse utilizan un protocolo común, el TCP/IP (Ramos, 2014).

Esta red que conecta millones de dispositivos de cómputo (host) a través del mundo, ofrece distintos tipos de servicios a los usuarios, por ejemplo, la web, mensajes de correo electrónico y servicios de telefonía (Kurose & Ross, 2004).

El éxito de Internet se basa mucho en el empleo de TCP/IP, el conjunto de protocolos de comunicación que permiten la comunicación y el intercambio de información de forma independiente de los sistemas en que ésta se encuentra almacenada (Luján, 2002).

Las direcciones que utiliza el protocolo IP para identificar las máquinas en Internet, están formadas actualmente por un número de 32 bits que se suele dividir en cuatro partes de 8 bits separadas por puntos. Como un número de 8 bits ofrece un total de 256 combinaciones, tenemos que cada una de las partes de la dirección IP es un número decimal entre 0 y 255. Son familiares en el mundo Internet direcciones con el formato 195.73.21.15 (Zurdo, 1998).

Con los 32 bits de la actual versión de IP (versión 4), se pueden combinar más de cuatro mil millones de direcciones distintas, lo cual dado el crecimiento exponencial de Internet, pronto dejará de ser suficiente¹. Para solucionar esto aparece la versión 6 de IP, cuyas direcciones están compuestas por 128 bits, incrementándose el número de posibles direcciones hasta unas cifras astronómicas (Zurdo, 1998).

Cuando en internet queremos acceder a un determinado archivo o recurso, éste se puede localizar gracias a su **URL** (Uniform Resource Locator) que es una dirección con un formato convencional que nos permite identificar el recurso de un modo unívoco en la Red.

¹ El 3 de febrero de 2011, la IANA asignó los últimos bloques libres, agotando el pool de direcciones IPv4 disponibles.

Los hosts se conectan mediante enlaces de comunicación, los cuales están formados de diferentes tipos de medios físicos, como cable coaxial, de cobre, fibra óptica y ondas de radio. Normalmente se conectan unos a otros a través de dispositivos de conmutación llamados routers (figura 4.1).

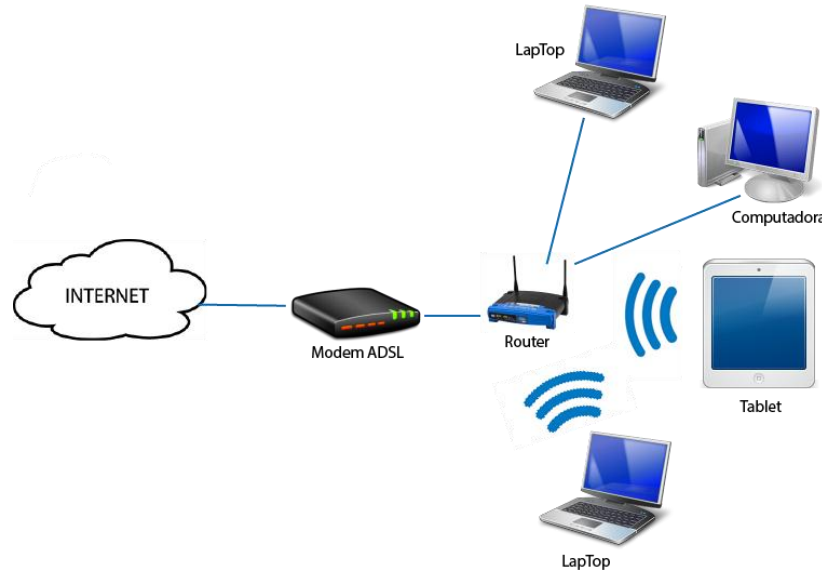


Figura 4.1 Acceso a Internet a través de diferentes dispositivos

Los hosts acceden a Internet a través de proveedores de servicios de internet (Internet Service Providers; ISP). Para permitir la comunicación entre usuarios de Internet y permitir que estos accedan a su contenido en todo el mundo, estos proveedores de baja escala están interconectados mediante proveedores de alta escala nacionales e internacionales (Kurose & Ross, 2004).

4.2.2 La Web

La WWW (World Wide Web), llamado popularmente la **web**, permite al usuario acceder a gran cantidad de información y a distintos tipos de documentos de manera sencilla. Básicamente permite visualizar en la pantalla del usuario “páginas” (páginas web) con información alojada en computadoras remotas. Estas páginas se conectan entre ellas a través de enlaces o links (cada vez que se presiona un link, este nos lleva a otra página web) (Ramos, 2014).

Un navegador web es un programa que permite visualizar la información que contiene una página web, pues interpreta el código de la página y los visualiza en la pantalla.

Las páginas web son archivos escritos en el lenguaje HTML. El desarrollo de estas páginas abarca un amplio grupo de tecnologías, desde las páginas más sencillas que sólo usan el lenguaje HTML hasta las más complejas que usan CSS, JavaScript, applets realizados en Java

o componentes ActiveX. Su modo de empleo es muy sencillo: se basa en el uso de etiquetas que indican que elementos contiene cada página, el formato que hay que aplicar a cada uno de ellos y como se tienen que distribuir por la página. Un sitio web es un conjunto de páginas web relacionadas entre sí (Luján, 2002).

El éxito espectacular de la web se basa en dos puntos fundamentales: el protocolo **HTTP** (Hypertext Transference Protocol) es el protocolo que los navegadores utilizan para ver las páginas web y el lenguaje **HTML** (HyperText Markup Language), que proporciona un mecanismo de composición de páginas enlazadas simple, fácil y altamente eficiente. La Web es una de las muchas aplicaciones distribuidas que utilizan los servicios de comunicación proporcionados por internet.

4.2.3 Arquitectura cliente/servidor

Esta arquitectura de red implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Normalmente los servidores son computadoras potentes dedicadas a gestionar unidades de disco (servidor de bases de datos), impresoras, tráfico de red o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.

La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio, permitiendo situar a cada función en la plataforma más adecuada para su ejecución y posibilita el acceso a los datos independientemente de donde se encuentre el usuario (Luján, 2002).

La arquitectura cliente/servidor permite la separación de funciones en tres niveles (Luján, 2002):

- **Lógica de presentación.** Se encarga de la entrada y salida de la aplicación con el usuario. Sus principales tareas son: obtener información del usuario, enviar la información a la lógica de negocio para su procesamiento, recibir los resultados del procesamiento de la lógica de negocio y presentar estos resultados al usuario.
- **Lógica de negocio (o aplicación).** Se encarga de gestionar los datos a nivel de procesamiento. Actúa de puente entre el usuario y los datos. Sus principales tareas son: recibir la entrada del nivel de presentación, interactuar con la lógica de datos para ejecutar las reglas de negocio que tiene que cumplir la aplicación (facturación, cálculo de nóminas, control de inventario, etc.) y enviar el resultado del procesamiento al nivel de presentación.
- **Lógica de datos.** Se encarga de gestionar los datos a nivel de almacenamiento. Sus principales tareas son: almacenar los datos, recuperar los datos, mantener y asegurar la integridad de los datos.

4.3 APLICACIONES WEB

Una aplicación Web se puede definir como una aplicación en la cual el usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet (o a través de una Intranet) y que recibe una respuesta que se muestra en el propio navegador; éstas son un tipo especial de aplicación cliente/servidor (Luján, 2002).

En las aplicaciones web suelen distinguirse tres niveles: el *nivel superior* que interacciona con el usuario (el cliente web, normalmente un navegador), el *nivel inferior* que proporciona los datos (la base de datos) y el *nivel intermedio* que procesa los datos (el servidor web).

El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HTML del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en Java. Por lo tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonido, etc.).

El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. Las tecnologías destinadas a programar la parte servidor de las aplicaciones web son: ASP, ColdFusion, JSP, servlets, PHP, etc. Todas ellas son muy similares, se basan en los mismos principios y ofrecen prestaciones y resultados equivalentes.

4.3.1 Entornos Web

Las aplicaciones web se emplean en tres entornos informáticos muy similares: Internet, intranet y extranet.

- Internet: posee un diseño descentralizado, ya que cada computadora en Internet es independiente. Sus operadores pueden elegir qué servicio de Internet usar y que servicios locales quieren proporcionar al resto de Internet.
- Intranet: es una red de computadoras basada en los protocolos que gobiernan Internet (TCP/IP) que pertenece a una organización y que es accesible únicamente por los miembros de la organización, empleados u otras personas con autorización. Una intranet puede estar o no conectada a Internet. Un sitio web en una intranet es y actúa como cualquier otro sitio web, pero los firewall lo protegen de accesos no autorizados.

- Extranet: es una intranet a la que pueden acceder parcialmente personas autorizadas ajenas a la organización o empresa propietaria de la intranet. Una extranet proporciona diferentes niveles de acceso a personas que se encuentran en el exterior de la organización. Esos usuarios pueden acceder a la extranet sólo si poseen un nombre de usuario y una contraseña con los que identificarse. La identidad del usuario determina que partes de la extranet puede visualizar. Además, para acceder a una extranet se suelen emplear medios de comunicación seguros, como Secure Socket Layer (SSL) y Virtual Private Network (VPN).

(Luján, 2002).

4.3.2 Ventajas y desventajas de las aplicaciones web

Muchas empresas han descubierto que las anteriores tecnologías se pueden emplear en las aplicaciones cliente/servidor que emplean. Además, una vez que se tiene una aplicación que funciona en una intranet, aparece la posibilidad de permitir su uso a través de Internet, lo que facilita el teletrabajo o la movilidad de los empleados de una empresa (Luján, 2002).

Una ventaja clave del uso de aplicaciones web es que el problema de gestionar el código en el cliente se reduce drásticamente. No sólo se ahorra tiempo porque reducimos la actualización a una sola máquina, sino que hay que desplazarse de un puesto de trabajo a otro. Una segunda ventaja, relacionada con la anterior, es que se evita la gestión de versiones. Se evitan problemas de inconsistencia en las actualizaciones, ya que no existen clientes con distintas versiones de la aplicación (Luján, 2002).

Una tercera ventaja es que si la empresa ya está usando Internet, no se necesita comprar ni instalar herramientas adicionales para los clientes. La última ventaja, es la independencia de la plataforma; para que una aplicación web se pueda ejecutar en distintas plataformas (hardware y sistema operativo), sólo se necesita disponer de un navegador para cada una de las plataformas, y no es necesario adaptar el código de la aplicación a cada una de ellas (Luján, 2002).

Una desventaja, que sin embargo está desapareciendo rápidamente, es que la programación en la web no es tan versátil o potente como la tradicional. El lenguaje HTML presenta varias limitaciones, como es el escaso repertorio de controles disponibles para crear formularios. Por otro lado, al principio las aplicaciones web era básicamente de “sólo lectura”: permitían una interacción con el usuario prácticamente nula. Sin embargo, con la aparición de nuevas tecnologías de desarrollo como Java, JavaScript y ASP, esta limitación tiende a desaparecer (Luján, 2002).

4.3.3 Arquitectura de las aplicaciones web

Las aplicaciones web se basan en una arquitectura cliente/servidor (figura 4.2). Existen diversas variantes de la arquitectura básica según como se implementen las diferentes funcionalidades de la parte servidor. Las arquitecturas más comunes son:

1. Todo en un servidor: una única computadora aloja el servicio de HTTP, la lógica de negocio y la lógica de datos y los datos. El software que ofrece el servicio de HTTP gestiona también la lógica de negocio.
2. Servidor de datos separado: a partir de la arquitectura anterior, se separa la lógica de datos y los datos a un servidor de bases de datos específico.
3. Todo en un servidor, con servicio de aplicaciones: en la arquitectura número 1 se separa la lógica de negocio del servicio de HTTP y se incluye el servicio de aplicaciones para gestionar los procesos que implementan la lógica de negocio.
4. Servidor de datos separado, con servicio de aplicaciones: a partir de la arquitectura anterior, se separa la lógica de datos y los datos a un servidor de bases de datos específico.
5. Todo separado: las funcionalidades básicas del servidor web se separan en tres servidores específicos.

(Luján, 2002).

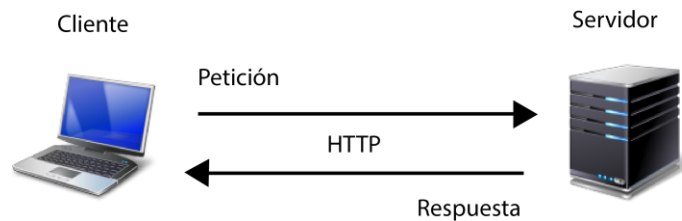


Figura 4.2 Diagrama de una arquitectura básica cliente/servidor

El objetivo de separar las distintas funcionalidades en distintos servidores es aumentar la escalabilidad del sistema de cara a obtener un mayor rendimiento. Al separar las distintas funcionalidades en distintos servidores, cada uno de ellos se puede configurar (dimensionar) de forma adecuada a los requisitos que presenta cada uno de ellos.

Otra ventaja que se obtiene al separar las funcionalidades, es que al aislar la lógica de negocio y la lógica de datos en servidores separados que no están conectados directamente a Internet se aumenta el nivel de seguridad, ya que no es tan fácil acceder a ellos (Luján, 2002).

4.4 APLICACIONES WEBMAPPING

El incremento en la capacidad de transmisión de información a través de Internet, proporciona nuevas vías de comunicación antes impensables. La publicación de cartografía es un hecho desde hace unos años. El empleo de herramientas de libre distribución permite de una forma barata dar a conocer por Internet un SIG (Coll, et. al, 2005).

Los mapas estáticos que constituían los primeros elementos con componente geográfica en la Web han evolucionado hasta verdaderas aplicaciones que pueden convertir un navegador Web en una plataforma SIG completa (figura 4.3) (Olaya, 2011).

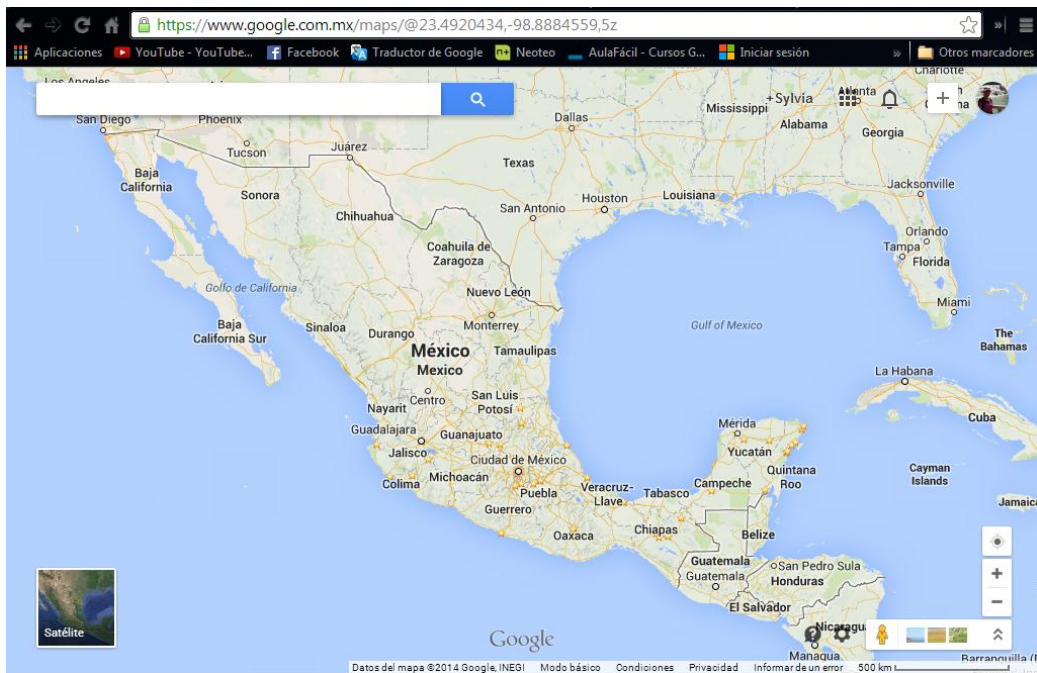


Figura 4.3 *Aplicaciones completas y poderosas como Google Maps son el presente de la cartografía y una gran herramienta para muchas personas*

4.4.1 Mapas web

Del mismo modo que podemos acceder a otros tipos de información a través de Internet o de una red local, también podemos emplear esta para acceder a información geográfica y trabajar con ella (Olaya, 2011).

En términos simples los mapas web pueden ser clasificados en tres grupos:

- Los mapas estáticos pueden ilustrar puntos, líneas y áreas de información. En su forma más simple, estos mapas son visualizados como imágenes .GIF o .JPG. No se les puede hacer acercamientos o ser editados. Estos mapas proveen una excelente manera de acceso a colecciones históricas de mapas (Peterson, 2005). Incorporar

este tipo de mapas a una página web no requiere ninguna tecnología particular, y puede llevarse a cabo con elementos genéricos tanto del lado del cliente como del lado del servidor, pues el dato realmente no es un dato geográfico como tal, sino una mera imagen (y esa imagen no va acompañada de información tal y como su sistema de referencia) (Olaya, 2011).

- Un mapa dinámico es aquel que no se compone de una imagen inmóvil, sino que esta varía y se adapta en función de los requerimientos del usuario o según alguna serie de parámetros prefijados. Como ejemplo están los mapas animados que encuadran una determinada zona y muestran la variación de una variable a lo largo del tiempo. Mapas de variables climatológicas o una serie animada de mapas que reflejan el avance de un incendio son ejemplos habituales de este grupo (Olaya, 2011).
- **Mapas interactivos.** La interactividad es la que aporta las posibilidades necesarias para comenzar a incorporar funciones SIG a la cartografía web. La forma de interactividad más básica que se implementa en una página web en el trabajo con cartografía es la que permite la modificación de la forma en que los datos geográficos se visualizan. Las herramientas que permiten modificar la escala de visualización (acercarse o alejarse) y desplazar el mapa, aportan a la cartografía web muchas posibilidades nuevas. Entre ellas, es de destacar que mediante estas herramientas la extensión de los datos no se encuentra limitada por la propia extensión de la pantalla o la dimensión del navegador (Olaya, 2011).

De especial importancia para el desarrollo de estas capacidades ha sido la popularización y mejora de las tecnologías que permiten el desarrollo de las denominadas Aplicaciones Ricas de Internet (RIA). Este tipo de aplicaciones llevan a la web algunos elementos de las tecnologías de escritorio, y en general permiten optimizar el volumen de datos necesario para operar con la aplicación dentro del entorno del navegador (Olaya, 2011).

Si no se emplean estas tecnologías, un cambio mínimo en la configuración de la página por parte del usuario (por ejemplo, modificar el encuadre del mapa en una aplicación SIG), requiere la recarga total de la página, de la misma forma que sucede cuando hacemos clic en un hiperenlace (Olaya, 2011).

En un entorno RIA, sin embargo, se cargan al inicio (en el primer acceso a la página) los elementos que constituyen la aplicación en sí, y posteriormente se transmite únicamente los datos que vayan siendo necesarios a medida que el usuario opere con la aplicación. AJAX es una técnica de desarrollo muy popular en este sentido, y de la que los SIG Web hacen uso habitualmente (Olaya, 2011).

4.4.2 WebMapping (Cartografía Web)

Es el proceso de diseño, implementación, generación y entrega de mapas en la web, este proceso está ligado con los estándares que da la Open Geospatial Consortium (OGC), que también tiene estándares para los SIG, con el fin de lograr un intercambio de información geográfica que beneficie a todos. Con base a estos estándares, se está tratando de usar un solo formato para los mapas que es el GML (Geographic Markup Language), aunque en estos momentos, se siguen usando muchos otros formatos como los sgv, shp, swf, entre muchos otros (Olaya, 2011).

La Cartografía Web es a menudo una presentación multimedia, pero están ganando cada vez más capacidades analíticas. Además de representar un conjunto de datos geográficos y permitir al usuario navegar en ellos e incluso editarlos, pueden extraerse resultados a partir de esos datos (Olaya, 2011). Las tecnologías Web Mapping, permiten incorporar las ideas de los SIG dentro de páginas Web, utilizando un navegador Web como aplicación principal.

Un caso especial de esta tecnología son los mapas que aparecen en los dispositivos móviles, como celulares inteligentes, PDAs, GPS y otros dispositivos. Si los mapas de estos dispositivos se muestran por un navegador web móvil, pueden ser considerados como WebMapping (Neumann, 2008).

4.4.3 Historia del WebMapping

Naturalmente, la historia del WebMapping está estrechamente ligada a la historia y los avances tecnológicos de la web.

No está claramente documentado cuando fue que le primer mapa fue publicado en la web, pero en junio de 1993, Xerox Parc, implementaron el primer servidor de mapa interactivo en la web. Esta aplicación de servidor de mapas escrito en Perl permitía al usuario especificar la extensión del mapa, elegir las capas del mapa, la proyección del mapa y algunos parámetros de estilo. La respuesta fue un archivo HTML y una imagen raster gif incrustada que representaba al mapa. Se puede considerar como el predecesor del actual estándar WMS (Web Map Service).

En 1994, el National Atlas of Canada estuvo en línea como el primer gran atlas en línea de todo el mundo. En marzo de 1995, Sun hizo la primera versión disponible de Java y se introdujo en los principales navegadores web. Desde el comienzo de 1995 hasta ahora, los applets de Java han sido utilizados para entregar el contenido de los mapas interactivos en la web. En diciembre de 1995, Brendan Eich (Netscape) introdujo la primera versión de JavaScript que permitió los primeros sitios web interactivos, incluyendo interactividad sencilla en mapas raster. A principios de 1996, Mapquest, uno de los primeros y populares servicios de buscador de direcciones y de ruta se puso en línea. Los resultados fueron visualizados usando mapas web y descripciones textuales. A mediados de 1996, Multimap,

un servicio de mapas web de Reino Unido, se puso en línea. Ahora es uno de los sitios web de UK más populares. A finales de 1996, Geomedia WebMap 1.0 fue publicado, uno de los primeros servidores de mapas comerciales.

A mediados del año 1997, UMN Map Server 1.0 fue lanzado. Se convirtió en el más popular servidor de mapas de código abierto y es utilizada por miles de sitios de WebMapping hoy en día. En septiembre de 2001, el W3C lanzó la especificación SVG 1.0, un lenguaje de marcado basado en XML para la integración de vectores gráficos, raster gráficos y texto, también soportando animación, multimedia, scripting e interactividad, y la internacionalización.

SVG abre muchas oportunidades para los mapas web interactivos pero tomó un tiempo hasta que SVG fue adecuadamente soportada por los navegadores web. El Tirol Atlas, publicado en 2001 poco después de SVG 1.0, convirtiéndose en una referencia, es el primer gran atlas en línea ampliamente utilizando la tecnología SVG. Sigue siendo uno de los pocos atlas que ofrece interactividad superior.

En enero de 2003, los perfiles móviles de SVG, SVG Tiny y SVG basic, fueron publicados, lo que permite los servicios de localización y los mapas móviles. En el mismo año, la NASA publicó WorldWind, una de las primeras aplicaciones virtuales interactivas de la Tierra. A principios de 2005, Google lanzó Google Maps una aplicación WebMapping desarrollada en la parte superior de Dynamic HTML, ECMAScript y XMLHttpRequests (también conocida como Ajax). El hecho de que Google abrió su API para terceros desarrolladores para reutilizarlo gratuitamente, los hizo rápidamente muy popular, provocando miles de Google Maps derivados en aplicaciones WebMapping.

Más tarde, en el mismo año, Google lanzó Google Earth, una aplicación virtual de la Tierra bought de Keyhole. KML un lenguaje de markup basado en XML, permite a los usuarios añadir sus propios contenidos de geodatos 3D (Neumann, 2008).

4.4.4 Arquitectura WebMapping

El avance de las redes locales y de Internet ha permitido que se acceda a la información geográfica a través de una arquitectura cliente-servidor. Para ello es necesario contar con componentes en el lado del servidor que distribuyan la información y componentes en el lado del cliente para acceder a esta. Las tecnologías WebMapping permiten incorporar las ideas de los SIG dentro de páginas web, utilizando un navegador web como aplicación principal (Olaya, 2011).

Por ejemplo, un usuario con una computadora con conexión a internet, envía una petición de un mapa mediante su navegador a algún servidor y éste se comunica con un servidor de mapas que es el encargado de interactuar con la base de datos espacial que contiene los datos e imágenes de los mapas (figura 4.4), el servidor de mapas con base a la petición

devuelve la información solicitada y el mapa solicitado es mostrado en el navegador del usuario (Olaya, 2011).

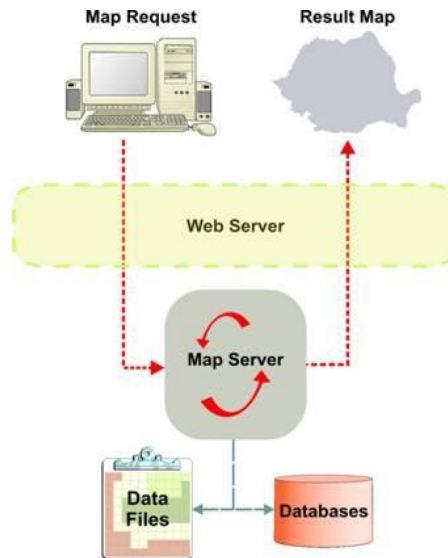


Figura 4.4 Arquitectura de una aplicación WebMapping, consta de 4 partes importantes: la base de datos espacial, el servidor de mapas, el servidor web y el cliente

Servidor Web: es el responsable de manejar las peticiones HTTP de los navegadores web y otras aplicaciones de usuario. El lado del servidor incluye, el envío de peticiones a recursos dinámicos, como lenguajes de script en el servidor. El servidor web más popular es Apache, seguido por Microsoft Internet Information Server, Netscape y otros (Neumann, 2008).

Servidor de Aplicaciones: conectan varios componentes de software con el servidor web y un lenguaje de programación. A modo de ejemplo, un servidor de aplicaciones puede permitir la comunicación entre la API de un SIG y el servidor web, una base de datos espacial u otras aplicaciones. Los servidores web de aplicaciones también son útiles en el desarrollo de aplicaciones WebMapping complejas en tiempo real (Neumann, 2008).

Servidor de mapas: más conocido por sus siglas en inglés IMS (Internet Map Server) provee mapas o cartografía a través de Internet. El servidor WMS puede generar mapas a petición, usando parámetros, tales como orden de las capas del mapa, el estilo/simbolización, la extensión del mapa, formato de datos, proyección, etc. Los formatos de imagen típicos para el mapa resultante son png, jpeg, gif o SVG. Un servidor WMS de código abierto es UMN Mapserver. Alternativas comerciales existen en la mayoría de los vendedores de SIG comerciales, tales como ESRI ArcIMS, Intergraph Geomedia WebMap y otros (Neumann, 2008).

Bases de datos Espaciales: son necesarias siempre que una aplicación WebMapping tiene que lidiar con el cambio dinámico de datos o con grandes cantidades de datos geográficos.

Permiten las consultas espaciales, manipulaciones de geometría y ofrecen diversos formatos de importación y exportación. Un ejemplo muy conocido de una base de datos espacial de código abierto es PostGIS (Neumann, 2008).

4.4.5 Protocolos

Los protocolos que se pueden emplear para hacer las solicitudes de los datos geográficos incluyen Web Map Service (WMS) y Web Feature Service (WFS).

Un componente fundamental de los mapas web es la imagen del mapa. Un **WMS** es un protocolo estándar que sirve imágenes de mapas georreferenciadas generadas por un servidor de mapas. El servicio ofrece una sencilla interfaz HTTP para solicitar imágenes de mapas de una o más bases de datos espaciales. Una petición WMS define la(s) capa(s) geográficas y el área de interés para ser procesada. La respuesta a la solicitud es una o más imágenes de mapas (devueltas como JPEG, PNG, etc.) que se pueden mostrar en una aplicación de un navegador web. La interfaz también es compatible con la capacidad de especificar si las imágenes enviadas deben ser transparentes para que las capas de varios servidores se puedan combinar o no.

Un servidor de mapas web también puede devolvernos los datos geográficos reales que componen las imágenes de mapa. Esto permite a los usuarios crear sus propios mapas y aplicaciones a partir de los datos de origen, convertir datos entre distintos formatos y ser capaz de manipular geográficamente en bruto los datos servidos. El protocolo utilizado para devolver los elementos de los datos geográficos se denomina **WFS**.

4.4.6 Ventajas y desventajas WebMapping

- No es necesario un software SIG específico, ya que no requiere más que un simple navegador Web.
- Información más actualizada, incluso en tiempo real. La Web es el canal ideal para transmitir la información de forma inmediata y flexible.
- Independencia del sistema. Un mapa Web puede verse y usarse del mismo modo en cualquier computadora, con independencia del sistema operativo, el navegador e incluso el dispositivo empleado. Si este mapa se basa en estándares abiertos, la solución es todavía más interoperable.
- Personalización de aplicaciones. Una de las tendencias más importantes en el ámbito del WebMapping es la creación de aplicaciones que personalizan una base común para determinado uso.
- Combinación de cartografía y otros elementos. Si llevamos las capacidades SIG a un navegador, además de estas dispondremos en ese navegador de muchas otras

posibilidades, tales como la representación de elementos multimedia (video, sonido, etc.) o el uso de hiperenlaces, utilizando geometrías vectoriales o zonas sensibles en un mapa web, cualquier parte del mapa puede enlazar a otras páginas web o servicios web. Por ejemplo, un mapa de la ciudad se puede vincular a los horarios del transporte público correspondiente a cada autobús o estación de tren (Neumann, 2008).

- Mediante el uso de perfiles de usuario, los filtros personales de estilo y simbolización, los usuarios pueden personalizar, configurar y diseñar sus propios mapas. Los problemas de accesibilidad pueden ser tratados de la misma manera (Neumann, 2008).

La importancia de las tecnologías Web se debe, por tanto, principalmente a una razón social y no a una tecnológica.

Por otro lado, a pesar de la creciente disponibilidad de herramientas libres y comerciales para crear WebMapping y aplicaciones Web SIG, los mapas web son aún complejos para desarrollar. Muchas de las tecnologías, los módulos, los servicios y las fuentes de datos tienen que ser dominados e integrados (Neumann, 2008).

A diferencia de los EE. UU., donde los datos geográficos son recopilados por instituciones gubernamentales y suelen estar de forma gratuita o barata, los datos geográficos suelen ser muy caros. Esto es un grave obstáculo para muchos proyectos de WebMapping de bajo presupuesto (Neumann, 2008).

4.4.7 Herramientas para el desarrollo de aplicaciones WebMapping

Cualquier entorno de programación, lenguaje de programación y framework se pueden utilizar para implementar proyectos WebMapping. En cualquier caso, tanto las tecnologías de servidor y de cliente tienen que ser utilizados (Neumann, 2008).

Open Geo Suite es una plataforma completa de aplicaciones geoespaciales. Contiene todo lo necesario para hacer una aplicación web de mapas. Su arquitectura es verdaderamente flexible, ya que permite utilizar las herramientas deseadas, sin depender de un único proveedor. Funciona bien con los sistemas existentes (propietarios u open source), que incluye Google, Oracle, ESRI y Microsoft. Es open source, porque está totalmente construido con componentes de código abierto y garantiza un rápido proceso de desarrollo, con una comunidad detrás que participa y una viabilidad a largo plazo (Morales, 2012).

La suite Open Geo se construye sobre PostGIS, una base de datos espacial, GeoServer, un servidor de datos y mapas compatible con los estándares, GeoWebCache, un acelerador de la visualización de los mapas, servidor de mosaicos de mapas y GeoExt, un cliente API de mapas construido sobre OpenLayers (figura 4.5) (Morales, 2012).

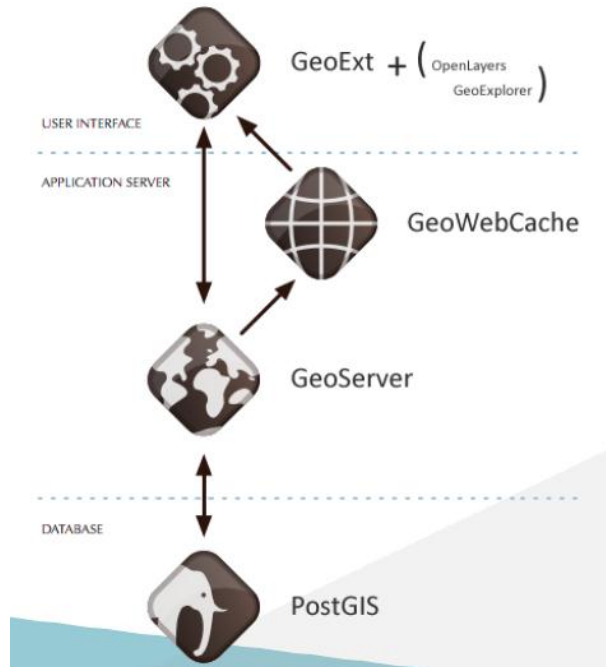


Figura 4.5 Arquitectura de OpenGeo Suite

- Base de datos PostgreSQL: Sistema de gestión de base de datos relacional, es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostGIS es una extensión que convierte el sistema de base de datos PostgreSQL en una base de datos espacial (Morales, 2012).
- Servidor de mapas GeoServer: GeoServer es un servidor de código abierto escrito en Java que permite a los usuarios compartir y editar datos geoespaciales. GeoServer publica los datos de cualquier fuente de datos especializados utilizando estándares abiertos. Es una implementación específica de un servidor web mapping, ofreciendo acceso a los datos en un conjunto conocido de formatos y fuentes (archivos y bases de datos), utilizando protocolos específicos (Morales, 2012).
- GeoWeb Cache: es una herramienta que aumenta el rendimiento de un WMS mediante un pre-dibujo y almacenamiento de imágenes de mapa para dar respuesta más rápida a las solicitudes de los clientes (Morales, 2012).
- OpenLayers: es una librería JavaScript para componer mapas dinámicos en páginas web. Se ha desarrollado para promover el uso de la información geográfica de todo tipo. OpenLayers es software libre y forma parte de los proyectos de la Open Source Geoespatial.

CAPÍTULO 5 DESARROLLO DE SOFTWARE ORIENTADO A OBJETOS CON UML

5.1 INTRODUCCIÓN

En esta era tecnológica en la cual vivimos, nuestras vidas están regidas de gran manera por las computadoras y por el software que las controlan. Cuando todo funciona bien las computadoras son de gran ayuda pero cuando no, el resultado puede ser nefasto, algo que se ha visto a lo largo de los años en múltiples ocasiones.

Aunque en la práctica no se puede garantizar el software perfecto, o sea cero defectos, la pregunta es cuándo el software es *suficientemente bueno*, según Yourdon los tres elementos más importantes del software suficientemente bueno son funcionalidad, calidad y tiempo (Weitzenfeld, 2005). Muchos de los sistemas en uso son ineficientes, engorrosos y difíciles de usar. Esto se debe a que, dado que el desarrollo de sistemas es una actividad humana, hay muchas posibilidades de cometer errores en cualquier etapa del proceso, y por ende, se pueden generar programas difíciles de utilizar y no generar una solución al problema original del cliente (Schmuller, 2001).

En los principios de la computación, los programadores no realizaban análisis muy profundos sobre el problema por resolver. Con frecuencia comenzaban a escribir el programa desde el principio, y el código necesario se escribía conforme se requería, esto en la actualidad es inapropiado en los negocios de alto riesgo. Hoy en día, es necesario contar con un plan bien analizado (Schmuller, 2001).

Cuando un software de computadora se desarrolla con éxito puede cambiar todas las cosas y de hecho las cambia para mejor. Para tener éxito al diseñar y construir un software se necesita una disciplina. Es decir, necesitaremos un enfoque de ingeniería. La ingeniería de software es el estudio de los principios y metodologías para desarrollo y mantenimiento de software (Zelkovitz, 1981). Construir software de computadora es como construir cualquier otro producto satisfactorio, aplicando un proceso que conduce a un resultado de alta calidad que satisface las necesidades de la gente que usará el producto (Pressman, 2002). La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño que los analistas, desarrolladores y clientes acepten como pauta, y el UML es esta notación. El UML ha llegado a ser el estándar en la industria del software, y su evolución continúa (Schmuller, 2001).

5.2 PROCESO DE DESARROLLO DE SOFTWARE

El proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requerimientos de un usuario (Jacobson et al, 2000) para generar un sistema de software de alta calidad. Es importante porque proporciona estabilidad, control y organización a una actividad que puede, si no se controla, volverse caótica (Pressman, 2002). No existe un proceso ideal y diferentes organizaciones han desarrollado enfoques completamente diferentes para desarrollar software, sin embargo todos tienen actividades fundamentales que son comunes para todos ellos:

- 1.- Especificación del software. Se debe definir la funcionalidad del software y las restricciones de sus operaciones.
- 2.- Diseño e implementación del software. Se debe producir software que cumpla su especificación.
- 3.- Validación del software. Se debe validar el software para asegurar que hace lo que el cliente desea.
- 4.- Evolución del software. El software debe evolucionar para cumplir los cambios en las necesidades del usuario.
(Sommerville, 2005).

5.2.1 Modelos del proceso de software

Un modelo de proceso del software es una representación abstracta de un proceso de software. Cada modelo de proceso representa un proceso desde una perspectiva particular por lo que sólo provee información parcial acerca de ese proceso.

Entre los modelos de proceso de software se encuentran:

- El modelo en cascada. Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etc.
- Modelo incremental en el que especificación, el diseño y la implementación del software se dividen en una serie de incrementos los cuales se desarrollan uno a uno.
- Modelo de espiral en el que el desarrollo gira en espiral hacia afuera, empezando con un esbozo inicial y terminando con el desarrollo final del sistema.
- Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un primer sistema se desarrolla rápidamente a partir de especificaciones abstractas. Entonces, se refina con la ayuda del cliente para producir un sistema que satisfaga sus necesidades.
- Desarrollo formal de sistemas. Este enfoque se basa en la producción de una especificación utilizando métodos matemáticos, para construir un programa. La

verificación de los componentes del sistema se lleva a cabo haciendo argumentos matemáticos acordes a su especificación.

(Sommerville, 2005).

Todos los modelos de proceso anteriores tienen ventajas y desventajas. Para sistemas muy grandes, existe la necesidad de utilizar diferentes enfoques para distintas partes del sistema, por lo que se debe utilizar un modelo híbrido. Además, conforme cambian los requerimientos del sistema, es necesario apoyar la iteración de procesos en las partes repetidas del proceso (Sommerville, 2005).

5.2.2 Metodologías para el desarrollo de software

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales.

Actualmente existen muchas metodologías, las cuales se pueden dividir en cuatro tipos principales: ágiles, tradicionales, estructuradas y orientadas a objetos. Sin embargo las metodologías híbridas están marcando la nueva tendencia, al considerar algunas de las mejores características de las metodologías.

En las metodologías tradicionales se considera la importancia de la documentación del sistema, lo cual permite extender y darle mantenimiento al software. Además, estas metodologías proporcionan un orden y una estructura bien definida para el desarrollo del software. No se tiene respuesta rápida a cambios; se genera documentación innecesaria, y se requiere mucho tiempo en el modelado del sistema (Piattini et al., 2007). Algunos ejemplos de metodologías tradicionales son: RUP (Rational Unified Process, Proceso Racional Unificado, IBM, 2011), OMT (Object Modeling Technique, Técnica de Modelado de Objetos, Rumbaugh et al., 1990 y Yourdon, 1976).

Las metodologías ágiles, presentan respuestas rápidas y efectivas al cambio; tienen un plan de proyecto flexible, y muestran simplicidad, de manera general, en el desarrollo. Sin embargo, tienen la desventaja de generar poca documentación y no hacer uso de métodos formales. Algunos ejemplos de este tipo de metodologías son: XP (eXtreme Programming, Programación Extrema, Beck, 2010) y Scrum (Scrum group, 2011) (Jiménez y Orantes, 2012).

Por otro lado, las metodologías estructuradas comenzaron a desarrollarse a fines de la década de los 70's con la programación Estructurada, mientras que las metodologías orientadas a objetos se enfocan principalmente en el modelado de un sistema en términos de objetos. A diferencia de las metodologías estructuradas, se identifican inicialmente los objetos del sistema para luego especificar su comportamiento (Weitzenfeld, 2005). Existe un gran número de metodologías orientadas a objetos:

- **Proceso Unificado (UP):** es la versión libre de la metodología creada por Grady Booch, Ivar Jacobson y James Rumgaugh. Se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, cada ciclo concluye con versión del producto. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones (Jacobson et al, 2000).
- Rational Unified Process RUP: es la versión comercial de la metodología UP
- eXtreme Programming: es una metodología utilizada para desarrollar software de alta calidad de la manera más rápida posible y con el mayor beneficio para el cliente. Se caracteriza por tener ciclos de desarrollo extremadamente breves, integración constante, retroalimentación continua por parte del cliente, pruebas automatizadas regulares y enfoque en equipo.

5.2.3 Metodologías Web

El avance de Internet y las comunicaciones ha provocado en los últimos años el nacimiento de nuevas propuestas metodológicas para la web. Sin embargo, la mayoría de ellas han centrado su trabajo principalmente en las etapas de de diseño e implementación. En la mayoría de estas propuestas el tratamiento de requisitos ha sido tratado con una menor importancia.

En su estudio, Escalona (2001) describe las siguientes metodologías:

- SOHDM (Scenario-based Object-Oriented Hypermedia Desing Methodology, 1998): esta propuesta ofrece un modelo de escenarios propio, denominado SAC, para representar los requisitos. Para el desarrollo de los mismos hace uso del diagrama de contexto propuesto en los DFD (Diagrama de Flujo de Datos). En la actualidad ha caído en desuso, principalmente por el uso de los DFD.
- OOHDM (Object Oriented Hypermedia Design Model, 1999): es una propuesta ampliamente aceptada para la web. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web. OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 4 etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas, implementación.
- UWE (UML-Based Web Engineering, 1999): es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. UWE ha sido ampliamente aceptado en los últimos años. Entre las ventajas más importantes de UWE es su uso 100% UML.

5.3 DESARROLLO DE SOFTWARE ORIENTADO A OBJETOS

Hay muchas formas de enfocar un problema utilizando una solución basada en el software, un enfoque muy utilizado es la visión orientada a objetos.

Vivimos en un mundo de objetos, estos objetos existen en la naturaleza, en entidades hechas por el hombre, en los negocios y en los productos que usamos. Pueden ser clasificados, descritos, organizados, combinados, manipulados y creados. Por esto no es sorprendente que se proponga una visión orientada a objetos para la creación de software de computadora, una abstracción que modela el mundo de forma tal que nos ayuda a entenderlo y gobernarlo mejor (Pressman, 2002).

La orientación a objetos fomenta una metodología basada en componentes para el desarrollo de software, de manera que primero se genera un sistema mediante un conjunto de objetos, luego podrá ampliar el sistema agregándole funcionalidad a los componentes que ya había generado o agregándole nuevos componentes, y finalmente podrá volver a utilizar los objetos que generó para el sistema cuando cree uno nuevo, con lo cual se reducirá sustancialmente el tiempo de desarrollo de un sistema (Schmuller, 2001).

5.3.1 Comparación con el modelo tradicional

El modelo tradicional, también conocido como programación estructurada, un programa o aplicación consta de múltiples datos y funciones “globales”. El término global describe el hecho que todos los datos o funciones son “visibles” en todo el programa y, por lo tanto, pueden ser llamados desde cualquier ubicación en la aplicación (Weitzenfeld, 2005).

Este tipo de programación tiene dos problemas principales: obliga a un programador a que organice su programa de acuerdo con la arquitectura de la computadora, en otras palabras, que piense como la máquina (debería ser lo opuesto), y por otro lado, al estar separados de las funciones, los datos se vuelven globalmente visibles para poder ser llamados. Dada esta situación, cualquier cambio en la estructura de alguno de los datos pudiera llegar a requerir la modificación de todas las funciones del programa en correspondencia con los cambios en los datos.

5.3.2 Ventajas del enfoque orientado a objetos

A diferencia de la programación tradicional, la programación orientada a objetos define una estructura de más alto nivel llamada objeto, que ofrece ventajas sobre la programación tradicional:

- ✓ Permitir al programador que organice su programa de acuerdo con abstracciones de más alto nivel, siendo éstas más cercanas a la manera de pensar de la gente. En otras

palabras, los objetos son las unidades de representación de las aplicaciones, por ejemplo, cuentas de banco, reservaciones de vuelos, etc. (Weitzenfeld, 2005).

- ✓ El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente poco acoplada. Esto lleva a menores efectos colaterales cuando se deben hacer cambios y provoca menos frustración en el ingeniero de software y en el cliente. Además, los sistemas orientados a objetos son más fáciles de adaptar y más fácilmente escalables (por ejemplo: pueden crearse grandes sistemas ensamblando subsistemas reutilizables) (Pressman, 2002).
- ✓ Los datos y funciones se guardan dentro de objetos (figura 5.1). Los datos están ubicados en el centro del objeto (un concepto puramente ilustrativo), lo cual hace que un cambio en su estructura sólo afecte las funciones del mismo objeto, pero no al resto de la aplicación (Weitzenfeld, 2005).

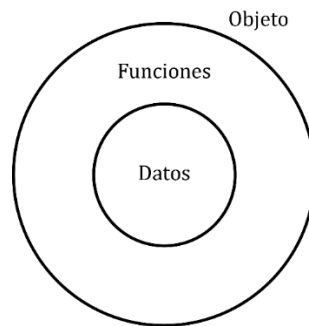


Figura 5.1 *Un objeto guarda sus datos y funciones, lo que permite que al realizar un cambio sólo afecte a ese objeto y no a toda la aplicación*

5.3.3 Características de la Orientación a Objetos

El dominio del problema se caracteriza mediante un conjunto de objetos con atributos y comportamientos específicos. Los objetos son manipulados mediante una colección de funciones (llamadas métodos, operaciones o servicios) y se comunican entre ellos mediante un protocolo de mensajes. Los objetos se clasifican mediante clases y subclasses.

Un objeto encapsula tanto datos como los procesos que se aplican a esos datos. Esta importante característica permite construir clases de objetos e inherentemente construir bibliotecas de objetos reutilizables (la reutilización es un atributo importantísimo en la ingeniería del software). Además, los componentes de software derivados mediante el paradigma de objetos muestran características (como la independencia funcional, la ocultación de información, etc.) asociadas con el software de alta calidad (Pressman, 2002).

La ingeniería de software orientada a objetos sigue los mismos pasos que el enfoque convencional. El análisis identifica las clases y objetos relevantes en el dominio del problema, el diseño proporciona detalles sobre la arquitectura, las interfaces y los componentes, la implementación transforma el diseño en código, y las pruebas chequean tanto la arquitectura como las interfaces y los componentes (Pressman, 2002).

5.3.4 Conceptos Básicos de la Orientación a Objetos

La orientación a objetos es un paradigma que depende de ciertos principios fundamentales: objetos, clases, herencia, polimorfismo, mensajes, cohesión y acoplamiento (Pressman, 2002).

Objetos: los objetos concretos y virtuales, están a nuestro alrededor, ellos conforman nuestro mundo y el software actual simula al mundo (o un segmento de él), y los programas, por lo general, imitan a los objetos del mundo (Schmuller, 2001). Un objeto encapsula datos (representados como una colección de atributos) y algoritmos que procesan estos datos. Estos algoritmos son llamados operaciones, métodos o servicios (Pressman, 2002).

Clases: es una descripción generalizada (por ejemplo, una plantilla, un patrón o un prototipo) que describe una colección de objetos similares. Por definición, todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para la manipulación de los atributos (Pressman, 2002).

Atributos: describen a las clases y a los objetos. Las entidades de la vida real están a menudo descritas con palabras que indican características estables. La mayoría de los objetos físicos tienen características tales como forma, peso, color y tipo de material (Pressman, 2002).

Encapsulamiento: mecanismo básico para ocultar los detalles internos del objeto de los demás objetos. La esencia del encapsulamiento es que cuando un objeto trae consigo su funcionalidad, esta última se oculta. Los objetos encapsulan lo que hacen; es decir, ocultan la funcionalidad de sus operaciones de otros objetos y del mundo exterior. La importancia del encapsulamiento, es que permite reducir el potencial de errores que pudieran ocurrir. En un sistema que consta de objetos, éstos dependen unos de otros en diversas formas. Si uno de ellos falla y los especialistas de software tienen que modificarlo de alguna forma, el ocultar sus operaciones de otros objetos significará que tal vez no será necesario modificar los demás objetos (Schmuller, 2001).

Herencia: es el proceso por el cual un objeto adquiere las propiedades de otro. Esto es importante, ya que supone la base del concepto de clasificación jerárquica. Por ejemplo, un perro es parte de la clase mamífero, que a su vez está contenida en una clase mayor, la clase Animal. Sin la utilización de jerarquías, cada objeto necesitaría definir explícitamente todas sus características. Sin embargo, mediante el uso de la herencia, un objeto sólo necesita definir aquellas cualidades que lo hacen único en su clase (Schildt, 2001).

Polimorfismo: mediante el polimorfismo se definen múltiples funciones con nombres similares sólo que en distintas clases. Las funciones son implementadas de manera diferente en las clases. El polimorfismo es útil para extender la funcionalidad del sistema al definir nuevas clases aún desconocidas al momento de especificarlo (Weitzenfeld, 2005).

Mensajes: son el medio a través del cual interactúan los objetos. El paso de mensajes mantiene comunicado un sistema orientado a objetos. Los mensajes proporcionan una visión interna del comportamiento de objetos individuales y del sistema OO como un todo. Los mensajes y los métodos son dos caras de la misma moneda. Los métodos son los procedimientos invocados cuando un objeto recibe un mensaje (Pressman, 2002).

Cohesión: es una indicación cualitativa del grado que tiene un objeto para centrarse en una sola cosa. Un componente OO, debe diseñarse de manera que posea todas las operaciones trabajando conjuntamente para alcanzar un propósito único y bien definido. Es decir, en la construcción de software se desea conseguir *una alta cohesión*.

Acoplamiento: está representado por el número de conexiones físicas entre los elementos del diseño (clases, objetos). En el diseño de software la meta es conseguir un acoplamiento lo más bajo posible, de esta manera, los errores no se propagan tan fácilmente. Se evitan los daños colaterales cuando el software es modificado.

Abstracción: los seres humanos abordan la complejidad mediante la abstracción. Por ejemplo, no consideramos un coche como un conjunto de cientos de miles de partes individuales, sino como un objeto correctamente definido y con un comportamiento determinado. Esta abstracción nos permite utilizar el coche para ir al mercado sin estar agobiados por la complejidad de las partes que forman el coche (Schildt, 2001). La abstracción consiste en elevar el nivel de las representaciones necesarias para un sistema de software, de manera que se reduzcan los detalles.

5.4 INTRODUCCIÓN A UML

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. El UML (Lenguaje Unificado de Modelado) es una herramienta que cumple con esta función, ya que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien esté involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo (Schmuller, 2001). Permite los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas.

En UML, un sistema viene representado por cinco vistas diferentes que lo describen desde diferentes perspectivas. Cada vista se representa mediante un conjunto de diagramas. En UML están presentes las siguientes vistas:

- *Vista de usuario.* Representa el sistema (producto) desde la perspectiva de los usuarios (llamados actores en UML). El diagrama de caso de uso es el enfoque elegido para modelar esta vista.

- *Vista estructural*: los datos y la funcionalidad se muestran desde dentro del sistema, es decir, modela la estructura estática (diagrama de clases y objetos).
- *Vista de comportamiento*: esta parte del modelo del análisis representa los aspectos dinámicos o de comportamiento del sistema. También muestra las interacciones o colaboraciones entre los diversos elementos estructurales descritos en las vistas anteriores (diagramas de colaboración y secuencias).
- *Vista de implementación*: los aspectos estructurales y de comportamiento se representan aquí tal y como van a ser implementados (diagrama de componentes).
- *Vista del entorno*: aspectos estructurales y de comportamiento en el que el sistema a implementar se representa (diagrama de distribución).

(Pressman, 2002).

5.4.1 Importancia del UML

Conforme aumenta la complejidad del mundo, los sistemas informáticos también deberán crecer en complejidad. En ellos se encuentran diversas piezas de hardware y software que se comunican a grandes distancias mediante una red, misma que está vinculada a bases de datos que, a su vez, contienen enormes cantidades de información. Para manejar tanta complejidad, la clave está en organizar el proceso de diseño de tal forma que las personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. El UML proporciona tal organización (Schmuller, 2001).

Otra característica del desarrollo de sistemas contemporáneos que demanda un diseño sólido son las adquisiciones corporativas. Cuando una empresa adquiere a otra, la nueva organización debe tener la posibilidad de modificar aspectos importantes de un proyecto de desarrollo que esté en progreso. Un anteproyecto bien diseñado facilitará la conversión. Si el diseño es sólido, un cambio en la implementación procederá sin problemas.

5.4.2 Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas del sistema, a las cuales se les conoce como **modelo**. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema (Schmuller, 2001).

5.4.2.1 Diagrama de Clases

Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. El área superior contiene el nombre, la central contiene los atributos, y la inferior las acciones. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que muestran la manera en que las clases se relacionan entre sí (Schmuller, 2001).

Los diagramas de clases cuentan con los siguientes elementos: *nombre de la clase*, *atributos* (es una propiedad o característica de una clase), *operación* (es algo que la clase puede realizar), *responsabilidad* (es una descripción de lo que hará la clase), *restricciones* (especifica una o varias reglas que sigue la clase) y *asociaciones*.

Cuando las clases se conectan entre sí de forma conceptual, esta conexión se conoce como asociación. La meta final es crear una idea estática de un sistema, con todas las conexiones entre las clases que lo conforman. La asociación puede funcionar en ambas direcciones y para representarlas en el mismo diagrama, se utiliza un triángulo relleno que indique la dirección de cada asociación (figura 5.3).

La multiplicidad es la cantidad de objetos de una clase que se relacionan con un objeto de la clase asociada. Se representa con números colocados sobre la línea de asociación (figura 5.2).

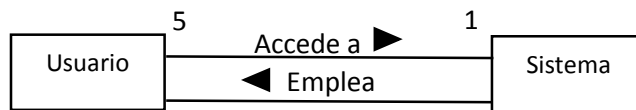


Figura 5.2 Diagrama de clases con sus asociaciones multiplicidad. Permiten al analista hablarles a los clientes en su propia terminología

Las asociaciones reflexivas, se presentan cuando en ocasiones, una clase es una asociación consigo misma. Esto puede ocurrir cuando una clase tiene objetos que puedan jugar diversos papeles. Esto se representa mediante el trazado de una línea de asociación a partir del rectángulo de la clase hacia el mismo rectángulo de la clase (figura 5.3)

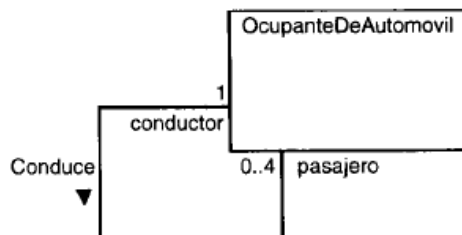


Figura 5.3 Asociaciones reflexivas

La *dependencia* es otro tipo de relación, en donde una clase utiliza a otra. El uso más común de una dependencia es mostrar que la firma de la operación de una clase utiliza a otra clase, la notación del UML para ello es una línea discontinua con una punta de flecha en forma de triángulo sin relleno que apunta a la clase de la que depende (figura 5.4).

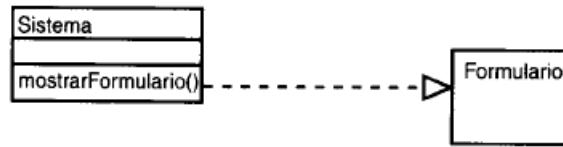


Figura 5.4 Otro tipo de asociación son las dependencias

5.4.2.2 Diagrama de Objetos

Un objeto es una instancia de una clase (una entidad que tiene valores específicos de los atributos y acciones). Para representar a un objeto, se utiliza un rectángulo, como en una clase, pero el nombre está subrayado, el nombre de la instancia específica se encuentra a la izquierda de los dos puntos (:), y el nombre de la clase a la derecha (figura 5.5). Todo objeto de una clase tiene un valor específico en cada atributo.

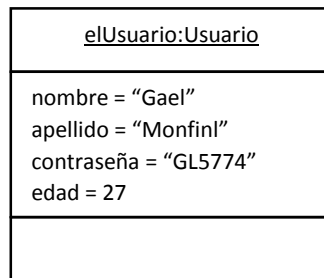


Figura 5.5 Representación de un objeto

5.4.2.3 Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. El caso de uso es un poderoso concepto que ayuda a comprender la forma en que un sistema deberá comportarse y para determinar la forma en que se usará el sistema. Con una colección de casos de uso se puede hacer el bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él.

Los casos de uso se pueden ver como una colección de situaciones respecto al uso de un sistema. Cada escenario describe una secuencia de eventos. Cada secuencia se inicia por una persona, otro sistema, una parte del hardware o por el paso del tiempo. A las entidades que inician secuencias se les conoce como actores. El resultado de la secuencia debe ser algo utilizable ya sea por el actor que la inició, o por otro actor.

Un diagrama de caso de uso se compone de elipses que representan a los casos de uso, los actores con sus nombres y líneas de interconexión (figura 5.6). Cada caso de uso es una colección de escenarios y cada escenario es una secuencia de pasos, tales pasos no aparecen en el diagrama.

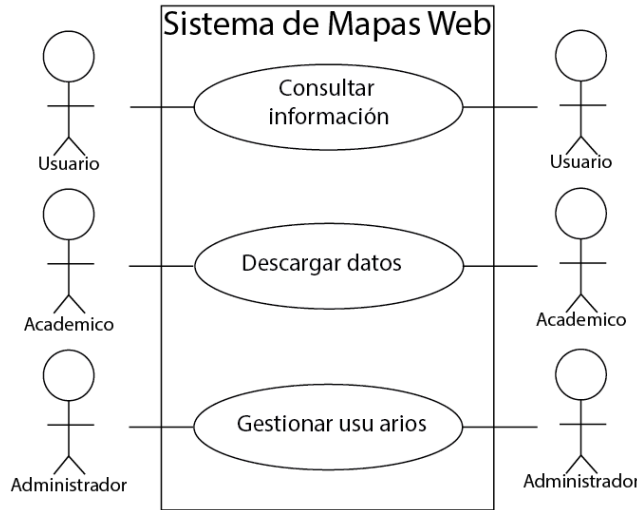


Figura 5.6 Representación de un modelo de caso de uso, hay un actor que inicia un caso de uso y otro que recibirá algo de valor de él.

Existen varias formas en que los casos de uso se pueden relacionar entre sí, la inclusión, la extensión y la generalización. La primera permite eliminar la duplicidad y la forma de hacerlo es tomar cada secuencia de pasos en común y conformar un caso de uso adicional a partir de ellos. Para representar la inclusión se utilizará una línea discontinua con una punta de flecha que conecta las clases apuntando hacia la clase dependiente. Justo sobre la línea, se agrega un estereotipo: la palabra “include” (figura 5.7). La extensión permite volver a utilizar un caso de uso agregándole algunos pasos.

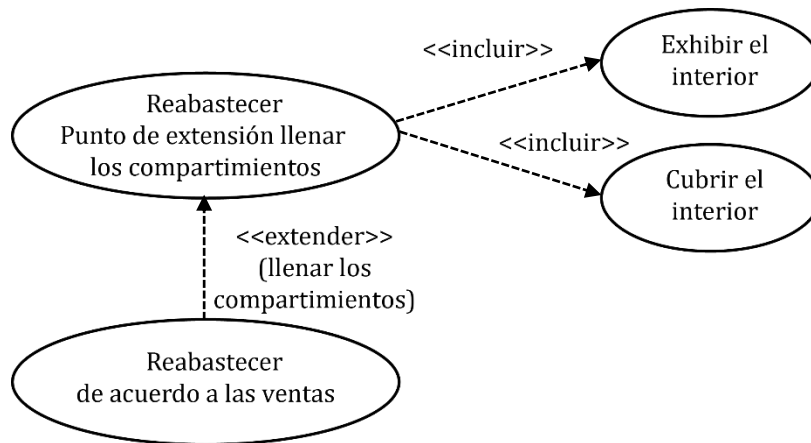


Figura 5.7 Representación de las relaciones Inclusión y Extensión

La última, la generalización, permite la herencia en los casos de uso. Se modelará con líneas continuas y una punta de flecha en forma de triángulo sin rellenar que apunta hacia el caso de uso primario (figura 5.8).

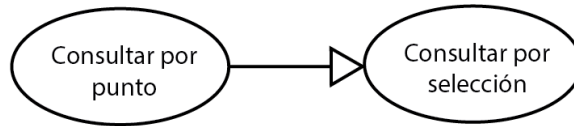


Figura 5.8 La generalización en los casos de uso

5.4.2.4 Diagrama de Estados

En cualquier momento, un objeto se encuentra en un estado particular. El diagrama de estados muestra las transiciones de un estado a otro y los estados en los que puede encontrarse un objeto. Una manera para caracterizar un cambio en un sistema es decir que los objetos que lo componen modificaron su estado como respuesta a los sucesos y al tiempo.

Para el diagrama de estados se utiliza un rectángulo de vértices redondeados que representa a un estado, junto con una línea continua y una punta de flecha, mismas que representan a una transición. Un círculo relleno simboliza un punto inicial y la diana representa a un punto final (figura 5.9) (Schmuller, 2001).

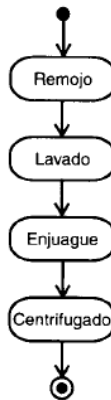


Figura 5.10 Una lavadora podrá estar en la fase de remojo, lavado, enjuague, centrifugado o apagada

5.4.2.5 Diagrama de Secuencias

Los diagramas de clases y objetos representan información estática. No obstante en un sistema funcional los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo.

El diagrama de secuencias consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical (figura 5.10).

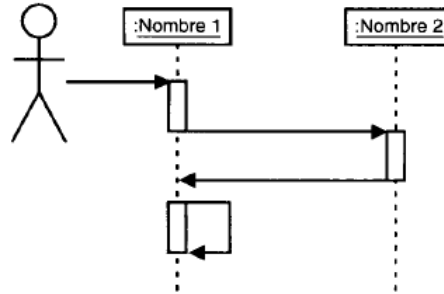


Figura 5.10 Diagrama de Secuencias

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior (Schmuller, 2001).

5.3.2.6 Diagrama de Colaboraciones

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema. El diagrama de colaboraciones está diseñado para este fin. Es similar al de secuencias, ya que también muestra la colaboración entre los objetos, pero destacan el contexto y organización general de los objetos que interactúan.

Para representar un mensaje, se dibujará una flecha cerca de la línea de asociación entre dos objetos, esta flecha apunta al objeto receptor. El tipo de mensaje se mostrará en una etiqueta cerca de la flecha; por lo general el mensaje indicará al objeto receptor que ejecute una de sus operaciones. El mensaje finalizará con un par de paréntesis, dentro de los cuales se colocarán los parámetros (en caso de haber alguno) con los que funcionará la operación (figura 5.11).

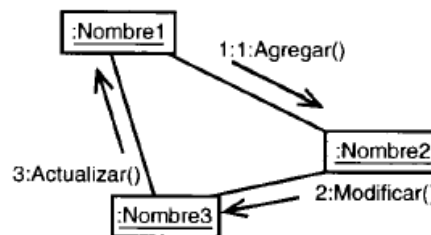


Figura 5.11 Diagrama de colaboraciones

5.3.2.7 Diagrama de Actividades

Las actividades que ocurren dentro de un caso de uso o dentro del comportamiento de un objeto se dan, normalmente, en secuencia, este diagrama muestra los pasos (conocidos como actividades). Un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. A cada actividad se le representa por un rectángulo con las esquinas redondeadas. El procesamiento dentro de una actividad se lleva a cabo y, al realizarse, se continúa con la siguiente actividad. Una flecha representa la transición de una a otra actividad. También cuenta con un punto inicial y uno final.

Casi siempre una secuencia de actividades llegará a un punto donde se realizará alguna decisión (figura 5.12). Ciertas condiciones llevarán por un camino y otras por otro (pero ambas son mutuamente exclusivas).

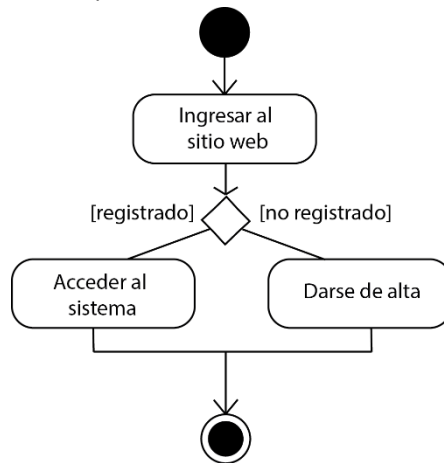


Figura 5.12 Toma de decisiones en el diagrama de actividades

5.3.2.8 Diagrama de Componentes

Este diagrama está íntimamente ligado con los sistemas informáticos. Un componente de software es una parte física de un sistema, y se encuentra en la computadora, no en la mente del analista. Un componente puede ser una tabla, archivo de datos, ejecutable, biblioteca de vínculos dinámicos, documentos y cosas por el estilo.

El símbolo principal de un diagrama de componentes es un rectángulo que tiene otros dos sobrepuestos en su lado izquierdo. Se debe colocar el nombre del componente dentro del símbolo (figura 5.13) (Schmuller, 2001).

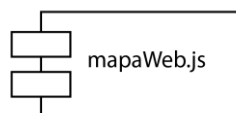


Figura 5.13 Representación de un componente

5.3.2.9 Diagrama de Distribución

Muestra la arquitectura física de un sistema informático. Puede representar equipos y dispositivos, mostrar sus interconexiones y el software que se encontrará en cada máquina. El elemento primordial del hardware es un nodo, que es un nombre genérico para todo tipo de recurso de cómputo. En el UML, un cubo representa a un nodo. Se asigna un nombre para el nodo, y se podrá utilizar un estereotipo para indicar el tipo de recurso que sea (figura 5.14)

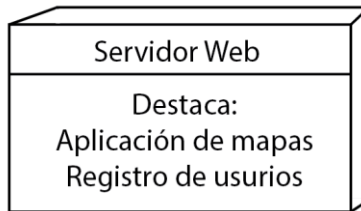


Figura 5.14 Representación de un nodo y su información

5.4.3 Otras características del UML

Aparte el UML proporciona características que le permiten organizar y extender los diagramas, entre ellas se encuentra:

- Paquetes: permite agrupar los elementos de un diagrama. A veces se querrá mostrar que ciertas clases o componentes son parte de un subsistema en particular.
- Notas: es frecuente que alguna parte del diagrama no presente una clara explicación del porqué está allí o la manera en que trabaja. Cuando este sea el caso la nota UML será útil. La nota es un rectángulo con una esquina doblada, y dentro del rectángulo se coloca la explicación.
- Estereotipos: permiten tomar elementos propios del UML y convertirlos en otros. Se representa como un nombre entre dos pares de paréntesis angulares. El concepto de interfaz provee un buen ejemplo. Una interfaz es una clase que realiza operaciones y que no tiene atributos, es un conjunto de acciones que tal vez se quiera utilizar una y otra vez en su modelo. En lugar de inventar un nuevo elemento para representar una interfaz, podrá utilizar el símbolo de una clase con <<Interfaz>> situada justo sobre el nombre de la clase.

Los diagramas de UML le permiten examinar un sistema desde distintos puntos de vista. En un modelo UML no es necesario que aparezcan todos los diagramas. El objetivo es satisfacer a cada persona implicada (Schmuller, 2001).

5.4.4 Uso del UML en aplicaciones Web

En 1998 se definió una extensión denominada WAE (Web Application Extension) para el diseño de aplicaciones web con UML y es la convención más difundida y aceptada hasta nuestros días. Algunos de los ejemplos más comunes de estereotipos que se pueden asociar a las clases y las relaciones entre estas, para representar una aplicación web son:

- *Server Page*: representa una página web que tiene scripts ejecutados por el servidor
- *Client Page*: representa páginas que son dibujadas por el navegador web
- *Form*: representa una colección de campos de entrada que forman parte de una página de lado del cliente

Los estereotipos para las relaciones entre clases son: *Link* (representa un apuntador desde una “client page” hacia una “client page” o “server page”), *Submit* (siempre se da entre una “form” y una “server page” y esta procesa los datos que la “form” le envía) y *Build* (sirve para identificar cuales “server page” son responsables de la creación de una “client page”) (Macías & Orozco, 2014).

5.5 PROCESO DE DESARROLLO DE SOFTWARE CON METODOLOGÍA ORIENTADA A OBJETOS Y UML

Una actividad es una unidad o paso básico de un proceso. En el proceso de software las actividades definen los pasos necesarios para lograr las metas y los objetivos; por ejemplo, especificar los requisitos del sistema. Las actividades deben ser fáciles de definir y seguir, simplificar la comprensión del sistema y ofrecer flexibilidad, precisión y extensibilidad. Las actividades básicas de desarrollo de software orientado a objetos (figura 5.15), son las siguientes: **determinación de requerimientos, análisis, diseño, arquitectura, construcción y pruebas.**

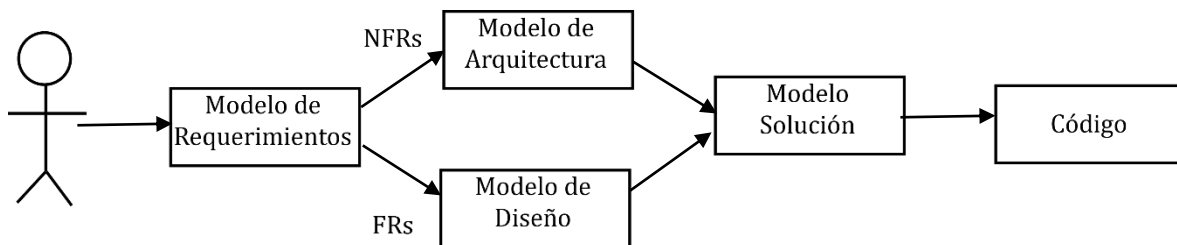


Figura 5.15 Proceso de desarrollo de software con metodología Orientada a Objetos

Existen varios documentos de apoyo a los procesos de software, dependiendo el tipo de empresa se seleccionan artefactos (modelos y diagramas), que son aplicables.

5.5.1 Visión del proyecto

Sirve para definir los requerimientos de alto nivel para dar una idea del sistema a desarrollar. Para determinar la visión del proyecto, primero hay que entrevistar al “dueño” del proyecto, las entrevistas iniciales para determinar la visión del sistema se deben enfocar en los siguientes aspectos: importancia real del proyecto, requerimientos funcionales, riesgos, restricciones, involucrados (Fernández, 2006).

La visión del proyecto utiliza el artefacto **documento de visión** que generalmente tiene cinco secciones:

1. Introducción: que incluya el postulado del problema.
2. Oportunidades de negocio: debe registrar la visión presente y pasada de la compañía y de qué manera el sistema de software va a ayudar al negocio.
3. Solución propuesta: registra todos los requerimientos funcionales y no funcionales identificados por el dueño del sistema.
4. Riesgos: todos los riesgos que se identificaron en las entrevistas iniciales.
5. Restricciones: todas las restricciones que se identificaron en las entrevistas iniciales.

5.5.2 Determinación de Requerimientos

Se especifican las necesidades del sistema a desarrollar. La especificación de requerimientos sirve como base para la negociación entre los desarrolladores y clientes del sistema, también sirve para planear y controlar el proceso de desarrollo. Existen dos tipos de requerimientos:

- Requerimientos funcionales (FRs): es una descripción de que actividades debe llevar a cabo el proyecto por un actor. Normalmente este tipo de requerimientos están vinculados con las entradas, las salidas, los procesos y los datos a almacenar en el sistema.
- Requerimientos no funcionales (NFRs): describen otras prestaciones, características y limitaciones que debe tener el sistema para alcanzar el éxito. Los requerimientos no funcionales engloban características como rendimiento, facilidad de uso, presupuesto, tiempo de entrega, documentación y seguridad.

Dentro de esta fase se utilizan los artefactos: el documento **SRS** (Especificación de Requerimientos de Software), y el **diagrama de casos de uso**. El documento SRS contiene: introducción, restricciones y suposiciones, riesgos, requerimientos funcionales, requerimientos no funcionales y glosario.

5.5.3 Análisis

Se busca comprender los requisitos del sistema con el propósito de estructurar la arquitectura del sistema, responde a la pregunta “**qué**” necesita hacer el sistema. Una vez

especificados los requisitos y definidos los casos de uso, se procede a la construcción del modelo de análisis. Se utilizan los siguientes artefactos: **abstracciones clave**, **tarjetas CRC** (Class Responsibility Collaboration) y **diagrama de clases**.

5.5.4 Diseño

Modela cómo soportará el sistema todos los casos de uso. El modelo de diseño se combina con el modelo de arquitectura para producir el modelo de solución.

La realización de un caso de uso es un proceso que conduce de un caso de uso y sus clases asociadas a un modelo de diseño que soporta el caso de uso. Se debe crear un modelo de diseño para cada caso de uso. El modelo de diseño es visualizado como un **diagrama de colaboración** de UML con componentes de diseño (componentes de frontera, componentes de servicio, entidades) y opcionalmente, se puede convertir el diagrama de colaboración en un **diagrama de secuencia** para tener otro punto de vista.

5.5.5 Arquitectura

En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa, la manera en que interactúan y la estructura de datos que se van a usar. Cuando se habla de componentes se refiere a: clientes y servidores, bases de datos, filtros y capas en un sistema jerárquico. La descomposición arquitectónica es necesaria para la especificación de diversas partes del sistema, reutilización a gran escala. La arquitectura es una descripción compacta y manejable de cómo se organiza el sistema y como interoperan los componentes.

Durante años, el software se desarrolló sin contar con el rol del Arquitecto. En la actualidad existe por dos cambios cruciales en los sistemas que han ocurrido: tamaño y distribución. En los sistemas cliente-servidor, el rol del arquitecto se reduce a la premisa: calcular cerca de los datos, validar cerca del usuario.

5.5.6 Construcción (implementación)

Se expresa la arquitectura del sistema de una forma aceptable para la computadora, es decir, el código. Es la traducción directa del diseño en un lenguaje de programación, en la implementación se construyen los componentes: archivos de código, scripts, ejecutables, tablas.

5.5.7 Pruebas

Se verifica y valida el sistema a nivel de componentes individuales y su integración. Se busca descubrir cualquier defecto en los requisitos, análisis, diseño e implementación.

CAPÍTULO 6 DESARROLLO Y RESULTADOS

6.1 DESARROLLO

En este capítulo se aplicarán conceptos de los estudios presentados en los capítulos anteriores, con el propósito de diseñar e implementar de manera eficiente el sistema web. Se plantea el desarrollo de cada etapa del “Proceso de desarrollo de software con metodología orientada a objetos y UML” para obtener como resultado un software de calidad, escalable y fácil de usar para las personas.

6.1.1 Visión del proyecto (Estudio del problema)

Antes de empezar con la primera fase, se recomienda tener una recopilación de lo que se sabe sobre el proyecto, en el “*Documento de visión*”, este registra la visión general del sistema de software y documenta los riesgos y restricciones.

** Inicio Documento de visión**

Información Temperatura de la Superficie del Mar

El Instituto de Ciencias de Mar y Limnología de la UNAM, se conecta al Instituto de Geografía por medio del software TERASCAN para obtener imágenes satelitales de la Temperatura de la Superficie del Mar (TSM). Se cuentan con imágenes satelitales desde 1996 hasta el 2013 y se han utilizado los satélites NOAA-12, NOAA-14, NOAA-15, NOAA-16, NOAA-17, NOAA-18 y NOAA-19. Actualmente sólo funcionan los satélites 16, 18 y 19. Las imágenes son de 5 canales, 2 visibles y 3 infrarrojos (térmicos).

Se cuenta con imágenes diarias y se descargan con el siguiente formato:

ns1s0.yymmdd.hhmm

En donde *n* es para referirse a la serie de satélites NOAA, *s1s0* corresponden al número de satélite, *yy* al año, *mm* al mes, *dd* al día, *hh* a la hora y *mm* corresponden a los minutos.

Esta información es respecto a cuándo, a qué hora y con cuál satélite se obtuvo la imagen.

Estas imágenes después de ser descargadas, son comprimidas, y presentan el mismo formato que el anterior, sólo agregando al final .z o .gz: **ns1s0.yymmdd.hhmm.z / gz**

Estas imágenes se juntan por mes y así son almacenadas en uno o dos DVDs (dependiendo del tamaño que alcancen). Cada DVD es etiquetado con el nombre del mes, el año y si es volumen 1 o 2: **nombreMes_año_1 / nombreMes_año_2**

Posteriormente se hace un procesamiento digital de imagen para obtener el formato sst: **ns1s0.yymmdd.hhmm.sst** y de igual forma estas imágenes son comprimidas y almacenadas en DVDs: **ns1s0.yymmdd.hhmm.sst.z**

Después se hace otro procesamiento y se obtienen los promedios semanales de las temperaturas, que se comprenden en los siguientes periodos de días: del 1-7 corresponde a la semana 01, del 8-15 corresponde a la semana 02, del 16-22 corresponde a la semana 03, del 23-30/31 corresponde a la semana 04, y tienen el siguiente formato: **ns1s0.yymm_01.h_ph_pm_pm_p** y de igual forma son comprimidas y almacenadas en DVDs.

Finalmente se procesan estas imágenes para obtener promedios de temperatura mensuales con el siguiente formato: **ns1s0.yymm.h_ph_pm_pm_p** y son comprimidas y almacenadas en DVDs.

Para cada una de estas imágenes, se puede obtener su matriz de datos (datos en formato ASCII), estos datos son la TSM y deberán estar asociados con su ubicación geográfica. Esta matriz de datos está delimitada por un “máster” de la República Mexicana, incluyendo su zona que se delimita por las siguientes coordenadas: **124°W, 78°W latitud /32°N, 6°N longitud**, que comprende la zona económica exclusiva de México.

Cada dato está separado por un minuto lo que da una matriz de datos de 2760x1680. Las unidades de la TSM son grados Celsius (°C) y se utiliza la proyección cartográfica Universal Transversa de Mercator (UTM).

Objetivo

Desarrollar un sistema web que permita mostrar a través de un mapa interactivo la información de los promedios mensuales de la TSM de México de diferentes años y satélites. El mapa proporcionará las herramientas básicas de un Sistema de Información Geográfica (SIG) que le permitan al usuario realizar la interacción con el mapa.

El sistema estará conformado por un sitio web, que contendrá información sobre el proyecto, así como la aplicación para la consulta de la TSM, una galería interactiva de imágenes disponibles de la TSM, descarga de archivos y mecanismos para registro y administración de usuarios.

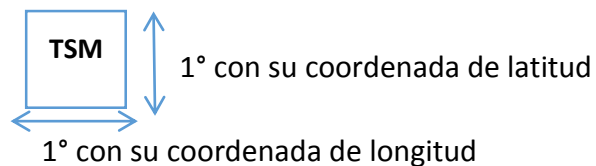
Oportunidades de negocio

Actualmente toda la información de la TSM está almacenada en DVDs. El Sistema de Visualización de la TSM ayudará en primer lugar a organizar toda esta información por medio de una base de datos, en segundo, pondrá esta información al alcance de diferentes usuarios que necesiten consultarla y en tercer lugar, a mostrar la información de una manera mucho más eficiente haciendo uso del gran poder de las aplicaciones de mapas web.

Descripción general del sistema

El sistema deberá tener una aplicación web que realizará las siguientes funciones:

1. Poder acceder a una base de datos espacial para recuperar la información consultada.
2. La consulta de la información de la TSM se llevará a cabo por medio del mapa interactivo y se deberá realizar de la siguiente manera:
 - ✓ El mapa web estará compuesto por una serie de celdas de que serán de $1^\circ \times 1^\circ$, cada grado tendrá su respectiva coordenada geográfica y por lo tanto cada celda contendrá su propia información de los promedios mensuales de la TSM de todos los satélites de los cuales se tenga información:



- ✓ Al hacer clic en cualquier de estos puntos se desplegará una tabla con toda la información correspondiente.
 - ✓ También se podrá seleccionar un conjunto de celdas y obtener información promedio de todas esas celdas.
3. Se deberá poder descargar el mapa en los formatos ascii y shapefile, para que puedan ser utilizado con algún Sistema de Información Geográfica (SIG)).
 4. El sistema web deberá incluir dentro del sitio web un visualizador de las imágenes satelitales de los diferentes años y satélites.
 5. El sistema tendrá un mecanismo de registro y administración de usuarios.
 6. El sistema podrá ser accedido por diferentes usuarios que se pueden dividir en dos grupos: usuarios de consulta (académicos) y usuarios de mantenimiento (administradores).

Restricciones

- ✓ El tiempo de respuesta del mapa sea considerablemente corto
- ✓ Los tipos de archivo que se descargarán deberán ser los originales ascii y archivos shapefiles.
- ✓ Se deberá utilizar un manejador de base de datos espacial.
- ✓ Se requiere que el mapa interactivo de la aplicación web, contenga los elementos fundamentales de los mapas de papel, tales como título, leyenda, escala, norte, proyección cartográfica, esto con el fin de que si se requiere imprimir el mapa, tenga la información necesaria para poder trabajar con él.

- ✓ Al ser un mapa interactivo web, se requiere que cumpla como tal, es decir, si se realiza un acercamiento, la escala debe coincidir con lo que se está visualizando en el mapa (debe de ser un mapa web, no una imagen de un mapa).
- ✓ Algunas partes del sistema sólo podrán ser accedidas sólo por algunos usuarios, por lo que se deberá tener un control de restricción para cada tipo de usuario.
- ✓ El sitio web se deberá poder visualizar en los principales navegadores.

Herramientas

Después de analizar diferentes situaciones en cuanto a presupuesto y recursos con los que cuenta el laboratorio, y ver que algunas opciones de código abierto son de tan buena calidad y potencia como las comerciales, se decidió trabajar con el siguiente software:

- ✓ Herramienta CASE “Enterprise Architect” para realizar los diagramas resultado del análisis.
- ✓ El GIS ArcGIS para el procesamiento de las imágenes.
- ✓ Power Designer para la creación del Diagrama E-R
- ✓ OpenGeo Suite 4.1 para toda la arquitectura de la aplicación WebMapping y que contiene los siguientes elementos: PostGreSQL/PostGIS (SMBD y SMBD con componente espacial), GeoServer (servidor de mapas), GeoWebCache (para los mosaicos) y OpenLayers (para la creación y publicación de mapas en la web)

Fin Documento de visión

6.1.2 Determinación de Requerimientos

Al realizar la determinación de requerimientos se debe registrar toda la información en el documento “*Especificación de Requerimientos del Sistema*” (SRS por sus siglas en inglés). Una vez teniendo esta información se debe proceder a realizar un **diagrama de casos de uso**, una descripción de los **escenarios principales** de los casos de uso y las **formas de casos de uso**. Se puede utilizar el **diagrama de actividades** para validar el de casos de uso.

6.1.2.1 Documento SRS

El documento SRS (Especificación de Requerimientos de Software), es la declaración oficial de qué deben implementar los desarrolladores del sistema. Debe incluir tanto los requerimientos del usuario para el sistema (FRs), como una especificación detallada de los requerimientos del sistema (NFRs) (Sommerville, 2005).

El documento SRS completo se encuentra en el anexo A.

6.1.2.2 Diagrama de casos de uso

Presenta la interacción que existe entre los diferentes tipos de usuario (usuario general, académico y administrador) y el sistema Web de Consulta de TSM. En el diagrama se puede apreciar que el caso de uso “E2: Interactuar con mapa web” se amplió a varios casos de uso especializados (E2a, E2b, etc.) para tener una mayor claridad, de la misma manera, los casos de uso “E1b” y “A1b” utilizan el tipo de asociación especial <<extend>> ya que los dos son variaciones del caso de uso base del que proceden (E1 y A1), mientras que “E3” y “E4” se asocian a través de un <<include>> que significa que “E4” dependerá del resultado de “E3”, es decir, si el usuario está registrado podrá descargar archivos (figura 6.1):

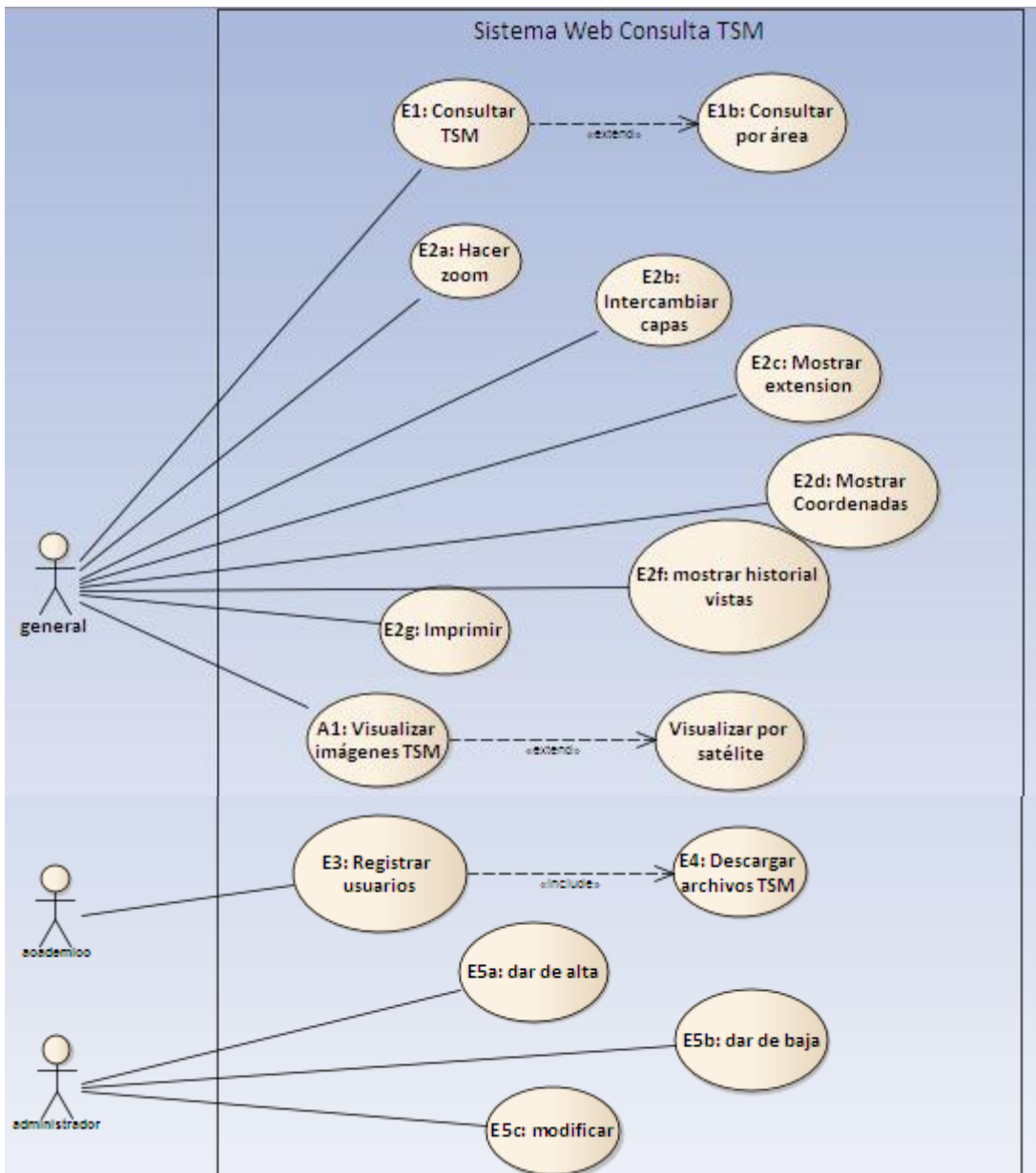


Figura 6.1 Diagrama de Casos de Uso

6.1.2.3 Escenarios principales de los casos de uso

Los escenarios describen como es que se va a implementar un caso de uso en una determinada situación, pueden existir diferentes escenarios para un mismo caso de uso que representarían los diferentes caminos por los que podría desarrollarse el caso de uso, los escenarios principales presentan el camino con una salida exitosa.

Nota: en estos escenarios utilizaremos la palabra “usuario” para referirnos a los tres tipos de usuario: general, académico y administrador, sólo cuando sea necesario se especificará que tipo de usuario es con el que se está trabajando.

E1: Consultar TSM

El usuario mediante su navegador, ingresa al sitio web y se encuentra con la página de bienvenida, como desea consultar la información de la TSM, a través del menú de navegación del sitio web presiona el botón (link que redirecciona a otra página del sitio web) de appTSM y se redirecciona a la página web donde se encuentra la aplicación appTSM; la aplicación despliega el mapa base de la República Mexicana y la capa por default corresponde a TSM_2013_diciembre, que es la última imagen obtenida del satélite.

El usuario desea consultar información sobre la TSM así que mediante su mouse se posiciona sobre el área del mapa que desea y hace clic (oprimir botón izquierdo del mouse), al hacerlo se despliega la tabla que muestra la información concerniente a esa punto. El usuario cierra esa tabla y consulta la información de otros puntos.

El usuario finaliza su visita por el sitio web y cierra la página.

E2: Interactuar con la appTSM

El usuario se encuentra dentro de la aplicación TSM e inspecciona las herramientas para interactuar con el mapa que tiene a su disposición.

Elije la herramienta “acercamiento” para poder trabajar con más detalle sobre una zona específica, utiliza la herramienta desplazamiento para poder moverse a través del mapa, como ha hecho un acercamiento muy alto, utiliza la herramienta “máxima extensión” para poder volver al tamaño original del mapa, después el usuario desea consultar los mapas de otros años, por lo que utiliza la herramienta de “intercambio de capas” para poder seleccionar la capa con el año que desea, utiliza la herramienta “información” para poder consultar la información de la TSM del punto que desea y sigue utilizando las herramientas que tiene a su disposición.

El usuario finaliza su visita por el sitio web y cierra la página.

E3: Registrar usuario

El usuario ingresa a la página inicial del sitio web, como es un usuario académico decide registrarse en el sistema, presiona el botón “Registrarse” y es dirigido a una nueva página de “Registro”.

En esta página, el usuario rellena el formulario con su nombre, apellido, correo electrónico, nombre de usuario, contraseña y su solicitud para ser dado de alta en el sistema, al terminar presionará el botón “Enviar” y sus datos serán almacenados en la base de datos y en la página saldrá un anuncio de que su registro fue completado con éxito y que se le enviará un correo para notificarle cuando haya sido dado de alta en el sistema.

El usuario presiona el botón “Regresar a la página principal” y con eso termina el proceso de registro.

E4: Descarga de TSM

El usuario se encuentra haciendo consultas dentro de la aplicación appTSM, al finalizar, como necesita esa información, decide descargarla. Ya que es un usuario académico y ya ha realizado previamente su registro, tiene la posibilidad de hacer descargas.

El usuario académico elige la herramienta “descargar” que abre una ventana nueva con una lista de los archivos que están disponibles para descargar, el usuario elige el que necesita y selecciona el formato que más le conviene.

Cierra la ventana de descarga y regresa a la pantalla de la appTSM y sigue realizando consultas e interactuando con el mapa. El usuario finaliza su visita al sistema, cierra su sesión y sale del sitio web.

E5: Administrar usuarios

El administrador del sistema se encuentra trabajando en su computadora cuando recibe una mail con la notificación de que un usuario se ha registrado en el sistema y pide ser dado de alta para poder realizar descargas.

El administrador abre la página de “Administración de usuarios” e inicia sesión, se despliega la página con los datos de las personas registradas, analiza la información del usuario que pidió ser registrado y decide darlo de alta, por lo que presiona el botón “Alta”. Un mail es enviado al usuario que solicitó darse de alta comunicándole que ya puede iniciar sesión en el sistema.

El administrador cierra la página de gestión de usuarios y continúa con su trabajo.

A1: Visualización TSM

El usuario ingresa a la página principal del sitio web y ve la pantalla de bienvenida y como decide ver las imágenes de la TSM, mediante el menú de navegación pulsa el botón “Galería TSM” y se redirecciona a la página web “Galería TSM”, el usuario ve la imagen que está por defecto, que es la más antigua.

El usuario desea visualizar las demás imágenes de ese año así que utiliza los botones de “meses” para navegar dentro de las imágenes. Al finalizar, el usuario desea ver las imágenes de otros años, así que mediante el menú “años”, el usuario elige el año y se despliega en la pantalla la primer imagen de ese año con su respectivo menú de meses y el usuario sigue navegando a través de las imágenes.

El usuario finaliza su visita por el sitio web y cierra la página.

6.1.2.4 Formas de casos de uso

Nos proporcionan una descripción mucho más detallada de los casos de uso y elementos que nos permiten visualizar aspectos importantes que en el diagrama de casos de uso y en la descripción de los escenarios, tal vez no sean tan fáciles de identificar.

Elemento	Descripción
ID y nombre del caso de uso	E1: consultar TSM
Descripción	Consultar información del mapa web
Actores	Usuario, usuario académico
Prioridad	Esencial
Riesgo	
Pre-condiciones (y suposiciones)	El usuario ha ingresado al sitio web y ha decidido ingresar a la appTSM
Disparador	El usuario hace clic sobre el mapa para obtener la información
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario hace clic sobre un punto 2. La información se muestra en una tabla 3. El usuario cierra la tabla
Flujos alternos	En el paso 1 usuario podría seleccionar un área en lugar de solo un punto
Post-condiciones	
Requerimientos no funcionales	E1-101

Elemento	Descripción
ID y nombre del caso de uso	E2: interactuar con la appTSM
Descripción	Interacción con el mapa web
Actores	Usuario, usuario académico
Prioridad	Esencial
Riesgo	
Pre-condiciones (y suposiciones)	El usuario ha accedido a la aplicación
Disparador	Selección de una herramienta de interacción
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario selecciona una herramienta 2. El usuario hace uso de esa herramienta
Flujos alternos	No hay flujos alternos
Post-condiciones	Hay una interactividad del usuario con el mapa
Requerimientos no funcionales	E1-102

Elemento	Descripción
ID y nombre del caso de uso	E4: descargar archivos
Descripción	Descargar los archivos disponibles que hay sobre la TSM
Actores	Usuario académico
Prioridad	Esencial
Riesgo	
Pre-condiciones (y suposiciones)	El usuario ha iniciado sesión en el sistema. El usuario ha accedido a la appTSM y puede que haya hecho consultas.
Disparador	Selección de la herramienta “descarga”
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario entra a la appTSM 2. Presiona el botón “Descargas” 3. Se abre una nueva ventana con los archivos disponibles 4. El usuario elige el que desea descargar 5. Inicia la descarga 6. El usuario cierra la venta de descargas y regresa a la appTSM
Flujos alternos	En el paso 2 si el usuario no ha sido registrado en el sistema, se le mostrará un mensaje para que se registre en el sistema.
Post-condiciones	
Requerimientos no funcionales	E3-101

Elemento	Descripción
ID y nombre del caso de uso	A1: visualizar imágenes TSM
Descripción	A través de una galería interactiva de imágenes, se podrán visualizar las imágenes disponibles de la TSM.
Actores	Usuario
Prioridad	Alta
Riesgo	
Pre-condiciones (y suposiciones)	El usuario ha ingresado a la página web donde se encuentra la galería de imágenes.
Disparador	El usuario presionó el botón "Galería TSM" del menú del sitio web.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario ingresa a la galería 2. Ve la primer imagen (imagen por default) 3. A través de los botones de meses, navega a través de las imágenes de ese año (año por default) 4. Elige otro año y sigue navegando a través de las imágenes 5. Cierra la página web
Flujos alternos	En el paso 4, puede que elija en lugar de otro año, la visualización a través de satélites.
Post-condiciones	
Requerimientos no funcionales	A1-101

Elemento	Descripción
ID y nombre del caso de uso	E3: Registrar usuarios
Descripción	Los usuarios se podrán registrar en el sistema para obtener más contenido.
Actores	Usuario, usuario académico.
Prioridad	Esencial
Riesgo	
Pre-condiciones (y suposiciones)	El usuario se da cuenta de que registrándose en el sistema puede tener acceso a más contenido.
Disparador	El usuario presiona el botón "Registrarse".
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario presiona el botón "Registrarse"

	<ol style="list-style-type: none"> 2. Se abre una nueva página con el formulario de registro 3. El usuario ingresa todos sus datos y envía la información. 4. Se presenta un mensaje diciendo que su registro fue completado con éxito.
Flujos alternos	No hay flujos alternos
Post-condiciones	El usuario puede iniciar sesión
Requerimientos no funcionales	

6.1.2.5 Diagrama de actividades

Para validar el diagrama de casos de uso se utiliza el **diagrama de actividades** que se muestra en la figura 6.2:

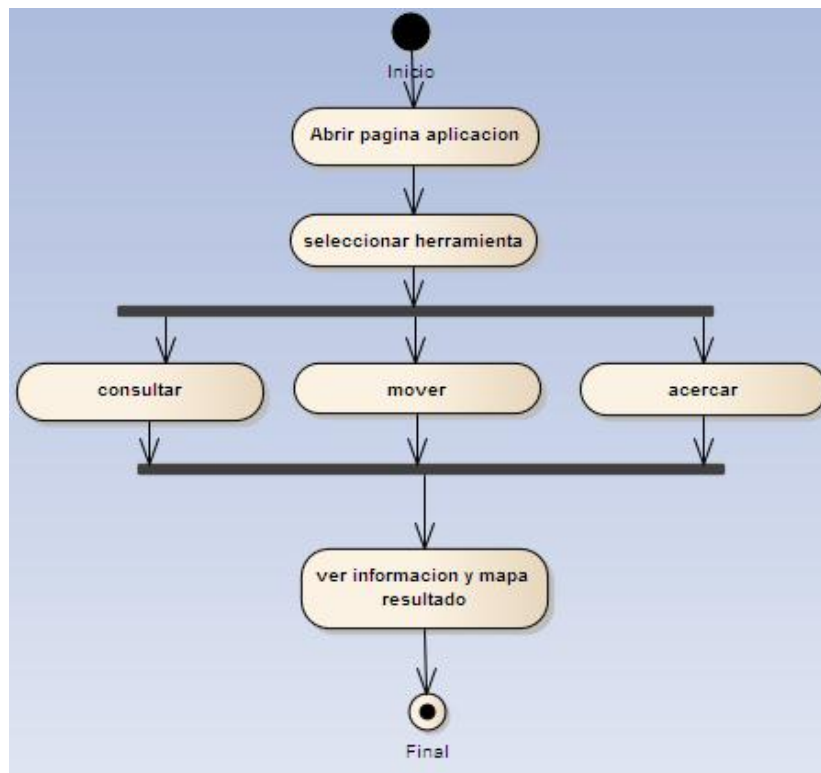


Figura 6.2 Diagrama de actividades del caso de uso "E1 Consultar TSM"

6.1.3 Análisis

El análisis parte de una definición del problema, casi siempre imprecisa y genera una descripción precisa y exacta del problema y describe a éste mediante clases y objetos. Se utilizan las **abstracciones clave**, las **tarjetas CRS**, así como el **diagrama de clases** y el **diagrama de objetos** para validarlo.

6.1.3.1 Abstracciones clave

Una abstracción clave es una clase u objeto que forma parte del vocabulario del dominio del problema. Representa los objetos principales en el sistema:

Candidato a abstracción clave	Causa de eliminación	Nombre seleccionado
Mapa interactivo		Mapa
Información TSM		InformaciónTSM
Consulta		Consulta
Satélites	No es relevante	
Base de datos espacial	No es relevante	
Imágenes TSM		ImagenTSM
Administrador		Administrador
Usuario General		Usuario
Herramientas		Herramienta
Archivo ASCII		Archivo
Archivo shapefile	Se junto con archivo ASCII como una sola clase	
Páginas web		PáginaWeb
Administración		Administracion

6.1.3.2 Tarjetas CRS

Es una técnica usual para identificar abstracciones clave. Las responsabilidades son atributos, operaciones o especificaciones del rango de datos de los valores de los atributos. Los colaboradores son otros objetos con los cuales el objeto candidato a Abstracción Clave está asociado:

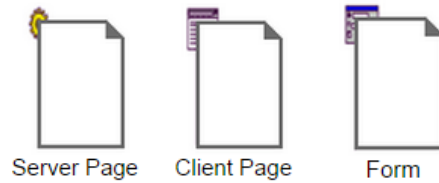
Usuario	
Hacer consultas, visualizaciones y registrarse	Mapa interactivo Información TSM Registro

Herramienta	
Proporcionar interactividad	Usuario Mapa interactivo

El resto de las tarjetas CRS se encuentran en el anexo B.

6.1.3.3 Diagrama de Clases

Las clases del diagrama, son las Abstracciones Clave del Sistema que se obtuvieron anteriormente. Este diagrama hace uso de los estereotipos para aplicaciones web con sus respectivos símbolos:



Se puede observar en el diagrama, que las clases “PaginaWeb” e “ImagenTSM” tienen el estereotipo “Client page”, mientras que “Mapa” es una “Server page”, esta clase se asocia con las interfaces (tienen la característica de ser clases que sólo tienen atributos y pueden ser reutilizados en un futuro) “Herramienta” y “Consulta” mediante la realización. Por otro lado, las clases “Registro” y “Login” (que tienen el estereotipo “Form”) se asocian con sus respectivas clases a través de “Submit”, es decir la información que recojan, la enviarán a las “Server pages” que la procesarán.

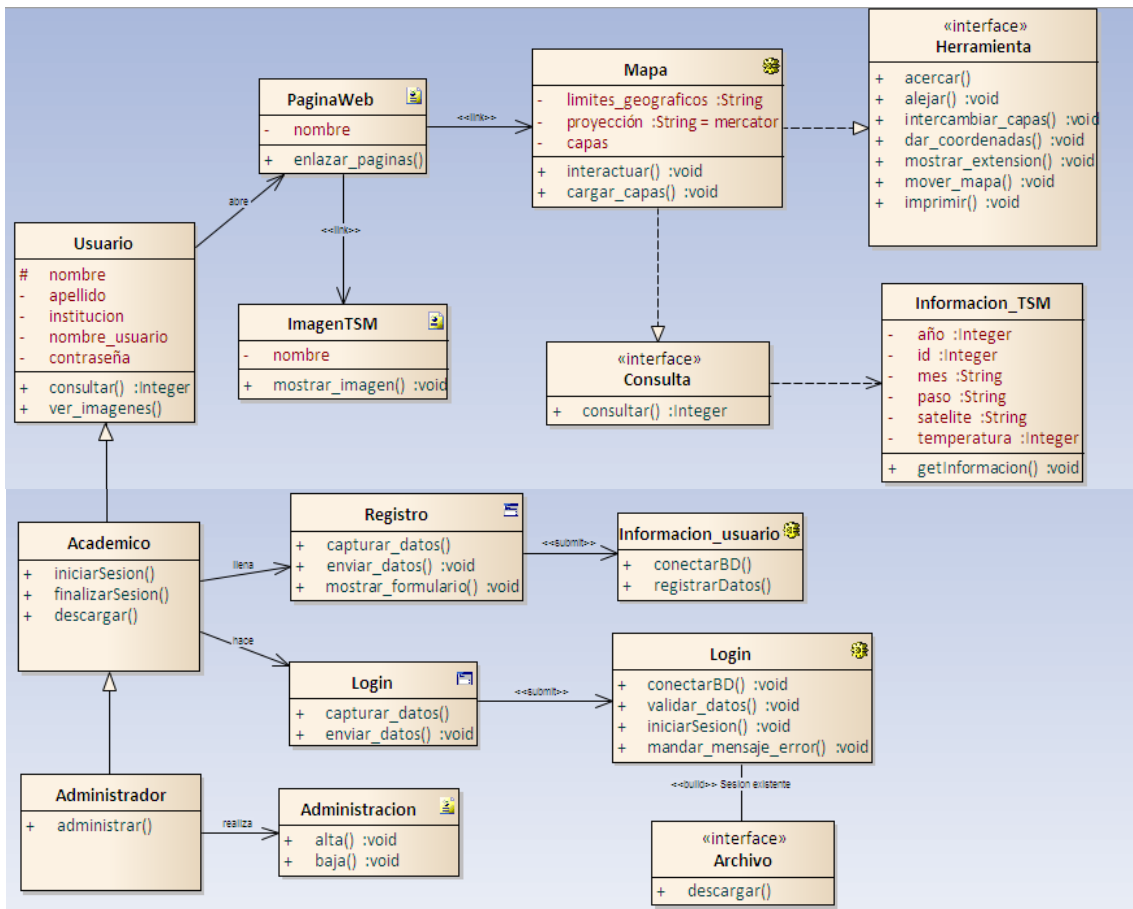


Figura 6.3 Diagrama de Clases perteneciente a la fase de análisis

6.1.3.4 Diagrama de Objetos

Para validar el diagrama de clases se utiliza el diagrama de objetos, que muestra al sistema en un momento dado, en la figura 6.4 se muestran los objetos creados a partir de las clases, con los atributos con posibles valores, por lo que se toman uno o más escenarios de los casos de uso (figura 6.4):

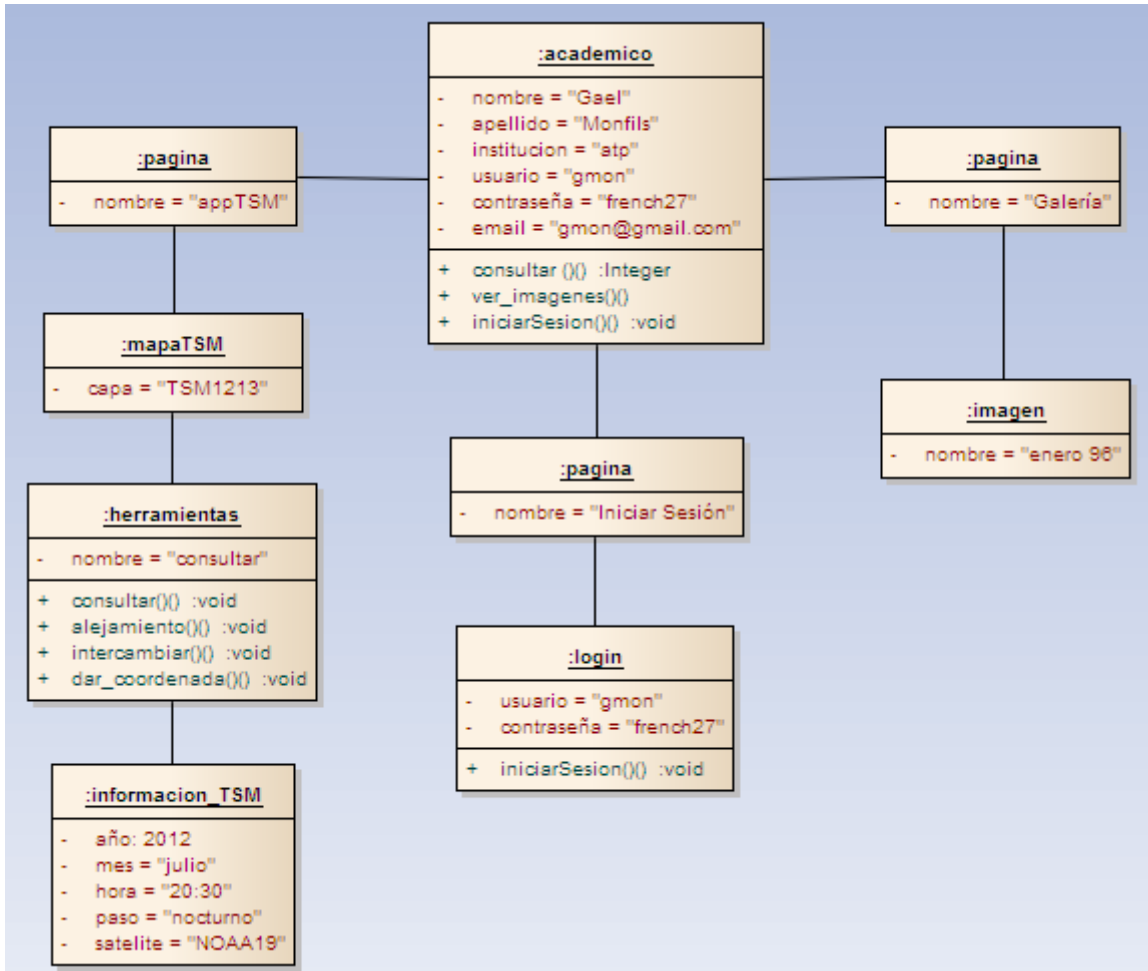


Figura 6.4 Diagrama de objetos

6.1.3.5 Modelo de datos (Modelo E-R)

En esta etapa se elabora el modelo E-R, obteniendo como producto final el diagrama Entidad-Relación que será el esquema de la base de datos.

Después de analizar detalladamente los requerimientos, se identificaron las entidades, sus relaciones, la cardinalidad, se especificaron para todos los atributos su tipo de dato, su rango (sólo a los que eran necesarios), sus valores no nulos, sus claves primarias, se tomó en cuenta la redundancia mínima y la consistencia; con todo lo anterior se obtuvo el

siguiente Diagrama E-R (figura 6.5), que cumple correctamente con la notación DER y con la integridad de datos:

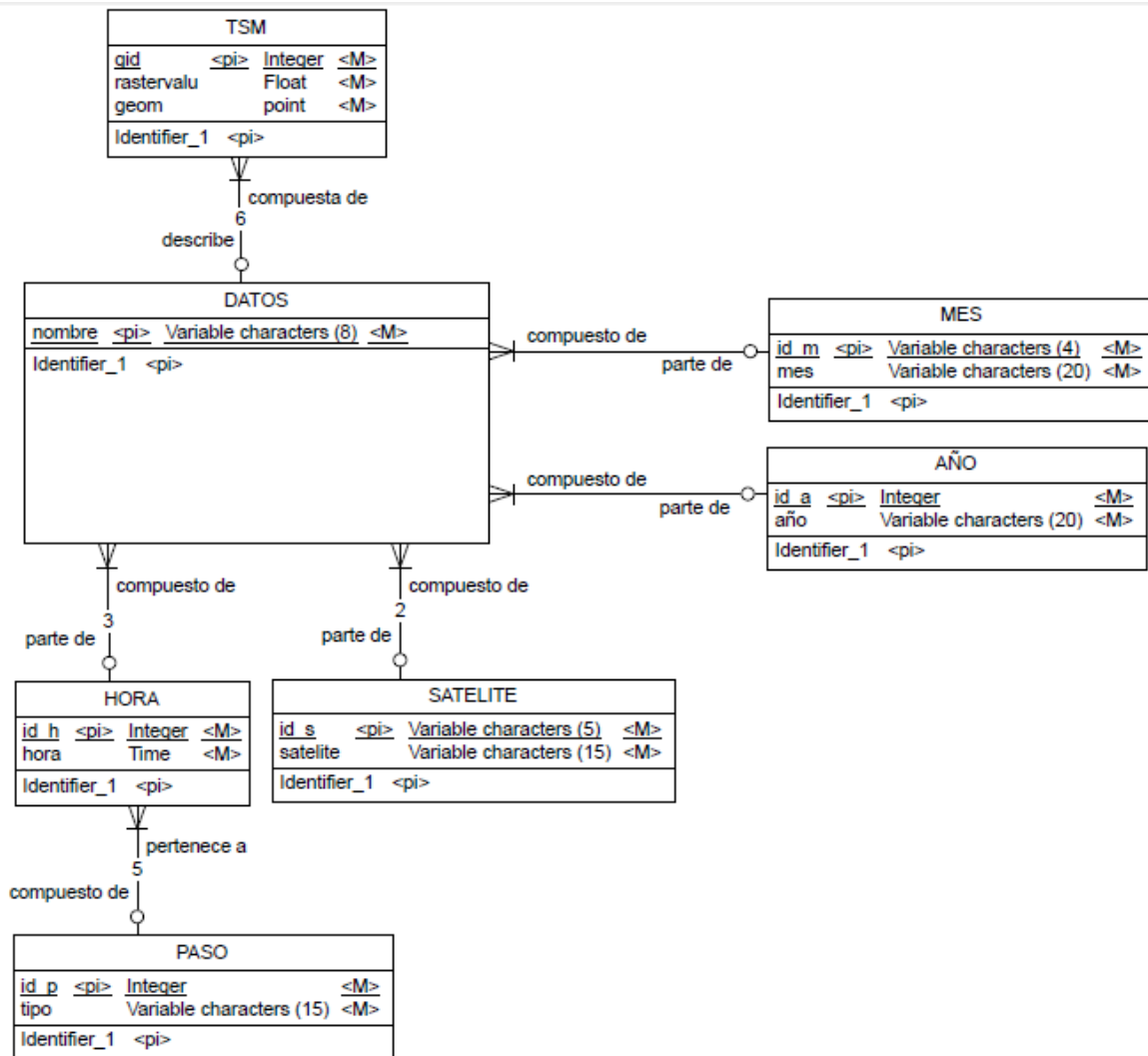


Figura 6.5 Diagrama Entidad-Relación

6.1.4 Diseño

En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requerimientos. Crea un punto de partida para las actividades de implementación (Olguín, 2004).

Utiliza los resultados del análisis y genera un modelo de software que detalla cómo debe hacer las cosas el sistema para cumplir los requisitos y solucionar el problema. Utiliza los **diagramas de colaboración**, los **diagramas de secuencias** de UML, así como el **modelo relacional** para la base de datos.

6.1.4.1 Diagramas de Colaboración

Muestran la interacción entre los objetos del sistema, los principales diagramas de colaboración se representan en la figura 6.6 y 6.7:

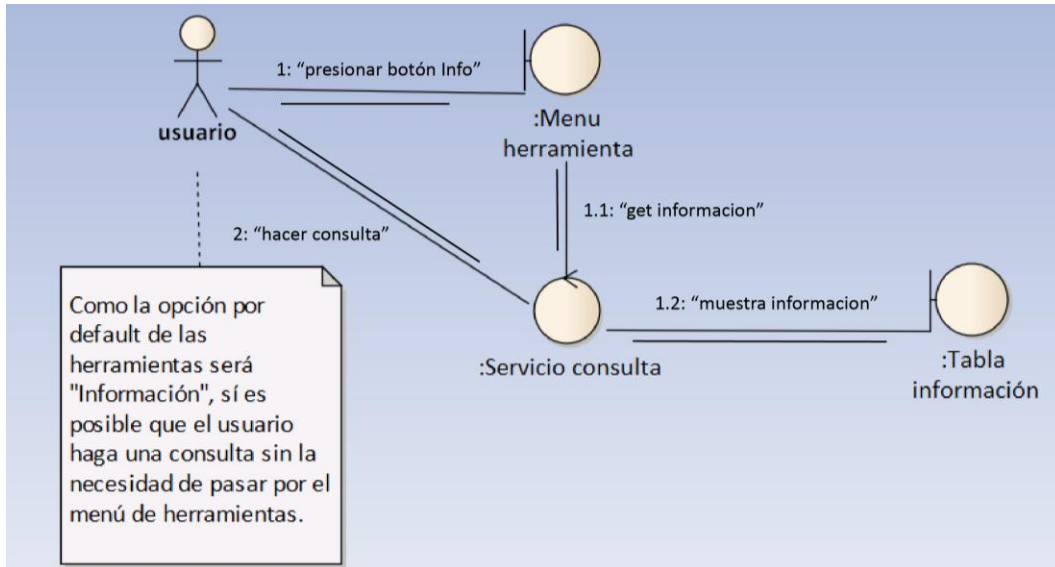


Figura 6.6 Diagrama de colaboración "E1 Consultar TSM"

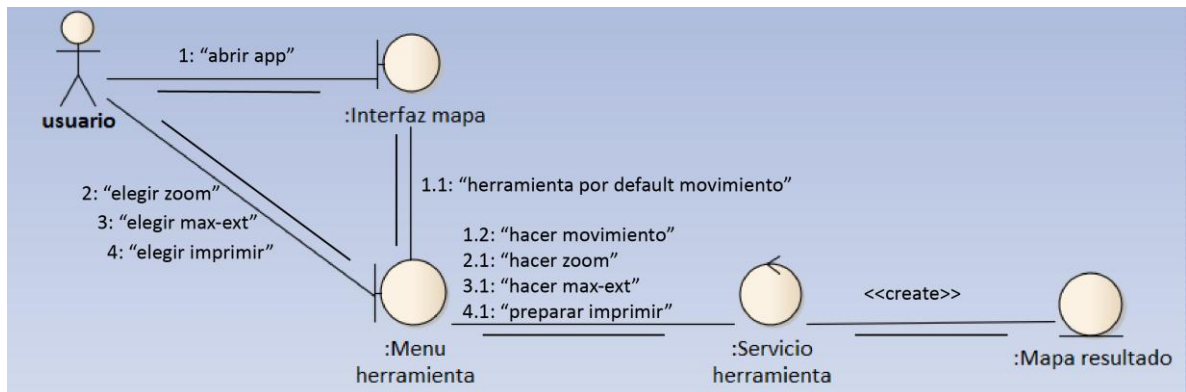


Figura 6.7 Diagrama de colaboración "E2 Interactuar con mapa web"

6.1.4.2 Diagramas de Secuencia

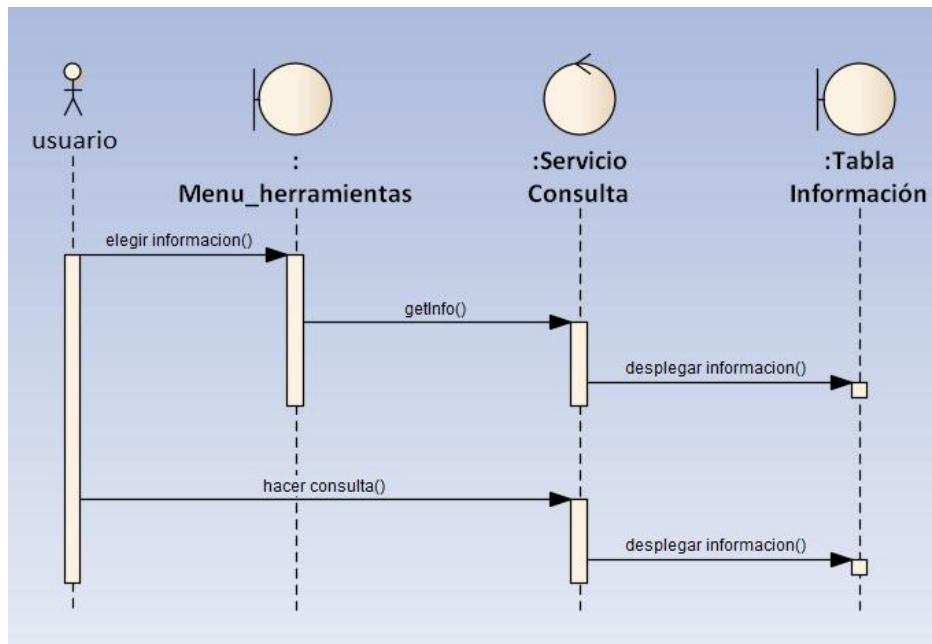


Figura 6.8 Diagrama de secuencias "E1 Consultar"

6.1.4.3 Modelo Relacional

Quando se diseña, se toman los requerimientos detallados del negocio y se busca la mejor manera de utilizar el ambiente tecnológico para cumplirlos. Diseñar una base de datos involucra tomar un modelo de datos (modelo Entidad-Relación), el cual proporciona una vista de los requerimientos de información del negocio, y transformarlo en una representación de software práctica:

1. Mapear entidades simples a tablas
2. Mapear atributos a columnas
3. Mapear identificadores únicos a llaves primarias
4. Mapear relaciones a llaves foráneas

6.1.5 Arquitectura

Es la estructura jerárquica de los componentes del programa (clientes, servidores, bases de datos), la manera en que interactúan y la estructura de datos que van a utilizar (Olguín, 2004).

6.1.5.1 Diagrama de Componentes

En los diagramas de componentes se utilizan `<<interface>>`, una interfaz es un conjunto de operaciones que especifica algo respecto al comportamiento de una clase. Se podrá reutilizar este conjunto de operaciones de clase en clase.

En el diagrama se utilizan las interfaces *formulario*, que es el conjunto de operaciones para obtener información de los usuarios a través de campos (servirá para crear el formulario de registro y de login), la interfaz *conexión*, que proporciona las operaciones para recibir la información de los formularios y hacer la conexión con la base de datos.

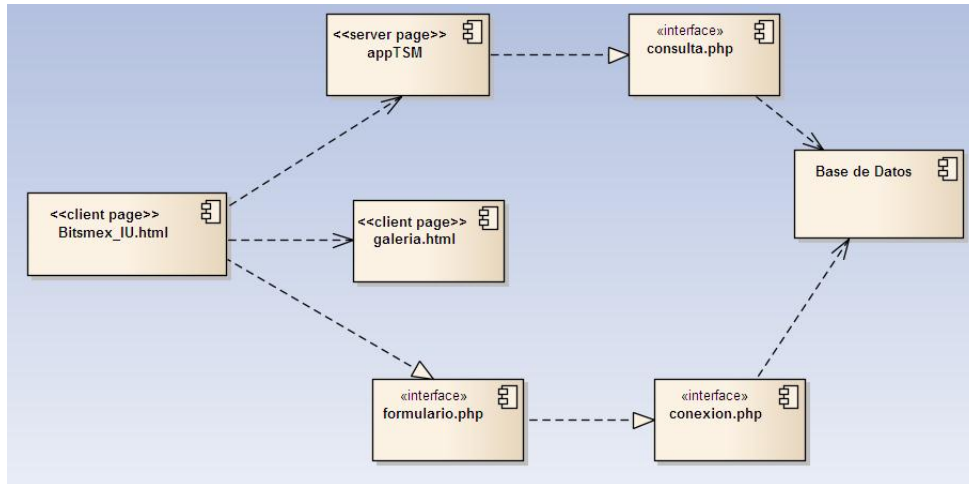


Figura 6.9 Diagrama de componentes

6.1.5.2 Diagrama de Distribución

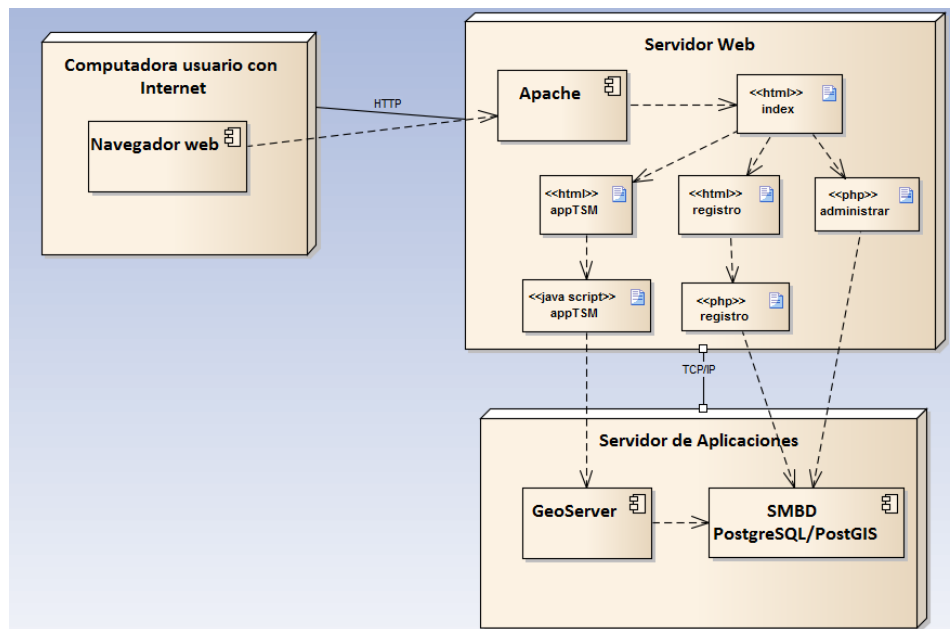


Figura 6.10 Diagrama de distribución

En segundo lugar se tuvo que agregar un encabezado a cada archivo. Éste contiene un conjunto de palabras clave, seguida de los valores de las celdas en un orden mayor de filas. El formato del archivo en general es (ESRI, 2013):

```

NCOLS XXX      NROWS XXX      XLLCORNER XXX YLLCORNER XXX      CELSIZE XXX
NODATA_VALUE XXX      row 1 row 2 .      .      row n
    
```

Ya teniendo los datos listos para trabajar, se hicieron los siguientes procesos con ArcGIS:

- ✓ *Espacio de trabajo:* se definió la proyección cartográfica “**GCS_WGS_1984 WKID: 4326 Authority: EPSG**” y que muestre en unidades “Degrees, minutes, seconds”.
- ✓ *ASCII to Raster:* Convierte un archivo ASCII que representa datos ráster en un dataset ráster (ESRI, 2013). Debido a que el archivo ASCII sólo contenía números, se transformó en ráster para poder visualizarlo como imagen en ArcGIS y poder trabajar con él (figura 6.12):

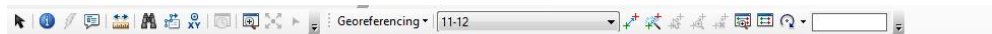


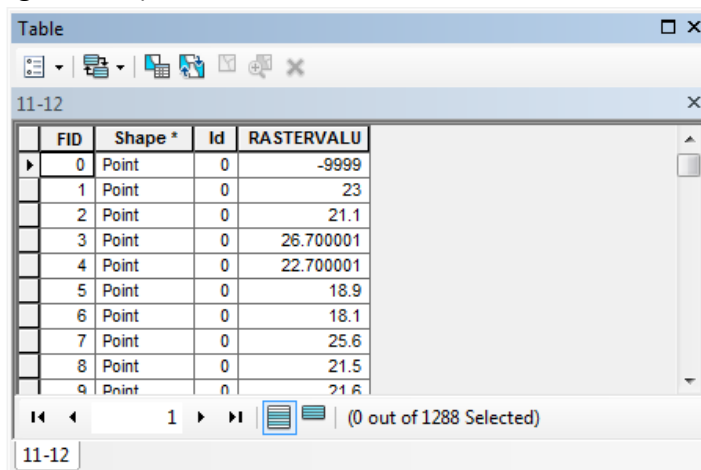
Figura 6.12 Con la herramienta ASCII to Ráster, fue posible visualizar un archivo que sólo contenía valores de números a una imagen

- ✓ *Layer Properties:* en este apartado se especificaron las coordenadas de los vértices de la imagen para poder georreferenciarla (figura 6.13), de esta manera el archivo ráster creado anteriormente queda georreferenciado y con su proyección cartográfica.

Left: Top: Right:
 Bottom:

Figura 6.13 Se establecen los valores del marco del mapa en coordenadas geográficas

- ✓ *Create Fishnet*: ésta herramienta sirve para crear una malla de puntos, que se utilizará en el siguiente punto.
- ✓ *Extract Values to Point*: Extrae los valores de celda basados en un conjunto de entidades de punto y registra los valores en la tabla de atributos de una clase de entidad de salida. Todos los campos de la clase de entidad de punto de entrada se incluirán en la clase de entidad de punto de salida. La clase de entidad de salida tendrá un nuevo campo agregado llamado RASTERVALU (ESRI, 2013).
Las “entidades de punto” es la malla de puntos creada anteriormente, con esta herramienta, se extraen los valores de temperatura del archivo raster y se guardan en formato shapefile, éste último archivo, al ser vectorial ya cuenta con un ADT (punto con su coordenada geográfica), su proyección cartográfica y el valor de la temperatura (figura 6.14):



FID	Shape *	Id	RASTERVALU
0	Point	0	-9999
1	Point	0	23
2	Point	0	21.1
3	Point	0	26.700001
4	Point	0	22.700001
5	Point	0	18.9
6	Point	0	18.1
7	Point	0	25.6
8	Point	0	21.5
9	Point	0	21.6

Figura 6.14 Valores del archivo shapefile

Estos archivos shapefile ya están listos para poder ingresarlos a la base de datos espacial y poder visualizarlos en un mapa, puesto que están georreferenciados.

LA BASE DE DATOS ESPACIAL

PostgreSQL tiene varios componentes de administración *front-end*, esto quiere decir que interactúa con el usuario de manera amigable (Morales, 2014):

- *psql*: es una herramienta de línea de comandos, permite escribir consultas de forma interactiva, enviarlas a PostgreSQL, y ver los resultados de la consulta. Además, proporciona una serie de comandos y varias características para facilitar la escritura de guiones y la automatización de una gran variedad de tareas (PostgreSQL, 2013).
- *pgAdmin*: es la plataforma más popular y rica en características, para la administración y desarrollo para PostgreSQL (pgAdmin, 2013).

Para la conexión, creación y administración de la base de datos espacial, se utilizó pgAdmin (todas las consultas que se pueden hacer en pgAdmin también se pueden hacer en línea de comandos con psql) (Morales, 2014).

- ✓ Primero se creó una nueva conexión.
- ✓ PostgreSQL es SDBD relacional, para trabajar con datos espaciales es necesario añadir extensión espacial PostGIS, con la siguiente sentencia sql:

```
CREATE EXTENSION postgis;
```

A través del árbol de objetos que nos proporciona pgAdmin, podemos ver todas las funcionalidades que la extensión espacial que acabamos de crear nos proporciona, dentro de nuestra base de datos y nuestro esquema, se crearon las **funciones espaciales** (que nos permiten trabajar con los objetos geográficos), las vistas **geometry_columns** y **geography_columns** y dentro de las tablas, una tabla llamada **spatial_ref_sys**.

Ésta última, es una tabla de meta-datos de la especificación OpenGIS, contiene un identificador numérico (**SRID**) y una descripción textual del sistema de coordenadas espacial de la base de datos (Ramsey, 2013).

Esquema lógico de PostgreSQL, dentro de este se encuentra el “Cluster de bases de datos”, que es un repositorio que engloba un conjunto de bases de datos, que contienen objetos que se pueden guardar en distintos tablespaces y un conjunto de usuarios que se conectan al cluster (Alarcón, 2007).

Así se tienen los tres elementos principales a nivel lógico en un cluster: **bases de datos** (engloba un conjunto de esquemas, en estos es donde se crean los objetos (tablas, índices, procedimientos, vistas, etc.), **tablespaces** (ubicaciones alternativas a la que por defecto tiene el cluster) y **roles** (engloba el concepto de usuarios y grupos de permisos (Alarcón, 2007)

- ✓ Se dejó el cluster que PostgreSQL creó por default, y sobre este se creó un nuevo rol (bitsmex), un tablespace (baseBitsmex, y su ubicación se encuentra en otro disco con mayor capacidad de almacenamiento) y la base de datos (llamada bitsmexTSM, que pertenece al rol “bitsmex” y se encuentra en el tablespace “baseBitsmex”):

```
-- Tablespace: baseBitsmex
-- DROP TABLESPACE "baseBitsmex"
CREATE TABLESPACE "baseBitsmex"
  OWNER bitsmex
  LOCATION 'D:\\baseBitsmex';
```

```
-- Database: "bitsmexTSM"

-- DROP DATABASE "bitsmexTSM";

CREATE DATABASE "bitsmexTSM"
  WITH OWNER = bitsmex
       ENCODING = 'UTF8'
       TABLESPACE = "baseBitsmex"
       LC_COLLATE = 'Spanish_Mexico.1252'
       CONNECTION LIMIT = -1;
```

- ✓ El siguiente paso fue la creación de las tablas de atributos alfanuméricos que contienen la información relacionada a la obtención de las imágenes satelitales, para lo cual se utilizó el lenguaje de definición de datos, mientras que para asegurar la integridad y seguridad se utilizó el lenguaje de control de datos (que en ambos casos se refiere al SQL). Se muestra aquí parte del código SQL, que muestra la creación de la tabla DATOS, sus constraints (llave primaria, llaves foráneas, NOT NULL) y la integridad referencial:

```
-- Table: datos

-- DROP TABLE datos;

CREATE TABLE datos
(
  nombre character varying NOT NULL,
  mes character varying,
  "año" character varying,
  hora character varying,
  satellite character varying,
  CONSTRAINT id_nombre PRIMARY KEY (nombre),
  CONSTRAINT "id_año" FOREIGN KEY ("año")
  REFERENCES "año" (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT id_hora FOREIGN KEY (hora)
  REFERENCES hora (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT id_mes FOREIGN KEY (mes)
  REFERENCES mes (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT id_satellite FOREIGN KEY (satellite)
  REFERENCES satellite (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE datos
OWNER TO postgres;
```

- ✓ Por otro lado, debido a que nuestros datos de la TSM (los datos geográficos) están en un formato shapefile se utilizó una herramienta que es parte de OpenGeo Suite llamada pgShapeLoader que hace una conexión a la base de datos y permitió cargar los archivos shapefile con todos sus atributos a la base de datos como tablas, indicarle el número de SRID y la creación de los índices espaciales.

Primero, crea la tabla con sus columnas, haciendo énfasis a la columna **geom** que es la que permite almacenar los datos geográficos, en ésta se especificó el *tipo de geometría* (Point, que es la manera en que se representa a cada temperatura) y el **SRID (4326)**, que corresponde al sistema de referencia **GCS_WGS_1984**.

La columna **rastervalu** es la que contiene los valores de la temperatura, se creó también un id incremental y los constraints:

```
-- Table: t112

-- DROP TABLE t112;

CREATE TABLE t112
(
  gid integer NOT NULL DEFAULT nextval('"1-12_gid_seq"'::regclass),
  rastervalu numeric,
  geom geometry(Point,4326),
  nombre character varying,
  CONSTRAINT "1-12_pkey" PRIMARY KEY (gid),
  CONSTRAINT n112 FOREIGN KEY (nombre)
    REFERENCES datos (nombre) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
ALTER TABLE t112
  OWNER TO postgres;

-- Index: "1-12_geom_gist"

-- DROP INDEX "1-12_geom_gist";

CREATE INDEX "1-12_geom_gist"
  ON t112
  USING gist
  (geom);
```

EL SERVIDOR DE MAPAS

Siguiendo en la arquitectura OpenGeo Suite, el servidor de mapas GeoServer, incluye una interfaz de administración Web a través de la que podemos hacer la mayor parte de las configuraciones de datos y servicios (Morales, 2014). Se realizaron las siguientes actividades para poder pasar de las tablas TSM de la base de datos a un formato gráfico (mapa) que pueda ser compartido a través de internet siguiendo los estándares de la OGC:

- ✓ Espacio de trabajo: sirven para agrupar datos de un mismo proyecto, todas las capas en GeoServer deben estar asociadas a uno, cada espacio de trabajo lleva un **nombre** y un **URI** (Uniform Resource Identifier) que es similar a una dirección URL, pero no necesita apuntar a un sitio web; se creó un espacio de trabajo llamado “**bitsmex**” y su URI “**http://geoserver.org/bitsmex**”.
- ✓ **Conexión con la base de datos:** la conexión se realiza a través de “Import Data”, este despliega una lista de los SDBD con los cuales se puede conectar, entre ellos PostGIS, cuando se selecciona, se llenan los parámetros para poder hacer la conexión (host, port, database, schema, username, password).
- ✓ **Carga de datos:** si la conexión se realiza con éxito, se despliega en la pantalla una lista con todas las tablas que posee la base de datos. El siguiente paso es seleccionar las tablas (datos geográficos) deseadas e iniciar la importación. Cuando están listas, es posible visualizarlas en el mismo navegador, como si fuera un SIG (en este punto las tablas ya pueden ser llamadas capas).
- ✓ **Estilos:** los datos geográficos no tienen instrucciones propias para su visualización. Por lo tanto la información adicional, en forma de un estilo, tiene que ser aplicada a los datos para que sean visualizados de una manera específica. GeoServer utiliza el lenguaje de marcado *Styled Layer Descriptor (SLD)* para describir los datos geoespaciales. SLD es un estándar basado en XML que ha sido creado por la OGC (Morales, 2014).

Un archivo SLD contiene la siguiente estructura jerarquizada:

- * Header
 - * FeatureTypeStyles
 - *Rules
 - *Symbolizers

El Header contiene metadatos y es generalmente idéntico en todos los SLD.

Un FeatureTypeStyle es un conjunto de reglas de estilo.

Un Rule es una directiva única de estilo.

Un Symbolizer es la instrucción de estilo actual (Morales, 2014).

Se utilizó SLD para poder dar un estilo propio a cada capa para su visualización. En nuestro caso queríamos mostrar una escala de colores asociada a la temperatura, utilizando colores azules para las temperaturas frías, verdes y amarillas para las templadas y rojas para las cálidas. De esta manera, a través del SLD se creó un estilo con varias reglas, aquí se muestra parte del código:

```
<sld:Name>Escala_Colores</sld:Name>
  <sld:Title>escala25</sld:Title>
  <sld:FeatureTypeStyle>
    <sld:Name>name</sld:Name>
    <sld:Rule>
      <sld:Title>10</sld:Title>
      <ogc:Filter>
        <ogc:PropertyIsBetween>
          <ogc:PropertyName>rastervalu</ogc:PropertyName>
          <ogc:LowerBoundary>
            <ogc:Literal>10</ogc:Literal>
          </ogc:LowerBoundary>
          <ogc:UpperBoundary>
            <ogc:Literal>10.99999999</ogc:Literal>
          </ogc:UpperBoundary>
        </ogc:PropertyIsBetween>
      </ogc:Filter>
      <b><sld:PointSymbolizer>
        <sld:Graphic>
          <sld:Mark>
            <sld:Fill>
              <sld:CssParameter
name="fill">#6E1291</sld:CssParameter>
              <sld:CssParameter name="fill-
opacity">0.53</sld:CssParameter>
            </sld:Fill>
          </sld:Mark>
        </sld:Graphic>
      </sld:PointSymbolizer>
    </sld:Rule>
```

En esta parte del código se define el nombre del estilo “Escala_Colores”, dentro, se define una regla llamada “10” que indica mediante “PropertyIsBetween” el intervalo en el cual se va a aplicar, en este caso entre los valores “10” y “10.99999999” del atributo “rastervalu” (la temperatura), y que a este intervalo, le va a aplicar el estilo “PointSymbolizer”, puntos con el color “#6E1291” degradado. Como la temperatura va de 10°C a 35°C se crearon 27 (una más para los “noData”) reglas dentro de un único estilo (figura 6.15):

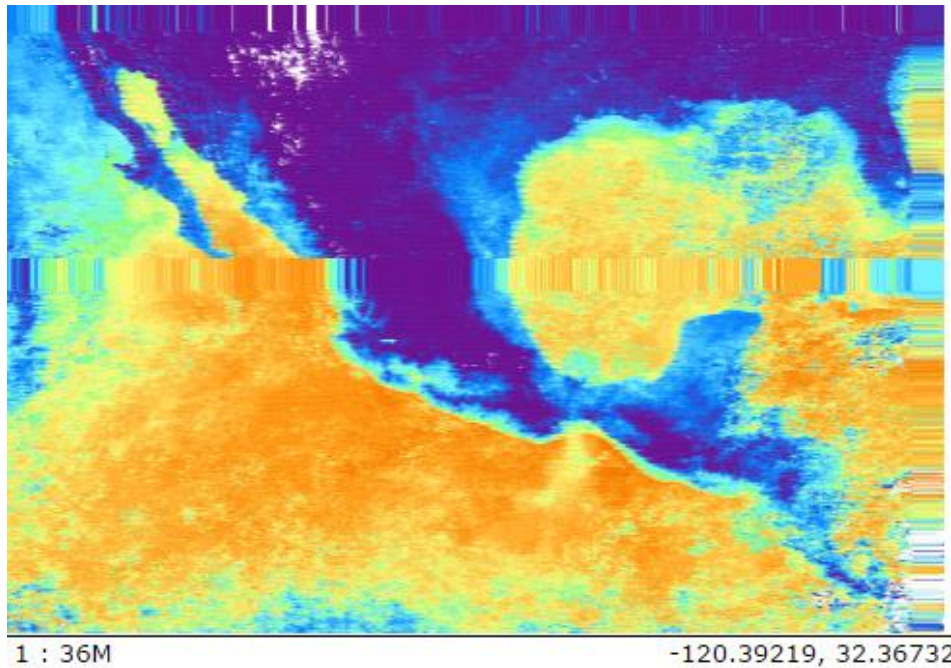


Figura 6.15 Visualización de la capa TSM con estilo SLD de escala de colores

Temperatura	Color	Temperatura	Color	Temperatura	Color
10-10.9	#6E1291	19-19.9	#73F3FF	28-28.9	#FF9111
11-11.9	#552BAA	20-20.9	#80FFE6	29-29.9	#F77800
12-12.9	#3C44C3	21-21.9	#80FF82	30-30.9	#DE5F00
13-13.9	#235DDC	22-22.9	#AFFF80	31-31.9	#C54600
14-14.9	#0A76F5	23-23.9	#E1FF80	32-32.9	#AC2D00
15-15.9	#0F8EFF	24-24.9	#FFF576	33-33.9	#931400
16.-16.9	#28A7FF	25-25.9	#FFDC5D	34-34.9	maroon
17-17.9	#41C1FF	26-26.9	#FFC344	35-35.9	maroon
18-18.9	#5AD9FF	27-27.9	#FFAA2A	noData	#FFFFFF

Tabla 6.1 Intervalos de temperatura con su respectivo color

EL SERVICIO DE CACHÉ

El siguiente paso es utilizar GeoWebCaché para mejorar el rendimiento de las capas que se presentarán en el mapa, que se hace por medio de la generación de tiles.

La generación de tiles (mosaicos), es cómo GeoWebCaché guarda las imágenes de mapa prerenderizadas. GeoWebCaché actúa como un proxy entre los clientes y un servidor WMS. Cuando el cliente (por ejemplo OpenLayers) hace una solicitud, GeoWebCache realiza las comprobaciones necesarias para ver si ya tiene la imagen correspondiente, si sí la tiene, se la envía al cliente de inmediato, en caso contrario, la solicitud se envía al servidor WMS

apropiado (por ejemplo GeoServer), cuando la respuesta vuelve, GeoWebCache guarda una copia (la cachea) y después la envía al cliente (Morales, 2014).

Una imagen de mapa no se almacena en caché “entera”. Es almacenada en forma de pequeñas imágenes rectangulares del mismo tamaño (tiles), que se combinan para formar mapas más grandes. Estos tiles se almacenan en más de una escala, de tal manera que los usuarios pueden ampliar y reducir los niveles de zoom sin necesidad de volver a muestrear (Morales, 2014).

- ✓ Para lograr un buen rendimiento con las capas de temperatura, se creó un caché de mosaicos; en la interfaz de GeoWebCache aparece una lista con todas las capas disponibles de GeoServer, y abajó de cada una un botón que dice “Seed this Layer” y se escriben los parámetros necesarios para hacer el caché (figura 6.16).

En “Type of operation” se eligió “Reseed” para genere todos los mosaicos, en “Grid Set” se puso el sistema de coordenadas “4326”, en “Zoom start” se puso 00 para renderice desde el zoom más alto (donde se visualiza todo el mapa) y “Zoom stop” es para decirle hasta que zoom va a renderizar, en nuestro caso sólo renderizará 4 niveles. Finalmente la información es enviada y sólo hay que esperar a que se generen todos los mosaicos.

Create a new task:

Number of tasks to use:

Type of operation:

Grid Set:

Format:

Zoom start:

Zoom stop:

Modifiable Parameters: STYLES:

Bounding box:

These are optional, approximate values are fine.

Figura 6.16 *Parámetros para la creación de caché de mosaicos*

EL MAPA INTERACTIVO

Con los procesos realizados anteriormente, el mapa con capas TSM ya puede ser publicado en Internet a través de una página web, sin embargo, para crear aplicaciones personalizadas WebMapping que puedan mostrar el mapa y que tengan controles para la interacción del usuario con el mapa, se necesita de OpenLayers una API desarrollada en JavaScript para lograr este propósito.

Se desarrolló la aplicación con JavaScript y la API OpenLayers, adicionalmente se utilizó GeoExt para crear una interfaz gráfica mucho más atractiva, se muestra a continuación las partes más importantes del código del script “appTSM”:

```
var map = new OpenLayers.Map(
  {
    projection: mercator,
    displayProjection: geographic,
    units: "dd",
    maxExtent: limites,
    controls: [
      new OpenLayers.Control.Navigation(),
      new OpenLayers.Control.ScaleLine(),
      new OpenLayers.Control.MousePosition(),
```

El objeto OpenLayers.Map es el encargado de crear el mapa, sobre él, se dibujaran todas las demás capas, los controles, las operaciones, se define la proyección cartográfica, las unidades y todo lo que se le quiera asociar.

```
//capas
var base = new OpenLayers.Layer.OSM();

var tsm = new OpenLayers.Layer.WMS(
  "Temperatura",
  "http://132.248.15.81:8080/geoserver/gwc/service", {
    'layers': ['Febrero12', 'Marzo12', 'Mayo12', 'Junio12', 'Julio12', 'Septiembre12',
      'Octubre12', 'Noviembre12', 'Diciembre12'],
    transparent: true, format: 'image/gif'},
  {isBaseLayer: false}
);

var temperatura = new OpenLayers.Layer.Vector(
  "TSM_2013", {
    projection: geographic,
    strategies: [new OpenLayers.Strategy.Fixed()],
    protocol: new OpenLayers.Protocol.WFS({
      url: "http://132.248.15.81:8080/geoserver/wfs",
      featureType: ["Enero12"],
      featureNS: "http://geoserver.org/bitsex"
    })
  });
map.addLayers([base, temperatura, tsm]);
```

En esta sección se crean las capas que se visualizarán sobre el mapa, la primera “base” es la capa base que muestra a la República Mexicana y sus mares, es proporcionada por el servicio Open Street Map (OSM), la siguiente es un conjunto de capas, de los diferentes meses del año 2012, uno de los parámetros más importantes es la ubicación desde donde

se sirven las capas: “<http://132.248.15.81:8080/geoserver/gwc/service>” aquí se indica la ip del servidor, el puerto **8080** (por el cual escucha geoserver) y **gwc** para indicarle que son las capas almacenadas en caché con el servicio GeoWeb Cache.

La última capa, es una capa vectorial WFS, que será la intermediaria entre el usuario y el servidor de mapas para hacer la consulta de información. Finalmente las capas son añadidas al mapa.

Para realizar la consulta de la información alfanumérica que se encuentra en la base de datos, se utilizaron eventos de JavaScript para obtener la posición del mouse al hacer clic, la información se pasó a un “popup” de OpenLayers.

```
temperatura.events.on({
  featuresselected: function(event) {
    var feature = event.feature;
    var htmlString = '<iframe frameborder="0" name="pop" width="500px" height="300px" scrolling="no"></iframe>'
    '<form action="http://localhost/Pruebas/js_php_2.php" method="post" target="pop"> |
    <input type="hidden" value="'+feature.attributes.id+'" name="lonlat"/> <input type="submit"/></form>' ;
    feature.popup = new OpenLayers.Popup.FramedCloud(
      "pop",
      feature.geometry.getBounds().getCenterLonLat(),
      null,
      htmlString,
      null,
      true
    );
    map.addPopup(feature.popup);
  },
});
```

Uno de los parámetros del “popup” es “contentHTML” que recibe una cadena con código HTML que ejecuta como tal, esta cadena definida antes del “popup” como una variable llamada “htmlString” y que es parte del motor para hacer las consultas a la base de datos; en este pequeño código html, se define una etiqueta “<iframe>” para que ahí se muestre el resultado de la consulta, después se define una etiqueta tipo formulario “<form>” y mediante el parámetro “action” le indicamos que ejecute el archivo **consulta.php**, la información que se le pasa al archivo php es el “**id**” del punto sobre el cual se hizo clic, este se obtiene mediante “**feature.attributes.id**” y se manda con la etiqueta “<input>”.

El archivo consluta.php es el que realiza la conexión a la base de datos espacial, realiza las consultas sql y despliega el resultado de las consultas en tablas html para poderlas mostrar en el popup cada vez que un usuario hace clic en un punto de temperatura.

Dentro del script “appTSM.js” también se definieron las herramientas del mapa: navegación, zoombox, historial, máxima extensión, imprimir y descargas, se creó el panel de herramientas.

```

//controles
var navegar = new OpenLayers.Control.Navigation({
  title:"Desplazar mapa",
  displayClass:'olControlNav'
});

var panel = new OpenLayers.Control.Panel({
  div: OpenLayers.Util.getElement('olControlEditingToolbar'),
  defaultControl: navegar
});

//botón zoom_area
var zb = new OpenLayers.Control.ZoomBox({
  title:"Acercamiento por área",
  displayClass:'olControlZoomBox'
});

```

Dentro del mismo script se utilizó geoExt para crear un árbol de capas y el panel que contiene al mapa. Una parte importante dentro de la definición de este panel, es el parámetro “render to: mappanel”, que le dice al mapa que se debe mostrar en un contenedor “<div>” con id = mappanel, en el archivo html que ejecute el script.

LA INTERFAZ GRÁFICA

El script anterior, para poder visualizarlo en un navegador es necesario cargarlo en un archivo html, en este archivo llamado “*appTSM.html*” también se cargan las librerías de OpenLayers y GeoExt, el script “*appTSM.js*” que contiene toda la aplicación WebMapping y que se describió anteriormente y los archivos CSS con los estilos que se crearon para las herramientas:

```

<script src="http://www.openlayers.org/api/2.11/OpenLayers.js"></script>
<script type="text/javascript" src="../script/GeoExt.js"></script>
<script type="text/javascript" src="appTSM.js"></script>
<link rel="stylesheet" href="estilo_mapa.css" type="text/css">

</head>
<body>
  <div id="contenedor">
    <div id="panel">
      <div id="olControlEditingToolbar"></div>
    </div>
    <div id="tree-id"></div>
    <div id="mappanel"></div>
  </div>
</body>
</html>

```

Se puede observar la etiqueta <div id="mappanel"> que es el contenedor que se encarga de recibir y mostrar el mapa web, existe otro contenedor <div id='olControlEditingToolbar'> que es el encargado de la barra de herramientas.

El resultado de todos los procesos anteriores es la consulta de la TSM a través de un mapa web (figura 6.17):

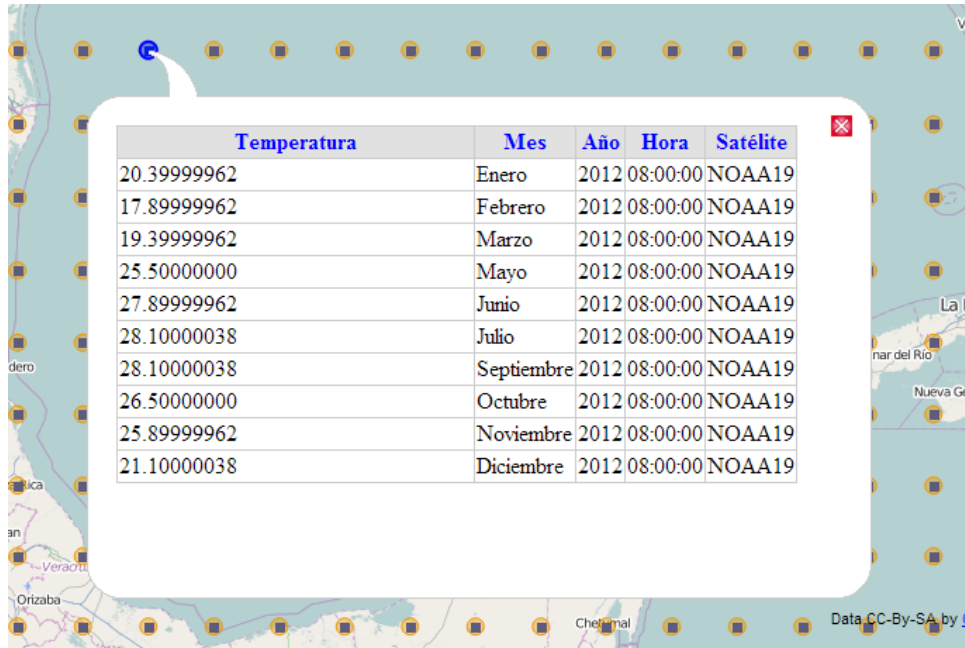


Figura 6.17 Resultado del caso de uso “Consultar TSM”

SUBSISTEMAS

Durante la etapa de análisis se pudieron identificar 4 subsistemas que juntos conforman el Sistema Web de Consulta de la TSM: appTSM, WebTSM, Galería TSM y appAdmon.

La appTSM se describió en el apartado anterior, para los subsistemas WebTSM, Galería TSM y appAdmon se utilizó HTML para la creación de las páginas web, java script para la interacción de la galería de imágenes y PHP para la administración de usuarios y la conexión a la base de datos.

6.1.7 Pruebas

En esta última etapa se realizan consultas a la base de datos para comprobar que está funcionando correctamente, que devuelve los valores esperados:

1.- ¿Cuáles son las coordenadas de los puntos cuya temperatura sea igual a 19.5 y provengan del satélite NOAA19?

```
select st_asText(t.geom) as "Coordenadas de 19.5°C y Satélite NOAA19"
from t112 t, satellite s, datos d
where t.rastervalu = 19.5
      and s.satellite = 'NOAA19'
      and s.id = d.satellite
      and d.nombre = t.nombre
```

	Coordenadas de 19.5°C y Satélite NOAA19 text
1	POINT(-78.0108695652173 23.625)
2	POINT(-111.728260869565 18.4464285714286)
3	POINT(-82.0978260869565 9.12499999999996)

2.- ¿Cuál es la temperatura, año a, el mes, la hora y el satélite del punto con coordenada "(-123.989130434783 13.2678571428571)"?

```
select t.rastervalu as "Temperatura", a.ano as "Año", m.mes as "Mes",
       h.hora as "Hora", s.satelite as "Satelite", p.tipo as "Paso"
from tsm112 t, ano a, mes m, hora h, satelite s, datos d, paso p
where st_asText(t.geom) = 'POINT(-123.989130434783 13.2678571428571)'
      and t.nombre = d.nombre
      and d.id_a = a.id_a
      and d.id_m = m.id_m
      and d.id_h = h.id_h
      and d.id_s = s.id_s
      and h.id_p = p.id_p
```

	Temperatura numeric	Año character varying(20)	Mes character varying(20)	Hora time without time zone	Satelite character varying(15)	Paso character varying(15)
1	19.00000000	2012	Enero	08:00:00	NOAA19	Diurno

Por otro lado se verifican los requerimientos no funcionales:

1. Tiempo para cargar la aplicación con el mapa: se pedía que el tiempo fuera menor a 5 segundos, el de la appTSM es de aproximadamente 5 segundos, sin embargo, considerando que se están cargando muchas capas y cada capa cuenta con mucha información aparte de sus estilos, sigue siendo un buen tiempo de carga y no sobrepasa los 5 segundos.
2. Tiempo de respuesta del mapa al interactuar con el: se pedía que la interacción no fuera mayor a 1 segundo, sobre todo al momento de utilizar el zoom, y efectivamente el tiempo de respuesta real del mapa es menor a 1 segundo, esto fue logrado gracias a la generación de mosaicos que se realizó con GeoWeb Cache.
3. En cuanto a la galería de imágenes, se utilizaron <iframes> así cada vez que se pedía una imagen sólo se actualizaba el <iframe> de la imagen y no la página entera.

6.2 RESULTADOS

Después del tratamiento de datos y del desarrollo de la metodología, se obtuvo un sistema funcional que cumple con los casos de uso planteados en el desarrollo. A continuación se presentan los subsistemas que se crearon y que en conjunto forma el Sistema Web de Consulta de la TSM.

Subsistema appWebTSM

Este subsistema se encarga de dar la bienvenida a los usuarios, mostrar la información del proyecto, de ser el enlace con las otras aplicaciones y páginas (figura 6.18), y proporcionar los enlaces para que un usuario se registre o inicie sesión (figura 6.19).



Figura 6.18 *Página de bienvenida del sistema*

Cuando se cliquea sobre el botón “Ingresar”, se redirecciona a una nueva página que contiene un formulario para poder ingresar los datos de tu usuario y contraseña (figura 6.16), si éstos son correctos, al cliquear sobre el botón “Ingresar”, se redirecciona a la nueva página de bienvenida, la cual, en la parte superior derecha, aparece el nombre del usuario que inició sesión y la posibilidad de cerrar sesión (figura 6.20); si los datos son incorrectos o de que el usuario no exista se envían mensajes de alerta.

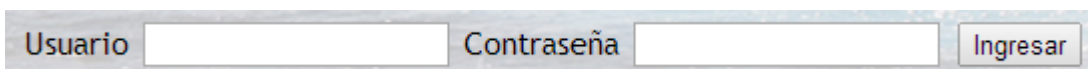


Figura 6.19 *Formulario para iniciar sesión*

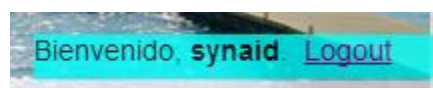


Figura 6.20 *Indicador de que has iniciado sesión y la opción para cerrar sesión*

En el caso en que un usuario quiera registrarse en el sistema, al dar clic en el botón “Regístrate”, se redireccionará a una nueva página con un formulario para llenar con sus datos, una vez terminado y cliqueando en el botón enviar, la información será almacenada en la base de datos y se redireccionará a la página de bienvenida normal.

Subsistema appTSM

La aplicación appTSM es sin duda la parte más importante y principal de todo el sistema (figura 6.21), que cuenta de tres partes principales, “Árbol de capas”, “Menú de Herramientas” y “Visualizador de mapa”.

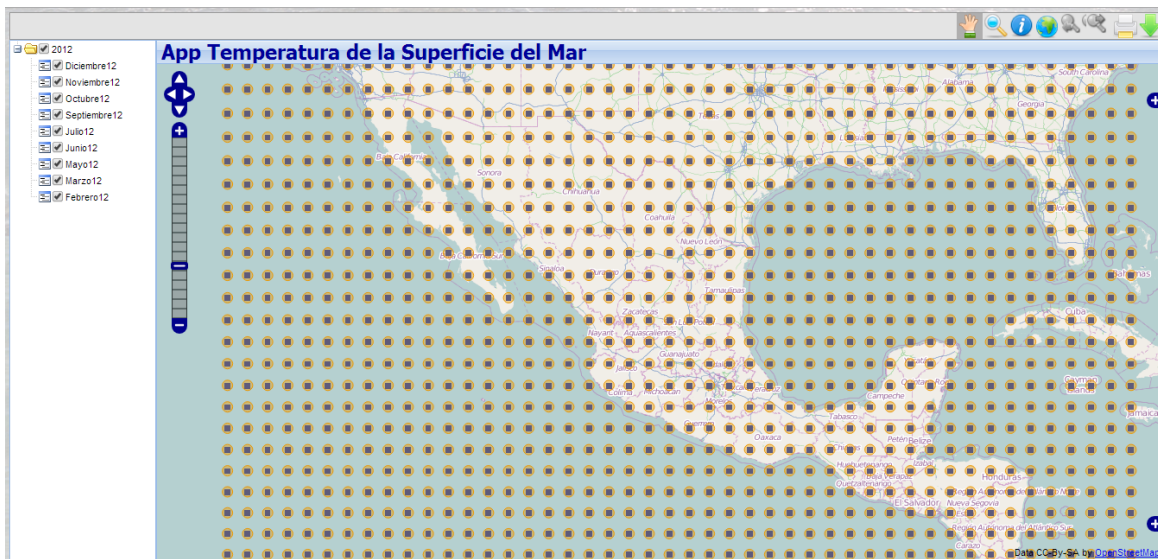


Figura 6.21 Aplicación WebMapping AppTSM

El árbol de capas (figura 6.22) permite seleccionar que capa se va a visualizar, mostrando así las diferentes distribuciones de temperatura a través de los colores (figura 6.23)

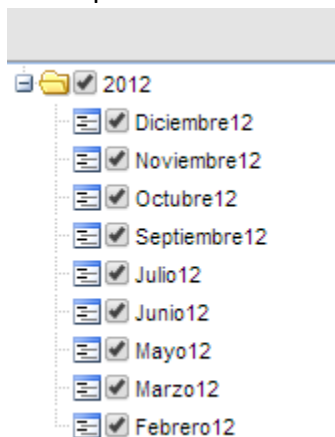


Figura 6.22 El árbol de capas está dividido por carpetas de meses, lo que hace más fácil el encontrar alguna capa en específico

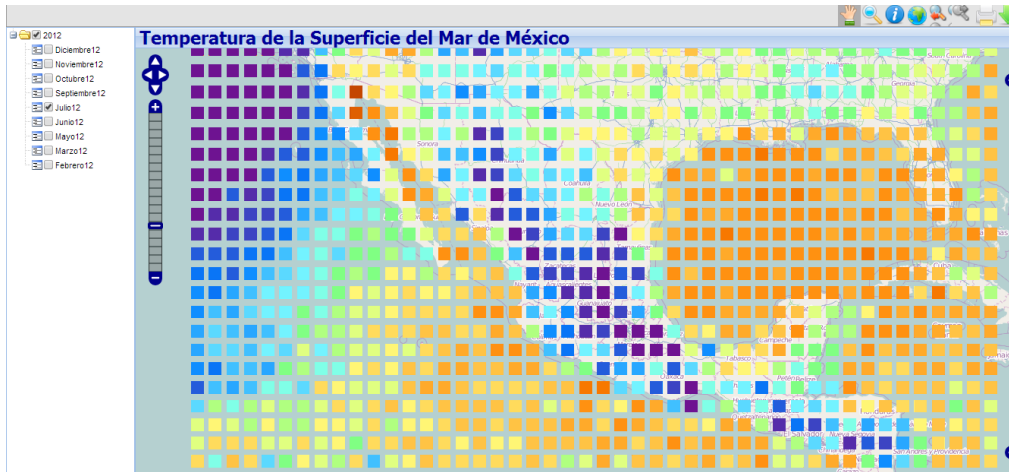


Figura 6.23 Capa de Julio de 2012 que muestra a través de una escala de colores como fue la distribución de temperatura

La barra de herramientas (figura 6.24) se compone de los siguientes elementos:

- Herramienta Selección: permite al usuario mover el mapa hacia cualquier lado.
- Herramienta Zoom por área: permite hacer acercamientos cuando se dibuja un área.
- Herramienta Información (por default): es la herramienta más importante, ya que es la que realiza las consultas a la base de datos, haciendo uso del protocolo WFS y el servidor de mapas. Al hacer clic sobre uno de los puntos del mapa, devuelve la información de la temperatura, el mes, el año, la hora y el satélite de todas las capas.
- Herramienta Máxima Extensión: al presionar sobre el botón, se regresa siempre a la máxima extensión del mapa, que en este caso son los límites definidos en el mapa, 124°W, 6°N y 78°W, 34°N.
- Herramientas Historial Zoom: permite regresar al zoom anterior o al que se acaba de dejar.
- Herramienta Impresión: imprime de la página web sólo el mapa, y como tiene su escala y mantiene sus proporciones, al imprimirlo es totalmente válido y útil, aun cuando se hagan acercamientos o modificaciones.
- Herramienta Descargas: redirecciona a una nueva página que contiene los archivos de la TSM, los originales ASCII y los shapefiles utilizados para la aplicación. Esta página está sólo disponible para los usuarios registrados, por lo que si se intenta ingresar a ella sin haber iniciado sesión, se le mandará un mensaje de que no tiene autorización para realizar descargas.



Figura 6.24 Barra de herramientas que permite una completa interacción con el mapa

Por último el Visualizador del mapa, es en sí el mapa y todas las capas que se cargan a él, todas las acciones que se realicen sobre el mapa, se reflejarán aquí. También contiene algunas otras herramientas, de lado izquierdo se encuentran unas flechas de navegación para poder mover el mapa, debajo de estas, se encuentra el zoom, que se puede controlar por medio de la barra, en la parte inferior se encuentra **la escala** del mapa, la cual cambia cada vez que el zoom se modifique. De lado derecho y arriba se localiza una pestaña que al presionarla se despliega otro control de capas, y más abajo existe otra pestaña que muestra un pequeño visualizador mostrando en que parte del mapa te sitúas y más hasta abajo se muestran **las coordenadas** en donde se localiza el puntero.

Subsistema Galería TSM

La galería TSM permite visualizar las imágenes satelitales (figura 6.25); existen dos principales categorías, “Años” y “Satélites”. En la primera se puede elegir el año que se desea visualizar, al hacerlo, se despliega en el centro de la página dos submenús, en el primero se puede elegir el mes, ya que existen diferentes imágenes para cada mes (de diferentes satélites y horas), el segundo submenú sirve para la elección de este tipo. Debajo de estos menús se despliega la imagen seleccionada y se puede interactuar fácilmente a través de estas imágenes.

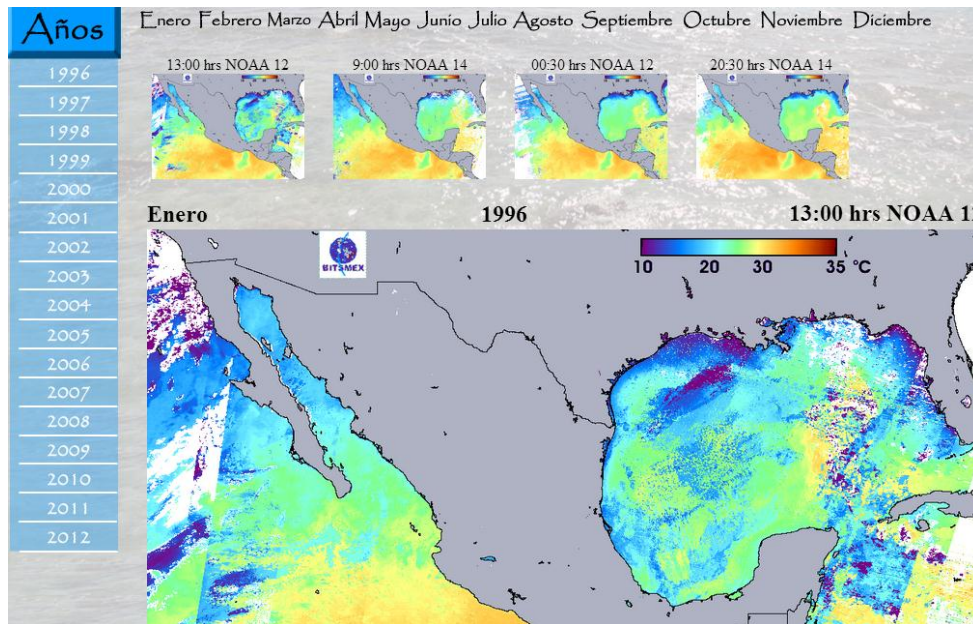


Figura 6.25 Subsistema “GaleríaTSM” que muestra la colección de imágenes de la TSM

CAPÍTULO 7 CONCLUSIONES

Los estudios que se realizaron de aplicaciones web para mapas y de software orientado a objetos, permitieron determinar una opción que cumpliera con el objetivo principal de la tesis. Aplicar una metodología para desarrollar un sistema web para la consulta de información de una base de datos espacial de variables oceanográficas, teniendo como resultado la consulta de mapas de la temperatura de la superficie de los mares de México.

La forma más eficiente para representar, consultar y distribuir la gran cantidad de información de la Temperatura de la Superficie del Mar (TSM) es a través de una aplicación WebMapping, construida sobre una arquitectura geoespacial OpenGeoSuite (base de datos espacial, servidor de mapas, servidor de teselas de mapas y un cliente de mapas web).

Un aspecto que llama la atención de las aplicaciones webmapping, es que la mayoría no utilizan todo el potencial que ofrecen las bases de datos, ya que sólo usan éstas para tener almacenados sus datos de forma segura y ordenada, es decir, al momento de hacer consultas sólo trabajan con las tablas (capas) con características geoespaciales, y las tablas alfanuméricas quedan excluidas, es decir, no es posible consultar información contenida en de tablas alfanuméricas que están relacionadas con las tablas espaciales a través de un mapa web; esto implica dejar a un lado todas las ventajas que las bases de datos relacionales nos proporcionan.

Tanto la base de datos como la aplicación webmapping del Sistema Web de Consulta de la TSM están diseñadas para hacer frente al inconveniente anterior, por un lado, para la base de datos se utilizaron todos los conceptos de modelado relacional de datos (relaciones, llaves primarias y foráneas, etc.) para poder realizar consultas complejas, demostrando que con una base de datos bien diseñada, se pueden obtener muy buenos resultados, y por otro lado dentro del programa de la aplicación, se desarrolló una función que permite a través de la interacción del usuario con el mapa, realizar consultas que utilizan tanto las tablas espaciales como las alfanuméricas.

Esta **función de consulta** también permite realizar consultas espaciales, como distancias, áreas, etc., y operaciones entre capas, por ejemplo, en un futuro se podrán crear nuevas capas que representen regiones del mar (a través de polígonos) y preguntar, ¿Cuáles son las temperaturas contenidas en el Golfo de Tehuantepec?, con lo cual se podrán agregar herramientas de análisis especializado a la aplicación.

Utilizar una metodología de desarrollo de software orientada a objetos, fue parte fundamental para poder crear el sistema que todo funcione correctamente, sea amigable con el usuario, contar con una documentación amplia para posibles cambios y mejoras, y que sea escalable, lo cual permitirá en el futuro seguir añadiendo más información de la TSM, otras variables oceanográficas.

Sin duda alguna se dio un gran paso en cuanto a la TSM, en primer lugar fue posible pasar de un archivo plano de datos a un archivo con características geoespaciales, esto permitió que estos archivos pudieran ser utilizados en la aplicación WebMapping, esta aplicación siempre estará actualizada, los usuarios no tendrán que instalar ningún tipo de software en sus computadoras para poder utilizarlo y podrán acceder a la aplicación en cualquier momento y desde cualquier computadora que tenga conexión a internet, y se establecieron las bases para que en un futuro la aplicación esté dotada de análisis espacial que ayudará a expertos en oceanografía física a afrontar y dar solución a problemas relacionados con la temperatura del mar de México.

Las aplicaciones WebMapping son el presente y futuro para trabajar y compartir información georreferenciada en un mundo en el cual la información debe estar disponible y actualizada todos los días.

BIBLIOGRAFÍA

Caire Lomelí, Jorge (2002). *Cartografía Básica*. México, Facultad de Filosofía y Letras, UNAM.

Celma Giménez, M.; Casamayor Ródenas J. C.; Mota H., L. (2003). *Bases de Datos Relacionales*. Pearson Educación.

Coll A., E.; Sanz S., J. G.; Martínez Llario, J. C.; Irigoyen G., J. (2005). *Introducción a la publicación de cartografía en Internet*. Valencia: Editorial de la Universidad Politécnica de Valencia.

Correia, Paul (2002). *Guía práctica del GPS*. Marcombo.

Date, C. J. (2001). *Introducción a los Sistemas de Bases de Datos*. 7ª ed. México: Pearson Educación.

Elmasri Ramez, A.; Navathe Shamkant, B. (2007). *Sistemas de Bases de Datos, Conceptos Fundamentales*. 2ª ed. México, D.F.: Pearson Prentice Hall.

Fernández Alarcón, Vicenç (2006). *Desarrollo de sistemas de información. Una metodología basada en el modelado*. Barcelona: Ediciones de la Universidad Politécnica de Catalunya.

Jacobson, Ivar; Booch, Grady; Rumbaugh, James (2000). *El proceso Unificado de Desarrollo de Software*. España: Addison Wesley.

Kurose, James F.; Ross, Keith W. (2004). *Redes de computadoras un enfoque descendente basado en Internet*. 2ª ed. España: Pearson educación.

Longley, Paul A.; Goodchild, Michael F.; Maguire, David J.; Rhind, David W. (2011). *Geographic Information Systems & Science*. 3ª ed. USA: Aptara.

Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. España: Editorial Club Universitario.

Mateu, Careles (2004). *Desarrollo de aplicaciones web*. Barcelona: Formación de Posgrado.

Morales, Aurelio. *Teoría del curso en línea Desarrollo de aplicaciones Web Mapping*. De enero de 2014, España.

Neumann, Andreas; Shekar, Shashi; Xiong, Hui (2008). *Encyclopedia of GIS*. Springer.

Oracle Education. *Manual del curso Desarrollo de modelos de datos y diseño de base de datos relacionales*. De marzo de 2001.

Piattini et al. (2007). *Análisis y diseño de aplicaciones de gestión. Una perspectiva de Ingeniería de Software*. España: Ra-Ma.

Pons Capote, O., et al, (2008). *Introducción a los Sistemas de Bases de Datos*. España: Paraninfo.

Pressman, Roger S. (2002). *Ingeniería del Software. Un enfoque práctico*. 5ª ed. España: McGraw-Hill.

Raisz, Erwin (1985). *Cartografía General*. 7ª ed. Barcelona: Omega.

Ramos M., Alicia; Ramos M., M. Jesús (2014). *Aplicaciones Web*. 2ª ed. Madrid: Ediciones Paraninfo.

Rigaux, Philippe; Scholl, Michel; Voisard, A. (2002). *Spatial Databases with Application to GIS*. USA: Morgan Kaufmann Publishers.

Schildt, Herbert (2001). *Manual de referencia Java 2*. 4ª ed. Madrid: McGraw-Hill.

Schmuller, Joseph (2001). *Aprendiendo UML en 24 horas*. Prentice-Hall

Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. 3ª ed. Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

Silva Romo, G.; Mendoza R., C. C.; Campos M., E. (2010). *Elementos de Cartografía Geológica*. México, Universidad Nacional Autónoma de México, Facultad de Ingeniería.

Sommerville, Ian. (2005). *Ingeniería del software*. 7ª ed. Madrid: Pearson Educación.

Weitzenfeld, Alfredo (2005). *Ingeniería de software orientada a objetos con UML, Java e Internet*. México: Thomson.

Zurdo, David; Acevedo, Fernando; Sicilia, Alejandro (1998). *Internet*. España: Paraninfo.

Páginas web

Alarcón, José (2007). *Administración PostgreSQL*. Consultado en Julio 2014. Artículo disponible en: <http://www.uv.es/alarmedi/iti2007/curso-iti-2007.pdf>

Batzli, Sam (2010). *America View Workshop: Introduction to Geospatial Web Services*. Consultado en enero 2014. Artículo disponible en: http://wisconsinview.ssec.wisc.edu/education/av_workshop_2010.html

Del Pilar Angeles, M. *Notas Bases de Datos*. Consultado en octubre 2013. Artículo disponible en: <http://profesores.fi-b.unam.mx/pilarang/docencia/bd/basededatos.html>

- Dunning, David (2013). *What are Spatial Databases?*. Consultado en diciembre 2013. Artículo disponible en: http://www.ehow.com/info_10069998_spatial-databases.html
- Escalona Cuaresma, María José (2001). *Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta*. Consultado en septiembre 2013. Disponible en: <https://www.lsi.us.es/docs/informes/EstadoActual.pdf>
- Esri. *ArcGIS Resources*. Consultado en enero 2014. Artículo disponible en: <http://resources.arcgis.com/es/help/main/10.1/index.html#/na/00qn0000001p000000/>
- Jiménez Hernández, Eréndira M.; Orantes Jiménez, S. D. *Metodologías híbridas para desarrollo de software: una opción factible para México*. Revista digital universitaria. Enero de 2012 Vol. 13, No. 1. Consultado en noviembre 2013. Disponible en: <http://www.revista.unam.mx/vol.13/num1/art16/>
- Macías, C.; Orozco, Sergio (2014). *Uso de UML en aplicaciones Web: páginas y relaciones*. Consultado en agosto 2014. Artículo disponible en: [http://www.milestone.com.mx/articulos/uso de uml en aplicaciones web.htm](http://www.milestone.com.mx/articulos/uso_de_uml_en_aplicaciones_web.htm)
- Morales, Aurelio (2012). *¿Qué es OpenGeo Suite?* Consultado en enero 2014. Artículo disponible en: <http://mappinggis.com/2012/05/que-es-opengeo-suite/>
- Olaya, Victor (2011). *Sistemas de Información Geográfica*. Consultado en octubre 2013. Artículo disponible en: <http://www.bubok.es/libros/191920/Sistemas-de-Informacion-Geografica>
- Olgún Espinoza, J. M (2004). *Análisis Orientado a Objetos*. Consultado en septiembre 2013. Artículo disponible en: <http://yaqui.mx.l.uabc.mx/~molguin/as/>
- Ramsey, Paul. *PostGIS Manual*. Consultado en mayo 2014. Artículo disponible en: <http://www.dcc.fc.up.pt/~michel/TABD/postgis.pdf>
- Reyes, Mario; Martínez Arce, Juventino (2003). *Tecnologías de información, cartografía y geografía en la era digital*. Boletín de Política Informática No. 2, disponible en: <http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/tecnologias.pdf>

ANEXOS

ANEXO A

Documento de Especificación de Requerimientos del Sistema (Software Requirements Specification, SRS)

1. Introducción

1.1 Propósito

Este documento tiene como propósito realizar un registro, una descripción detallada, una organización y una relación de los requerimientos tanto funcionales como no funcionales que estén presentes en el sistema que se va a desarrollar, el Sistema Web de Visualización de TSM, que permitirá hacer una primera aproximación de los casos de uso del sistema, por el mismo motivo, también se hará un registro de los usuarios involucrados, las restricciones y los riesgos que puedan estar contenidos en el sistema. Este documento será la base para poder construir el sistema.

Se incluye una sección de glosario para aclarar los términos el negocio que serán usados frecuentemente.

1.2 Alcance

El Sistema Web de Consulta de la TSM (SWCT) tendrá una aplicación (appTSM) que se encargará de hacer consultas a través de un mapa interactivo, para mostrar la información de los promedios mensuales de la Temperatura de la Superficie de los Mares de México de diferentes años y satélites. Este mapa interactivo deberá tener las funciones básicas de un Sistema de Información Geográfica, es decir, deberá tener controles para realizar acercamientos y alejamientos (zoom), una herramienta de desplazamiento por el mapa, intercambio de capas, mostrar en todo momento la latitud y longitud del cursor, una herramienta para hacer descargas y otra para imprimir. La información estará almacenada en una base de datos espacial y se conectará con la aplicación web.

El sistema deberá contar con un mecanismo de registro para usuarios académicos, esto permitirá que estos usuarios puedan tener acceso a más contenido y funciones. Los demás usuarios que no sean académicos podrán tener acceso a la mayor parte del sistema. El administrador del sistema deberá tener la posibilidad de dar de alta y baja a los usuarios académicos.

El sitio web que será parte del sistema deberá tener una sección que muestre también las imágenes de la TSM de los diferentes años y satélites, y la información correspondiente al satélite.

1.3 Involucrados

Actor (rol)	Involucrado Primario	Involucrados Secundarios
Administrador	Encargado del Laboratorio de Oceanografía Física	Usuario normal, usuario académico
Usuario		Administrador
Usuario académico	Invitado académico	Administrador

1.4 Acrónimos y abreviaturas

Acrónimo/Abreviatura	Descripción.
SWCT	Sistema Web de Consulta de la TSM
appTSM	Aplicación web de la TSM
TSM	Temperatura de la Superficie del Mar
SIG	Sistema de Información Geográfica
RF	Requerimientos Funcionales
RNF	Requerimientos No Funcionales
SRS	Documento de Especificación de Requerimientos
ASCII	American Estándar Code for Information Interchange

2. Restricciones y suposiciones

- ✓ Se deberá utilizar un manejador de base de datos con extensión espacial para la correcta administración de los datos geográficos.
- ✓ El acceso al sistema deberá ser vía web y desde cualquier computadora o dispositivo móvil que tenga una conexión a Internet y un navegador web.
- ✓ Los formatos para la descarga de la información deberán ser ascii y shapefiles.
- ✓ El sitio web se deberá poder visualizar en los principales navegadores.

- ✓ Es deseable que la aplicación muestre la información con un tamaño de celda de 1 minuto por 1 minuto, sin embargo si el tamaño de la celda es de 1 grado por 1 grado es suficiente.
- ✓ Dado el presupuesto con el que se cuenta para desarrollar el sistema, se pide que se utilicen herramientas de software libre para el desarrollo de los programas y para la gestión de los datos (Sistema Manejador de Base de Datos).
- ✓ Las imágenes satelitales deberán ser procesadas (el formato actual es ASCII) para poder tener un formato compatible con la base de datos.
- ✓ Dado que los archivos originales (ASCII) son sólo una matriz de datos y no tiene georreferenciación, es muy importante que se realice un tratamiento a los datos para que se asocien con sus respectivas coordenadas y estén georreferenciados correctamente.
- ✓ Cada coordenada deberá ser añadida como un nuevo dato, para poder ingresarlo en la base de datos como la componente espacial.
- ✓ Se deberá trabajar con el sistema de referencia 'Mercator'.
- ✓ Cualquier persona podrá tener acceso al sistema, sin embargo, no todas las personas tendrá el mismo grado de accesibilidad a todas las partes del sistema, por lo que se deberán registrar con el mecanismo de registro.
- ✓ El mecanismo de registro, deberá tener campos para guardar la información del usuario de su nombre, apellido, correo electrónico, institución de procedencia (opcional), nombre de usuario y contraseña.
- ✓ Todos los datos de los usuarios que se registren deberán ser almacenados en la base de datos.
- ✓ Los usuarios normales (que no estén registrados), tendrán acceso a la mayor parte del sitio web, es decir a la página de bienvenida, información, galería de imágenes, contacto y a la appTSM, sin embargo no podrán hacer uso de la función de "descargar".
- ✓ Los usuarios académicos tendrán el mismo acceso que los usuarios normales y podrán hacer uso de la función "descargar".
- ✓ El mecanismo de gestión de usuarios deberá ser también a través de Internet, y esta página sólo será visible por el administrador, que será el único que podrá acceder a ella.
- ✓ El sitio web deberá estar disponible durante todo el día.

3. Requerimientos funcionales

3.1 Características principales del Sistema

- ✓ La información deberá ser almacenada en una base de datos espacial y deberá tener conexión con la aplicación web.
- ✓ La aplicación se conectará con la base de datos espacial para la consulta de información.
- ✓ La aplicación web deberá mostrar la información al hacer clic en cualquiera de las celdas que estén comprendidas dentro de los límites que abarque el mapa base y sólo en las áreas que comprenden el mar.
- ✓ La consulta de información también se podrá realizar al seleccionar un área.
- ✓ La información que se mostrará por cada celda será la siguiente: temperatura promedio mensual, hora promedio, año, paso y satélite del cual se obtuvo dicha información.
- ✓ La aplicación web tendrá un mapa interactivo por medio del cual se realizarán todas las consultas.
- ✓ El mapa web interactivo tendrá herramientas para hacer acercamientos, alejamientos, intercambio de capas, desplazamiento, impresión del mapa, mostrar la posición del cursor en coordenadas y mostrar la escala cuando se hagan acercamientos o alejamientos.
- ✓ Se deben poder descargar las imágenes satelitales en los formatos ascii y shapefile.
- ✓ El sistema deberá tener un mecanismo de registro de usuarios.
- ✓ El administrador del sistema deberá tener una función que le permita ver todos los datos de los usuarios registrados y poder dar de alta, baja o modificar a los usuarios.
- ✓ El sistema deberá permitir la visualización de las imágenes de la SST.
- ✓ El sitio web deberá contener una página de bienvenida, otra de información sobre el proyecto del instituto, otra donde se puedan visualizar las imágenes de los diferentes años y satélites y la página que mostrará la aplicación web.

3.2 Descripción de los actores

Actor No. 1: Administrador, es el encargado de dar mantenimiento al sistema, actualizar la información, dar de alta y baja a los usuarios, así como modificar sus permisos, y asegurarse que la página esté disponible todo el tiempo.

Actor No. 2: Usuario normal, cualquier persona que desee ingresar al sitio web, podrán tener acceso al Sistema desde un navegador y conexión a Internet, y tendrán la posibilidad de consultar la información e interactuar con el mapa web.

Actor No. 3: Usuario académico, académicos y especialistas, tendrán la misma accesibilidad que un usuario normal, pero tendrán la ventaja de poder descargar los archivos.

3.3 Casos de Uso.

Caso de Uso	Prioridad	Num	Descripción
Consultar TSM	Esencial (E)	1	Permite a los usuarios la consulta de la información de la TSM a través del mapa interactivo.
Interactuar con el mapa web	Esencial	2	Permite a los usuarios la interacción con el mapa (intercambio de capas, zoom, desplazamiento, mostrar posición del puntero), por medio de herramientas.
Registrar usuarios	Esencial	3	Permite a los usuarios registrarse en el sistema.
Descargar información TSM	Esencial	4	Permite descargar la información TSM consultada a los formatos shapefiles y ASCII.
Administrar usuarios	Esencial	5	Permite al administrador gestionar a los usuarios registrados.
Visualizar imágenes TSM	Alta (A)	1	Permite visualizar las imágenes TSM de los diferentes años y satélites.

3.4 Aplicaciones

Subsistema	Descripción / Casos de Uso
appTSM	Esta aplicación permite la conexión a la base de datos espacial, la consulta y despliegue de la información de la TSM y la interactividad con el mapa web. Soporta los FRs: E1, E2 y E4.
appWebTSM	El sitio web permite al usuario ver toda la información correspondiente al proyecto, el acceso a la appTSM, el registro de los usuarios y el acceso a los usuarios registrados. Soporta los FRs: E3, A2.
GaleríaTSM	Esta pequeña aplicación permite la visualización de las imágenes de la TSM de forma interactiva. Soporta los FRs: A1.
appAdministración	Permite al administrador del sistema dar de alta, baja y modificar a los usuarios registrados. Soporta los FRs: E5.

3.5 Requerimientos funcionales para cada caso de uso.

FR	Descripción
E1-1	Permitirá consultar la información relacionada con la TSM, como es temperatura promedio mensual, hora promedio, satélite, año y paso.
E1-2	Permitirá hacer consultas no sólo de un punto, también se podrán hacer consultas de áreas que abarquen varios puntos.
E2-1	Permitirá a los usuarios hacer acercamientos, alejamientos, intercambio de capas, desplazamiento, impresión del mapa, mostrar la posición del cursor en coordenadas y mostrar la escala.
E3-1	Permitirá a los usuarios poder registrarse en el sistema.
E4-1	La descarga de un archivo deberá permitir al usuario especificar el formato en que desea obtener el archivo.

E5-1	Permitirá al administrador gestionar a los usuarios registrados.
A1-1	La aplicación de la galería deberá permitir escoger el año y el mes que desea visualizar.
A1-2	La aplicación de la galería deberá permitir la visualización escogiendo el satélite, que mostrará todas las imágenes asociadas a ese satélite.

3. Requerimientos no funcionales

NFR	Descripción
E1-101	Dada la experiencia que se tiene con otras aplicaciones de mapas web, se sabe que el tiempo de carga del mapa web es considerablemente largo (aproximadamente 5 segundos), se espera que el tiempo para mostrar el mapa sea menor a 5 segundos.
E1-102	Esto también nos ha mostrado que a la hora de hacer acercamientos y alejamientos del mapa, el rendimiento de las aplicaciones disminuye la experiencia de continuidad del usuario, ya que el tiempo de espera a la hora de utilizar el zoom es bastante alto (aproximadamente 3 segundos), se espera que este tiempo baje a menos de 1 segundo, para que el usuario pueda tener una experiencia de alto rendimiento.
E4-101	Los archivos para descargar deberán ser en formatos ascii y shapefiles.
A1-101	La galería de imágenes permitirá al usuario escoger el mes y año que desea visualizar, como se tienen muchas opciones para elegir, no sería óptimo que cada vez que el usuario elija una opción, la página entera se recargue para mostrar la imagen deseada, por lo que se espera que esta pequeña aplicación pueda ser capaz de sólo actualizar la parte que corresponde a la imagen para que el usuario disfrute de una experiencia óptima y que permita realizar un mejor análisis.

5. Glosario del proyecto

Término	Definición
Shapefile	Es un formato que se utiliza para almacenar la ubicación geométrica y la información de atributos de las entidades geográficas. Las entidades geográficas se pueden representar por medio de puntos, líneas o polígonos.
Ascii	
SIG	Sistema de Información Geográfica (SIG) es un sistema computarizado compuesto, por hardware, software, datos y aplicaciones que es usado para registrar digitalmente, editar, modelizar y analizar datos espaciales y presentarlos en forma alfanumérica y gráfica.
Mapas Web	Son mapas digitales que son presentados y distribuidos a través de la web. Son útiles ya que se pueden compartir y utilizar a través de una conexión web sin que el usuario deba instalar un software, están disponibles a cualquier hora y en cualquier lugar, su modificación y distribución es fácil y barata.
Mapas interactivos	Son mapas digitales que permiten al usuario interactuar con el mapa por medio de controles y herramientas, lo que permite su modificación y hasta manipulación de datos.
Bases de datos espaciales	Son bases de datos que aparte de manejar datos alfanuméricos es capaz de almacenar y manipular datos geográficos.

FIN DEL DOCUMENTO SRS

ANEXO B

Tarjetas CRCs (Class Responsibility Collaboration)

Consulta	
Consultar información TSM	Usuario Información TSM Mapa interactivo

Satélites	
----	----

controles	
----	----

Formatos	
Elegir	----

Base de Datos Espacial	
Conectar con el sistema	----

Imágenes TSM	
Ser visualizadas	Usuario

Zoom	
Hacer acercamientos y alejamientos	Mapa interactivo Usuario

Capas	
Mostrar diferentes capas	Mapa interactivo Usuario

Coordenada	
Mostrar la coordenada de la posición actual del puntero	Mapa interactivo Usuario

Descarga	
Hacer descargas de archivos	Mapa interactivo Usuario