



Universidad de las Ciencias Informáticas Facultad 6

Curso 2010-2011

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERO EN CIENCIAS INFORMÁTICAS**

**Título: Sistema de Información Geográfica para la UJC
Nacional.**

Autor: Julio Collado Pérez.

Tutor: Karel de la Cruz Rigueiro.

Ciudad de La Habana, Junio del 2011.

“Año 53 de la Revolución”

Dedicatoria

En especial a mis abuelos por ser mi mayor inspiración.

A mi madre y mi hermana por ser mis sentidos y lo que más quiero.

A mi papá por ayudarme, a pesar de la distancia y a Arnaldito por enseñarme gran parte de lo que sé hoy.

A Jesús por apoyarnos en todo momento y aceptar a mi familia tal y como es.

A mi tío Nelson por sus consejos y representar tanto para mí.

A mis tíos Niurka, Elena, Dora, Dámaso por todo lo que han hecho por mí.

A mis queridos primos Arnaldito y Odalis por ser los mejores.

A mi amigo Rafael por ser este su sueño también.

Y por último sin dejar de ser menos importante a mi novia y compañera Leydanis por estar siempre donde más la necesité, a mi lado.

Agradecimientos

A mi madre porque se lo debo todo y a mi hermanita por hacer que me esfuerce tanto para dar el ejemplo.

A toda mi familia por el gran apoyo que me proporcionó en todos estos años de estudios.

A la familia de mi novia por todo el afecto que me han dado todos estos años.

A los profesores Karel, Adrián, Alain, Alejandro, Aliana, Pedro y David por su ayuda en el desarrollo de mi tesis.

A mis mejores amigos Yanet, Zulema, Noel, Rafael y José Alberto por ser parte de mi vida.

A todas las amistades que hice en estos 5 años y a mis compañeros de aula.

A Leydani y al destino por ponerla en mi camino.

A todas esas personas que por un motivo u otro olvidé mencionar, pero pueden estar seguros que se los agradezco.

A todos.

Arigatou.

Julio

Declaración de Autoría

Declaramos que somos los únicos autores del trabajo titulado:

“Sistema de Información Geográfica para la UJC Nacional”

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Julio Collado Pérez.

Karel de la Cruz Rigueiro.

Datos del Contacto

Síntesis del Tutor: Ing. Karel de la Cruz Rigueiro.

Profesión: Ingeniero en Ciencias Informáticas.

Categoría docente: Adiestrado.

Año de graduado: 2009.

Resumen

Producto del vertiginoso auge de las Tecnologías de la Información y las Comunicaciones (TIC), las cuales han impuesto un nuevo orden mundial al proceso de gestión de la información, la mayoría de las empresas optan por alternativas automatizadas con el objetivo de seguir desarrollando sus principales procesos a la altura que supone la dinámica de la toma de decisiones de la actualidad. Actualmente la Unión de Jóvenes Comunistas (UJC) se encuentra en un proceso de informatización en aras de mejorar el control sobre toda la información que maneja.

Durante este proceso se identificaron un conjunto de dificultades que están relacionadas con la manera en que se manipulaba la información referente a los locales que maneja la UJC destinados a la recreación en la etapa veraniega, trayendo como consecuencia que fuera difícil hacer un análisis estadístico sobre dicha información. Además, la UJC carece de un sistema que represente geográficamente esta información sobre un mapa.

La investigación que se presenta es el resultado del ciclo completo de desarrollo de un software utilizando la metodología RUP. Obteniendo una aplicación web que permite representar geográficamente los distintos locales de los organismos e instituciones del país, destinados para la recreación. Además de poder realizar análisis estadísticos sobre la información referente a dichos locales.

Índice

Introducción	9
CAPÍTULO 1: Desarrollo de Sistema de Información Geográfica sobre GeneSIG	14
1.1 Conceptos asociados al dominio del problema	14
1.2 GeneSIG	18
1.2.1 CartoWeb	18
1.3 Plugins	19
1.3.1 Estructura de directorios de un Plugin	20
1.3.2 Interfaces	21
1.3.3 Mecanismos de Comunicación.....	21
1.3.4 Integración de un nuevo plugin a CartoWeb	23
1.3.5 Plugins y Coreplugins existentes	24
1.4 Conclusiones parciales.....	24
CAPÍTULO 2: Tendencias y tecnologías para el desarrollo del Sistema de Información Geográfica.	26
2.1 Aplicación informática.....	26
2.2 Servidor Web	26
2.3 Servidor de Mapas	27
2.4 Sistema Gestor de Base de Datos Espaciales	28
2.5 Herramienta CASE de Desarrollo de Software	29
2.5.1 Visual Paradigm	29
2.6 Lenguaje Unificado de Modelado (UML 6.4).....	30
2.7 Metodología de desarrollo de software a utilizar	30
2.7.1 Rational Unified Process (RUP).....	31
2.8 Entorno de Desarrollo NetBeans	32
2.9 Lenguajes de Programación.....	33
2.9.1 Lenguajes de Programación del lado del Cliente	33
2.9.1.1 JavaScript.....	33
2.9.2 Lenguajes de Programación del lado del Servidor.....	34
2.9.2.1 PHP	34
2.11 Técnica de desarrollo web.....	35
2.12 Conclusiones parciales.....	35
CAPÍTULO 3: Presentación de la solución propuesta.....	36
3.1 Entorno donde trabajará el sistema	36

3.1.1 Eventos principales del entorno	36
3.1.2 Diagrama de clases del modelo de dominio	37
3.1.3 Glosarios de términos del dominio	37
3.2 Requisitos funcionales	38
3.3 Requisitos no funcionales	39
3.4 Descripción del sistema propuesto	41
3.4.1 Descripción de los actores.....	41
3.4.2 Diagrama de casos de uso del sistema.....	41
3.4.3 Descripción textual de los casos de uso del sistema.....	42
3.4.3.1 Crear mapa temático.....	42
3.5 Conclusiones parciales.....	48
CAPÍTULO 4: Construcción de la solución propuesta	49
4.1 Arquitectura Propuesta	49
4.2 Modelo de diseño	50
4.2.1 Patrones de diseño	51
4.2.2 Diagrama de clases de diseño.....	52
4.3 Principios de diseño.....	53
4.3.1 Estándares de la interfaz de la aplicación	53
4.4 Diseño de la Base de Datos	54
4.4.1 Diagrama de Clases Persistentes.....	54
4.4.2 Diagrama Entidad-Relación	54
4.5 Diagrama de despliegue.....	55
4.6 Modelo de implementación.....	55
4.6.1 Diagrama de Componentes	56
4.7 Conclusiones parciales.....	57
Conclusiones	58
Recomendaciones	59
Bibliografía	60
Trabajos citados	64
Glosario de Términos	66

Introducción

Una de las grandes revoluciones que se han producido en los últimos siglos en la historia de la humanidad es la Tecnológica o de la Información. Esta se ha multiplicado como nunca antes, de manera que nos encontramos ya no sólo con las que podríamos considerar como tecnologías tradicionales, sino también con las denominadas nuevas tecnologías. Las Tecnologías de la Información y las Comunicaciones (TIC), se han convertido en elemento determinante para la concreción y puesta en acción de un nuevo modelo de sociedad. Este modelo está denominado de la información o del conocimiento, evolucionando a medida que el desarrollo del conocimiento se ha puesto al servicio de las crecientes necesidades sociales.

Sin duda alguna, no se podría comparar un sistema automatizado de los que existen actualmente con uno desarrollado años atrás; la diferencia es notable en cuanto a diseño, rapidez, eficacia. Orientados al perfeccionamiento de herramientas en aplicaciones creadas con el objetivo de cumplir un propósito determinado, se experimentan mejorías que puedan extender la utilización del software con nuevas funcionalidades. De esta manera permiten satisfacer las peticiones de usuarios que a nivel mundial van generalizándose a la vez que aparecen nuevas metas para quienes trabajan en función de la informatización.

Las tendencias actuales a las que se exponen las TIC están encaminadas a controlar o manejar las disímiles vías de información existentes en todos los sectores de la sociedad y la economía. Estos datos por su importancia y valor llegan a ser extensos, desorganizados y complejos en cuanto a su interpretación. Es por todo esto que la acción de gestionarlos digitalmente hoy en día se ha convertido en todo un reto para las tecnologías.

Para elevar el nivel de vida y cubrir todas las demandas que inciden en el mundo actual, donde las computadoras y el hombre mantienen un estrecho vínculo, se hace evidente la búsqueda de soluciones tangibles. Así surgen los Sistemas de Información Geográfica (SIG), los cuales se han convertido en una de las más importantes herramientas de trabajo para investigadores, analistas y planificadores en todas sus actividades. Además, tienen como insumo el manejo de la información (Bases de Datos) relacionada con diversos niveles de agregación espacial o territorial, esencial para el análisis y toma de decisiones en muchas áreas vitales para el desarrollo económico y social.

Los SIG son una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, y construido para satisfacer necesidades concretas de información. En el sentido más estricto, es un sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología de los SIG es más que un software; es también una industria importante y una disciplina académica. Desde su origen hace 30 años se han utilizado en investigaciones científicas, en la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing y la logística. Un SIG podría permitir a los grupos de emergencia calcular fácilmente los tiempos de respuesta en caso de un desastre natural. Pueden ser usados para encontrar los humedales que necesitan protección contra la contaminación. También utilizados por una empresa para ubicar un nuevo negocio y aprovechar las ventajas de una zona de mercado con escasa competencia.

En la actualidad, Cuba se encuentra centrada en un proceso de cambios en el ámbito tecnológico, dirigidos a lograr grandes resultados a nivel mundial en el desarrollo de este campo. Dentro de muchas de estas tecnologías, se encuentran las relacionadas con el desarrollo de los SIG y su vinculación y adaptación con las alternativas libres (software libre) existentes en el mundo.

Producto de la batalla que se lleva a cabo en el país por la informatización surge la Universidad de las Ciencias Informáticas (UCI). La cual se ha unido desde hace algunos años a los programas con que cuenta Cuba para alcanzar un logro inmediato y superior al existente en el sector informático. Se está dedicando tiempo, costo y esfuerzo para la creación de diversos productos que vayan ayudando a lograr un avance en el desarrollo de la informática.

En la facultad 6 de la UCI se encuentra ubicado el centro de desarrollo Geoinformática y Señales Digitales (GEYSED). Este tiene como misión desarrollar soluciones informáticas que contribuyan a la formación integral de profesionales que respondan a

las necesidades del progreso científico-técnico y socioeconómico, permitiendo una posición en el mercado nacional. El mismo posee dos áreas del conocimiento bien definidas internacionalmente: la Geoinformática y el procesamiento de Señales Digitales.

Dentro de este centro se encuentra el Departamento Geoinformática, distinguido por la asunción de desarrollos de SIG bajo tecnologías OpenSource y cumpliendo con los estándares OpenGIS. Incluye la implementación de soluciones informáticas para la toma de decisiones empresariales basadas en SIG, desarrollo de soluciones sobre bases de datos espaciales, implementación de servicios de mapas, administración de metadatos geográficos y soluciones para dispositivos móviles. Además, agrupa proyectos dedicados a la conceptualización y desarrollo de soluciones informáticas para la geología y la minería, con temáticas de bases de datos, portales de entidades geológicas y tratamiento de información asociada al patrimonio geológico.

Dentro de los resultados del arduo trabajo de este departamento se encuentra la Plataforma Soberana para el Desarrollo de Sistemas de Información Geográfica GeneSIG, la cual constituye una herramienta informática ideada por un grupo de especialistas de GEOCUBA, las Fuerza Armadas y la UCI. Tiene como objetivo permitir la representación geoespacial de la información asociada a cualquier negocio que lo requiera. Una muestra de ello lo constituye SIG-UCI, así como otras aplicaciones que están en desarrollo con varios clientes tanto nacionales como extranjeros.

Dentro de las organizaciones nacionales que han mostrado interés por el uso de estas tecnologías, se encuentra la Unión de Jóvenes Comunistas (UJC). Dentro de sus prioridades políticas e ideológicas, también podemos encontrar la de hacer todo lo posible por garantizar ofertas recreativas para el disfrute de la familia cubana. Esto se hace posible a través de una variedad de instalaciones de diferentes organismos dedicadas a la recreación en la etapa veraniega. Actualmente se les dificulta obtener todo tipo de información sobre estas instalaciones, para realizar balances estadísticos que ayuden a la toma de decisiones en aras de lograr una mejor distribución de las diferentes instalaciones y ofertas recreativas.

La UJC hasta el momento cuenta solamente con un documento Excel donde tiene guardado toda una serie de datos sobre los centros recreativos de los diversos organismos e instituciones del país y su cantidad por cada una de las provincias. Debido a esto, el trabajo se les dificulta a la hora de manipular dicha información, ya sea para la recolección de estadísticas sobre cuáles son las ofertas con más

demanda, ubicación geográfica de los locales, para realizar análisis de información socioeconómica a partir de datos específicos o para mantener actualizada toda la información. Además, para modificar o recolectar cualquier tipo de información hay que remitirse a la UJC a nivel de municipio, después a nivel provincial, donde se congregan todos los datos y posteriormente se llevan al documento Excel para su posterior revisión y análisis.

Debido a lo antes planteado se ha identificado el siguiente **problema a resolver**: ¿Cómo georreferenciar los distintos locales de los organismos e instituciones del país destinados para la recreación? Como **objeto de estudio**: el proceso de desarrollo de un plugin para CartoWeb, por ende, el **campo de acción** quedaría enmarcado como el proceso de personalización de la plataforma GeneSIG para la Oficina Nacional de la UJC.

Se persigue con esta investigación lograr el siguiente **objetivo general**: Personalizar la Plataforma GeneSIG para la Oficina Nacional de la UJC. Para lograr este objetivo general se han definido los siguientes **objetivos específicos**:

- Modificar la interfaz de usuario.
- Implementar el plugin de tematización.

Se plantea como **idea a defender** que: La personalización de la Plataforma GeneSIG para la Oficina Nacional de la UJC facilitará la representación de toda la información socioeconómica y su análisis.

Los **métodos científicos** a utilizar son:

Métodos Teóricos:

- Análisis histórico lógico: Se utilizó para el análisis histórico de los procesos de gestión de información en la UJC.
- Análisis síntesis: Se empleó para el análisis de las tendencias y tecnologías actuales para el desarrollo de plugins sobre CartoWeb.

Como **posible resultado** de la investigación se espera:

1. Aplicación SIG para la Oficina Nacional de la UJC.
2. Documentación técnica del proceso de desarrollo.
3. Documento de tesis.

Estructura de la investigación:

La presente investigación científica cuenta con una estructura de 4 capítulos que serán detallados con las siguientes cualidades:

El **Capítulo 1** contempla la fundamentación teórica de esta investigación, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, se especifican detalladamente todos los argumentos que esclarecen el objeto de estudio y se explican de manera sintetizada otras soluciones existentes.

En el **Capítulo 2** se especifican las herramientas y/o plataformas tecnológicas (hardware, software, equipamiento, etc.) necesarias para instalar la solución tomando en cuenta los requisitos macros del SIG. A partir del estudio de diferentes metodologías de desarrollo de software se selecciona y argumenta la que mejor se ajusta a la propuesta de solución a la cual se desea arribar.

Por su parte en el **Capítulo 3** se realiza un levantamiento de los requisitos funcionales y no funcionales del sistema con el fin de lograr los objetivos planteados. A partir de la recopilación de los requisitos funcionales se construye el Modelo de Casos de Uso del Sistema.

En el **Capítulo 4** se realizan los diagramas de colaboración del análisis, diagramas de secuencia del diseño correspondientes a los casos de uso identificados, diagrama de Despliegue y Modelo de implementación. Se efectúan las pruebas de caja blanca y caja negra. Para finalizar a partir del posible resultado, se realiza la validación de la propuesta de solución.

CAPÍTULO 1: Desarrollo de Sistema de Información Geográfica sobre GeneSIG

En el presente capítulo son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, entre los cuales se pueden encontrar: Sistemas de Información, Información geográfica, Sistemas de Información Geográfica, Plataforma, Tecnologías de la información y las Comunicaciones, Datos espaciales, Mapa, Plugin, CartoWeb, Entorno libre. Se especifican detalladamente todos los argumentos que esclarecen el proceso de desarrollo de un plugin para CartoWeb y se mencionan de manera sintetizada otras soluciones existentes.

1.1 Conceptos asociados al dominio del problema

Con el objetivo de que el lector pueda tener una comprensión mayor de los temas que serán abordados en este capítulo, directamente relacionados con el objeto de estudio de la investigación, se describen detalladamente a continuación un grupo de conceptos asociados al dominio del problema.

Sistemas de Información

Es un conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización.

Hay tres actividades en un sistema de información que producen la información necesaria para la toma de decisiones, controlar operaciones, analizar problemas y crear nuevos productos o servicios. Estas actividades son:

Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.

Procesamiento: convierte esa entrada de datos en una forma más significativa.

Salida: transfiere la información procesada a la gente que la usará o a las actividades para las que se utilizará.

Los sistemas de información también requieren retroalimentación, que es la salida que se devuelve al personal adecuado de la organización para ayudarle a evaluar o corregir la etapa de entrada.

Información geográfica

Se denomina Información Geográfica (IG) a aquellos datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos datos espaciales suelen llevar una información alfanumérica asociada. Se estima que el 80% de los datos corporativos existentes en todo el mundo poseen esta componente geográfica (Gianfelici, 2008).

Sistemas de Información Geográfica

Es un sistema computarizado diseñado para permitir a los usuarios coleccionar, manejar y analizar grandes volúmenes de datos de atributo asociados y espacialmente referidos. El Sistema de Información Geográfica (SIG) se utiliza para resolver investigaciones complejas, para los problemas de manejo, y para la planeación. Los componentes principales del SIG son: una interfaz del usuario, la capacidad para el manejo de la base de datos del sistema, la creación de base de datos, la capacidad para la captura de datos, el paquete de la manipulación y el análisis de datos espaciales.

Un SIG puede mapear cualquier información almacenada en planillas o bases de datos, que tenga un componente geográfico que permita ver patrones, relaciones y tendencias, que no pueden verse en un formato de tabla o lista. Da una perspectiva totalmente nueva y dinámica de la información, y ayuda a tomar mejores decisiones. La gran popularidad que se ha desatado en los últimos años en cuanto al uso de los SIG y su eficacia en la toma de decisiones incurre en el propósito de dar a conocer un compendio de algunos criterios y definiciones que se publican a continuación.

“Un sistema de hardware, software y procedimientos, diseñado para realizar la captura, almacenamiento, manipulación, análisis, modelación y presentación de datos referenciados espacialmente para la resolución de problemas complejos de planificación y gestión” (NCGIA, 1990).

“Una tecnología de la información que almacena, analiza y presenta datos espaciales y no espaciales” (Parker, 1988).

“Conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos” (Obando).

En otra palabras, los SIG son tecnologías de la información que comprenden un conjunto de métodos, herramientas y un sistema de base de datos georreferenciados que permite almacenar, analizar, modelar, manipular y transformar la información

geográfica o datos espaciales, con el propósito de resolver varios problemas relacionados con la gestión y la planificación, influyendo considerablemente en la toma de decisiones.

Plataforma

Es precisamente el principio, en el cual se constituye un hardware, sobre el cual un software puede ejecutarse o desarrollarse. El hardware o software subyacente para un sistema. La plataforma define un estándar alrededor del cual un sistema puede ser desarrollado. Una vez que ha sido definida, los desarrolladores de software pueden producir el software apropiado y los gerentes pueden comprar el hardware apropiado para su uso. El término a menudo es usado como un sinónimo de sistema operativo. El término plataforma cruzada se refiere a la capacidad de software o hardware para correr de modo idéntico sobre plataformas diferentes.

La informática de plataforma cruzada se hace cada vez más importante a medida que las redes de área local se perfeccionan en la integración de las máquinas de diferentes plataformas. Por ejemplo, el ambiente de programación de plataforma cruzada permite a un programador desarrollar programas para muchas plataformas simultáneamente.

Tecnologías de la Información y la Comunicaciones (TICs)

El concepto de TICs surge como convergencia tecnológica de la electrónica, el software y las infraestructuras de telecomunicaciones. La asociación de estas tres tecnologías da lugar a una concepción del proceso de la información, en el que las comunicaciones abren nuevos horizontes y paradigmas. Agrupan también un conjunto de sistemas, herramientas y soportes necesarios para el tratamiento y acceso a la información, especialmente los ordenadores y programas necesarios para convertirla, almacenarla, administrarla, transmitirla y encontrarla.

Datos espaciales

Los datos espaciales son informaciones sobre la localización, las formas de un objeto geográfico y las relaciones entre ellos, normalmente con coordenadas y topología.

Los datos geográficos son entidades espacio-temporales que cuantifican la distribución, el estado y los vínculos de los distintos fenómenos u objetos naturales o sociales.

Los datos espaciales se refieren a entidades o fenómenos que cumplen los siguientes principios básicos:

- Tienen posición absoluta: sobre un sistema de coordenadas (x, y, z).

- Tienen una posición relativa: frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado, etc.).
- Tienen una figura geométrica que las representan (punto, línea, polígono).
- Tienen atributos que lo describen (características del elemento o fenómeno).

Mapa

Se trata de una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que suele ser plana, o esférica como en el caso de los globos terráqueos.

Plugin

Un plugin es un programa adicional que puede ser añadido a una aplicación para aumentar la funcionalidad de esta. Este programa adicional es cargado y corrido por la aplicación principal, permitiéndoles a los usuarios finales añadir y actualizar los complementos de forma dinámica sin necesidad de hacer cambios a la aplicación principal.

Entorno libre o Software libre.

El Software Libre se refiere a la libertad de los usuarios de correr, copiar, distribuir, estudiar, cambiar y mejorar el software. Se refiere a las cuatro libertades de los usuarios de software:

1. La libertad de correr el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades. El acceso al código fuente es una precondición para esto.
3. La libertad de distribuir copias de manera que se puede ayudar a los demás.
4. La libertad de mejorar el programa, y liberar las mejoras al público de tal manera que toda la comunidad se beneficia. El acceso al código fuente es una precondición para esto” (Colectivo de autores).

Estas libertades son derechos, no obligaciones. Toda persona es libre de no usar estas libertades, pero también podría elegir el uso de todas ellas.

Es importante conocer que el software libre no excluye el uso comercial ni la distribución comercial, ya que no es software gratuito. El software libre hace legal proveer ayuda y asistencia, pero no lo hace imprescindible.

En este sentido el software libre, es a menudo la mejor opción para la empresa, ya no en términos económicos, debido a que por ejemplo la implementación, ayuda o

asistencia para software libre generalmente es más barata, sino en términos de control y/o despersonalización.

Otra gran ventaja es que la mayoría de los programas de software libre tienen una política de mejora basada en contribuciones de personas no relacionadas directamente con la creación y desarrollo del programa. Cuando se crea un módulo o herramienta con código fuente público hace posible que otros usuarios lo utilicen o mejoren.

1.2 GeneSIG

La plataforma GeneSIG está implementada sobre herramientas y tecnologías libres, cumpliendo además con las especificaciones OpenGIS que establece el Open Geospatial Consortium (OGC) que garantizan la interoperabilidad global entre los SIG y en consecuencia con la política de migración a software libre y de soberanía tecnológica que impulsa Cuba.

Es un producto encaminado a realizar la representación y análisis geoespacial de información geográfica, servicios que son proporcionados mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnología SIG. Además su estructura arquitectónica permite personalizar sus funcionalidades a cualquier negocio que lo requiera a través de la reutilización de sus componentes. La base arquitectónica de GeneSIG se ha modelado haciendo uso del patrón arquitectónico AOO (Arquitectura Orientada a Objetos), ya que el framework que se utiliza para el desarrollo es CartoWeb y este posee una arquitectura orientada a objetos.

1.2.1 CartoWeb

CartoWeb 3.0 es una aplicación construida en PHP¹ sobre UMN MapServer que explota AJAX². Su característica más diferenciadora respecto a otros proyectos de clientes Web ligeros sobre MapServer, es que ofrece un framework que ha sido diseñado con una arquitectura bastante modular y escalable, permitiendo separar la lógica de un servidor (cartoserver) encargado del diálogo con el servidor de mapas y la provisión de servicios a un cliente (cartoclient). (Espinosa, 2008)

CartoWeb consta de dos módulos principales: CartoServer el cual procesa los datos y se comunica con MapServer, y CartoClient que es el encargado de manejar los vínculos con el cliente, recepción de solicitudes y el formato de las respuestas. Sin

¹ Hypertext Pre-Processor.

² Asynchronous JavaScript And XML

embargo CartoWeb puede funcionar como un único módulo. También está construido sobre un conjunto de ladrillos que interactúan entre sí. Estos ladrillos son nombrados plugins, los cuales realizan tareas como la navegación por el mapa, gestión de capas, permisos de usuarios y la exportación.

1.3 Plugins

Existen dos tipos de plugin en CartoWeb (coreplugin y plugin), los coreplugin son aquellos que manipulan el sistema a bajo nivel, principalmente para el trabajo con los mapas, herramientas de búsqueda y selección de capas; son cargados por defecto una vez que la aplicación comience a ejecutarse, generalmente son utilizados por los demás plugin, a pesar de esto la estructura interna de ambos es la misma y la gran diferencia radica en su configuración.

Para activar un plugin es necesario configurar tres ficheros fundamentales:

client.ini: Es el encargado de informarle al CartoWeb que módulo cliente de plugins debe cargar cuando este se inicie, dicha información se debe especificar en la propiedad loadPlugins, es importante resaltar que si el plugin no consta de este módulo no se debe incluir, porque de ser así daría error.

Conf: loadPlugins= <nombre del nuevo plugin>

Path: ../cartoweb3/projects/geoweb/client_conf/client.ini

geoweb.ini: Es el encargado de informarle al CartoWeb que el módulo servidor de plugins debe cargar cuando este se inicie, dicha información se debe especificar en la propiedad mapInfo.loadPlugins, es importante resaltar que si el plugin no consta de este módulo no se debe de incluir, porque de ser así daría error.

Conf: mapInfo.loadPlugin= <nombre del nuevo plugin>

Path: ../cartoweb3/projects/geoweb/server_conf/geoweb/geoweb.ini

cartoclient.tpl: Es la plantilla principal del proyecto, en este fichero se debe especificar todos los js que se utilizarán en la aplicación.

Conf: <script type="text/javascript" src="../file.js"></script>

Path: ../cartoweb3/projects/geoweb/templates/cartoclient.tpl

Independiente del tipo de plugin pueden tener tanto para el módulo servidor como el cliente un fichero de configuración en el cual se pueden inicializar las variables que sean necesarias.

Client Path: ../cartoweb3/projects/geoweb/client_conf/plugin_name.ini

Server Path: ../cartoweb3/projects/geoweb/server_conf/geoweb/plugin_name.ini

Para acceder a las variable definidas en los ficheros ".ini " se utiliza la llamada a la función ClientPlugin::getConfig() o ServerPlugin::getConfig() en dependencia del módulo en que se esté programando, la sintaxis es:

```
$resultado = $this->getConfig()-><nombre de la variable>;
```

1.3.1 Estructura de directorios de un Plugin

A continuación se hará una breve descripción de los directorios:

- client/ contiene todos los componentes de extensión php del plugin que implementan el módulo CartoClient del CartoWeb.
- server/ contiene todos los componentes de extensión php del plugin que implementan el módulo CartoServer del CartoWeb.
- common/ contiene todos los componentes que implementan aquellas clases que servirán de puente para la comunicación entre client.php y server.php, incluyendo el fichero *.wsdl.inc, donde se especifica cómo acceder a estas vía SOAP³.
- templates/ contiene específicamente el fichero *.tpl basado en el generador de plantillas Smarty para el procesamiento de datos.
- htdocs/gfx/ contiene las imágenes que serán utilizadas en el plugin.
- htdocs/css/ contiene las hojas de estilo que serán utilizadas en el plugin.
- htdocs/js/ contiene todos los componentes de extensión js del plugin que implementan el módulo cliente en javascript del CartoWeb.

³ Simple Object Access Protocol

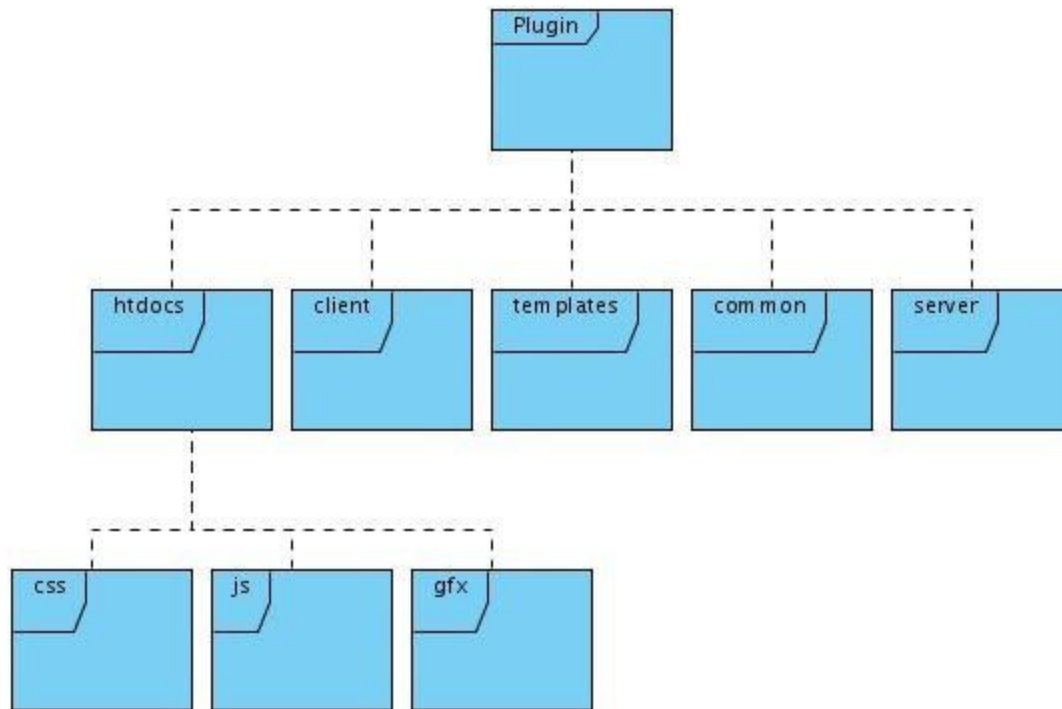


Figura 1: Componentes básicos de un plugin

1.3.2 Interfaces

En la creación de un plugin las interfaces juegan un papel fundamental, pues proveen a los diferentes módulos, mecanismos de comunicación entre ellos. Entre las principales interfaces que deben ser implementadas en el módulo cliente, se encuentran:

- GuiProvider : Para la comunicación con Smarty.
- ServerCaller : Para la comunicación con el módulo Servidor.
- Ajaxable : Para la comunicación con Ajax.

y para el módulo Servidor:

- ClientResponderAdapter : Para la comunicación con el módulo Cliente.

1.3.3 Mecanismos de Comunicación

CartoWeb provee un mecanismo de comunicación entre plugin, permitiendo que se puedan reutilizar los componentes que se hayan desarrollado de forma simple y sin necesidad de aumentar el acoplamiento de clases. Accediendo desde la propiedad `serverContext` a la función `ServerContext::getPluginManager()` es posible obtener el listado de los objetos servidores de cada uno de los plugin que se encuentran activos.

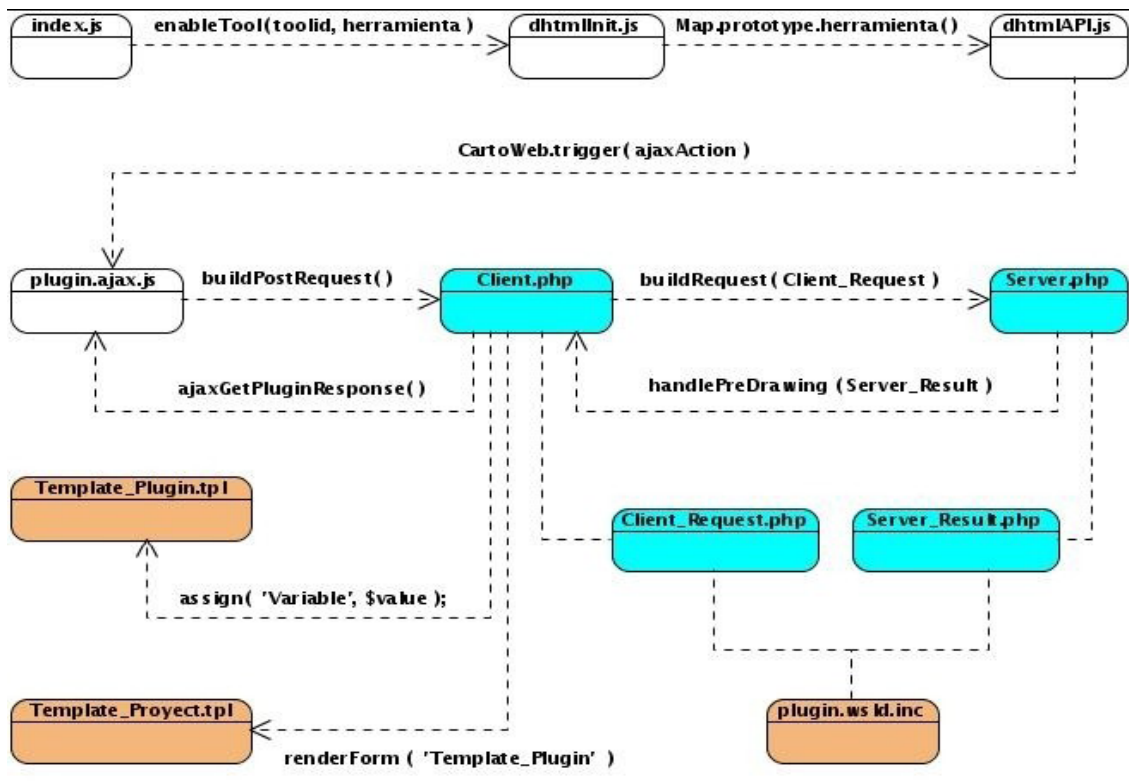


Figura 2: Mecanismo de Comunicación

Comunicación Client.php con Smarty

Para enviar valores a las variables declaradas en la tlp local del plugin, se debe de crear un objeto smarty “\$smarty = new Smarty_Plugin(\$this>getCartoclient(), \$this)”, una vez creado, se utiliza la función “assign” pasando por parámetro el nombre de la variable de smarty y el valor que le será asignado:

```
“$smarty->assign( 'Variable', $valor )”
```

Comunicación Client.js con Client.php

La comunicación entre el cliente javascript y cliente php se establece a través de Ajax, por tanto, desde el Client.js se pueden enviar todo los valores de los componentes que se encuentren dentro del formulario que está definido en la plantilla principal del proyecto una vez que se halla ejecutado el Ajaxaction, para ello es necesario redefinir la propiedad buildPostRequest del mismo y especificarle que ejecute la función encargada de recoger todos los valores y enviarlo al client.php “AjaxHandler.buildPostRequest()”, en dependencia del método seleccionado para el envío.

También es posible enviar datos desde el Client.php al Client.js, para ello se debe redefinir la función Ajaxable::ajaxGetPluginResponse(AjaxPluginResponse \$ajaxResponse), utilizando la función “addHtmlCode” del objeto “AjaxPluginResponse”

que se le pasan dos parámetros, el primero es el nombre de la variable javascript y el segundo es el valor que recibirá la misma, el cual será capturado dentro de la propiedad "handleResponse" del AjaxPlugins.<nuevo plugin>.

Comunicación Client.php con Server.php

La comunicación entre el módulo cliente y servidor se establece mediante SOAP, se deben crear dos clases contenedoras de la información que se desea enviar y estas a su vez deben ser especificadas en el fichero "<nuevo plugin>.wsdl.inc" que se encuentra en el directorio "common/", donde la clase ClientRequest sería entonces la que contendría todas las variables que se desee enviar desde client.php a server.php. Para ello se redefine el método ServerCaller::buildRequest() con la finalidad de que retorne una instancia de ClientRequest con sus valores pertinentes. Estos datos son recibidos en el módulo servidor una vez que se redefine el método ClientResponderAdapter::handlePreDrawing(\$ClientRequestObj), el cual recibe por parámetro la instancia de ClientRequest enviada por el client.php, permitiendo procesar dicha información.

Para enviar los datos que han sido procesados por el servidor es necesario redefinir el método ClientResponderAdapter::handlePreDrawing(\$ClientRequestObj) en función de retornar una instancia de la clase ServerResult, contenedora de toda la información que se desea enviar desde el server.php al client.php.

Un elemento importante a tener en cuenta, es que tanto ClientRequest como ServerResult deben heredar de la clase CwSerializable y redefinir el método unserialize(), con el objetivo de definir cada una de las variables que estas tengan definidas, de lo contrario no se efectuaría la comunicación vía SOAP.

1.3.4 Integración de un nuevo plugin a CartoWeb

CartoWeb ha sido diseñado con una arquitectura bastante modular y escalable, la cual facilita el desarrollo de sus propios plugins. Para integrar un nuevo plugin a CartoWeb, se deben configurar los ficheros client.ini y geoweb.ini como se explica a continuación:

Paso 1: Configurar el fichero "client.ini" para especificarle a la aplicación GeoWeb que debe cargar el módulo cliente del plugin "Perfil" una vez que esta se inicialice.

```
../cartoweb3/projects/geoweb/client_conf/client.ini => línea: 16
```

```
loadPlugins = bboxHistory, perfil
```

Paso 2: Configurar el fichero “geoweb.ini” para especificarle a la aplicación GeoWeb que debe cargar el módulo servidor del plugin “Perfil” una vez que esta se inicialice.

```
../cartoweb3/projects/geoweb/server_conf/geoweb/geoweb.ini =>linea: 4
```

```
mapInfo.loadPlugins = hilight, layerReorder, exportPdf, edit, perfil
```

1.3.5 Plugins y Coreplugins existentes

Plugins

Autenticación: Permite la gestión de los derechos, permisos y usuarios.

ExportCsv: Permite exportar los resultados de la consulta a un archivo CSV.

ExportHtml: Para exportar el mapa como HTML⁴.

ExportPdf: Para exportar el mapa en formato PDF.

Hilight: Gestiona la identificación de los objetos seleccionados en el mapa.

Esquema: Autoriza a las líneas de trama, puntos o polígonos en el mapa.

Coreplugins

Imágenes: Establece el formato de presentación de la tarjeta principal.

Capa: Permite describir la estructura de capas de información.

Situación: Describe y establece las herramientas de navegación geográfica (acercar, alejar, mover).

Mapquery, consultas, tablas: Se refieren a la configuración de plugins y funcionamiento de las aplicaciones.

Statictools: Permite la medición de superficies y distancias.

1.4 Conclusiones parciales

Con el contenido abordado en este capítulo se fundamentaron los componentes teóricos que sostienen esta investigación, logrando una mejor visión del problema. GeneSIG utiliza como framework para el desarrollo CartoWeb, el cual proporciona un marco básico para el desarrollo de aplicaciones basadas en MapServer. CartoWeb está construido sobre plugins los cuales realizan funciones como el trabajo con los mapas, herramientas de búsqueda y selección de capas. A partir del estudio realizado

⁴Hypertext Markup Language

en este capítulo se puede proceder a seleccionar las tecnologías necesarias para el desarrollo del componente de software.

CAPÍTULO 2: Tendencias y tecnologías para el desarrollo del Sistema de Información Geográfica.

En este capítulo se presentan las tendencias actuales de desarrollo y las tecnologías usadas para el desarrollo de los plugins en la personalización de GeneSIG.

2.1 Aplicación informática

Una aplicación informática puede clasificarse en Aplicación Web o Aplicación de Escritorio. Esta no es más que un programa que sirve como interfaz entre el usuario y la computadora. Generalmente son diseñadas para dar respuesta a problemas determinados en un tiempo relativamente corto.

Partiendo de las necesidades planteadas por parte de la Oficina Nacional de la UJC, se puede deducir que se demanda el desarrollo de una aplicación informática de tipo WEB, puesto que se hace necesario que dicho sistema sea centralizado, para evitar la duplicación de información y hacer más fácil la actualización y recogida de los datos.

Ventajas y Beneficios de una Aplicación WEB:

- Se facilita el trabajo a distancia.
- No se requieren complicadas combinaciones de Hardware y Software para utilizar estas aplicaciones, solo un computador con un buen navegador Web.
- Es multiplataforma ya que se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.
- Teniendo en cuenta que el sistema se encuentra centralizado y ubicado en un servidor WEB al cual acceden los usuarios, permite que sea fácil su actualización, de esta manera todos los usuarios que accedan a ella hacen uso de la última versión del sistema.

2.2 Servidor Web

Un servidor web es un programa que permite a los navegadores web recuperar archivos de ordenadores conectados a Internet. El servidor web escucha las solicitudes de los navegadores web y al recibir una solicitud de un archivo lo envía al

navegador. La función principal de un servidor web es proporcionar páginas a otros ordenadores remotos; por consiguiente, debe estar instalado en un ordenador que esté conectado a una red. También supervisa y registra las estadísticas de acceso al servidor (1979-2010).

Apache 2.2 es considerado el servidor web por excelencia pues desde su surgimiento ha demostrado que es estable y que tiene mejor rendimiento que sus competidores. Además brinda algunas ventajas que son fundamentales para el desarrollo Web como son:

- **Fiabilidad:** Más del 90% de los servidores con más alta disponibilidad funcionan bajo un servidor Apache.
- **Software libre:** El servidor Apache es totalmente gratuito y es distribuido bajo la licencia de Apache en la cual se permite realizarle cambios al código fuente.
- **Extensibilidad:** Se pueden añadir módulos para ampliar aún más las capacidades del servidor (autores, 1997-2005).

Su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es uno de los mayores triunfos del software libre. Entre sus características fundamentales se pueden resaltar: multiplataforma, flexible, rápido y eficiente, se desarrolla de forma abierta, es modular, trabaja con gran cantidad de lenguajes de script como son PHP, Perl, y otros (autores, 2010).

2.3 Servidor de Mapas

Para poder hacer uso de una aplicación Web se requiere de un servidor Web que dé respuesta a las peticiones realizadas desde el cliente, ya que este permite la interacción con la información espacial almacenada en servidores de datos espaciales accesibles vía web.

En el mundo existen diferentes servidores de mapas, algunos propietarios y otros basados en software libre, algunos de estos son: MapServer, MapGuide, Geoserver, ArcIMS, Geomedia y Manifold; estos tres últimos pertenecientes al software propietario.

Para el desarrollo del sistema se seleccionó el servidor de mapas MapServer 5.4.1, teniendo en cuenta que es una aplicación desarrollada sobre software libre, es de fácil instalación y configuración.

Entre las características y ventajas más destacadas de MapServer se encuentran:

- Velocidad de acceso a datos.
- Fácil configuración y administración.

- Es multiplataforma, ya que puede ser ejecutado sobre sistemas operativos Linux, Windows, Mac OS X, Solaris.
- Permite la representación de mapas tanto de tipo Vectorial como Raster.
- Tiene una amplia integración con sistemas gestores de base de datos tanto propietarios como libres.

2.4 Sistema Gestor de Base de Datos Espaciales

Un Sistema Gestor de Base de Datos (SGBD) es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Estos sistemas se componen de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Permiten a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos garantizando además su seguridad.

Para el desarrollo de un SIG, uno de los componentes principales es contar con una base de datos donde se guarde la información asociada a los objetos que se mostrarán sobre el mapa. Estas bases de datos tienen una diferencia con relación a las convencionales, y es que debe posibilitar el almacenamiento de datos espaciales, permitiendo luego su georreferenciación. Para el desarrollo de este sistema se ha elegido como SGBD a PostgreSQL 8.4 haciendo uso del módulo PostGIS 1.5.1 para el trabajo con datos espaciales.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y es libre, publicado bajo la licencia BSD. Este tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Una de sus principales características es la alta concurrencia.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último que se actualizó. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

El módulo PostGIS añade entidades geográficas al PostgreSQL haciendo que soporte geometrías espaciales, sin embargo el PostGIS añade la capacidad de almacenamiento y recuperación según la especificación SFS (Simple Features Specification) del consorcio internacional Open GeoSpatial (OGC). Además del almacenamiento de datos geográficos. Este módulo también implementa diversas funcionalidades topológicas, posibilitando el desarrollo de Sistemas de Información Geográfica (SIG). La topología también forma parte de la especificación SFS

(OpenGIS), garantizando al PostGIS interoperabilidad con incontables sistemas que también adoptan el SFS.

Algunas de las principales funciones del PostGIS son:

- Disjoint: Se analiza si dos geometrías poseen algún punto en común.
- Intersects: Analiza si dos geometrías poseen alguna intersección.
- Crosses: Verifica si dos geometrías se cruzan.
- Within: Analiza si una geometría está contenida en la otra.
- Contains: Analiza si una geometría contiene a la otra.

Algunas ventajas de PostGIS:

- Es un producto liberado bajo licencia GNU/GPL.
- Brinda posibilidad de acceso simultáneo a una misma información.

2.5 Herramienta CASE de Desarrollo de Software

Se puede definir a las Herramientas CASE ⁵ como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación). CASE es también definido como el grupo de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

En otras palabras CASE es la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales. Se decide aplicarlas en el proyecto ya que representan una forma que permite modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información Gerenciales (Valle, 2005).

2.5.1 Visual Paradigm

Visual Paradigm 3.4 es una herramienta UML⁶ profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. También es multiplataforma para el modelado de aplicaciones. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

⁵ Computer Aided Software Engineering

⁶ Unified Modeling Language

Se selecciona dicha herramienta porque ayudará a realizar una rápida construcción de la aplicación con calidad; además presenta características que son favorables para el trabajo con tecnologías libres. Facilita a los ingenieros de software diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software. Permite construir sistemas de software a gran escala de manera confiable a través del uso de un enfoque Orientado al Objeto. Presenta un generador de código que soporta más de 10 lenguajes y proporciona la ingeniería inversa.

2.6 Lenguaje Unificado de Modelado (UML 6.4)

Es el lenguaje más conocido y utilizado en la actualidad para el diseño y modelado de sistemas de software. Este permite visualizar, especificar, construir y documentar, modelos de sistemas de software, incluyendo su estructura y diseño, que capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Es un lenguaje de propósito general para el modelado visual y orientado a objetos, que permite una abstracción del sistema y sus componentes, al mismo tiempo, posibilitando establecer una serie de requisitos y estructuras necesarias para plasmar en un sistema de software previo al proceso intensivo de escribir código (HERNANDEZ).

Se utiliza este lenguaje de modelado ya que presenta un conjunto de herramientas que permiten modelar (analizar y diseñar) sistemas orientados a objetos, es fácil de aprender y utilizar. Con la modelación de los artefactos durante las primeras fases de ciclo de vida del software, se posibilita que en fases posteriores los implementadores tengan un mayor dominio y una mejor comprensión sobre qué es lo que se debe implementar, permitiendo tanto al cliente como a los desarrolladores tener una representación real de los alcances y la factibilidad que puede o no llegar a tener el producto.

2.7 Metodología de desarrollo de software a utilizar

Todo desarrollo de software es riesgoso y difícil de controlar, por lo que si llevamos una metodología de por medio, obtenemos como resultado clientes satisfechos y desarrolladores aún más satisfechos. La tendencia de crear sistemas más sofisticados, adaptados a las nuevas tecnologías, a las necesidades de los usuarios que cambian constantemente, de mejorar los productos de una versión a otra, y de realizar todo este proceso de una forma más rápida, hace cada vez más complejo el proceso de desarrollo de software. Las metodologías guían todo el proceso de desarrollo de software, determinando pautas a seguir y haciendo hincapié en la planificación, con el objetivo de hacer más eficiente el proceso.

En la universidad existen varias metodologías de software que se emplean en los Proyectos Productivos de la universidad así como SCRUM, XP⁷ y RUP⁸. Llevar a cabo el uso de estas metodologías garantiza un producto final de calidad y el desarrollo de un proyecto en el tiempo establecido. Se debe realizar un análisis de las características de cada proyecto informático para determinar cuál es la más factible. Por la experiencia acumulada y su aplicación en el desarrollo de la plataforma GeneSIG, se utilizará RUP la cual brinda buenos resultados y un desarrollo del software efectivo.

2.7.1 Rational Unified Process (RUP)

La metodología RUP divide en 4 fases el desarrollo del software:

- **Inicio:** Determinar la visión del proyecto.
- **Elaboración:** Determinar la arquitectura óptima.
- **Construcción:** Obtener la capacidad operacional inicial.
- **Transmisión:** Obtener el resultado final del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requisitos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requisitos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- Despliegue: En esta fase el software es entregado a los usuarios finales y al equipo que realizará la operación y soporte del mismo.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.

⁷ Extreme Programming

⁸ Rational Unified Process

- Ambiente: Administrando el ambiente de desarrollo (Sanchez, 2004).

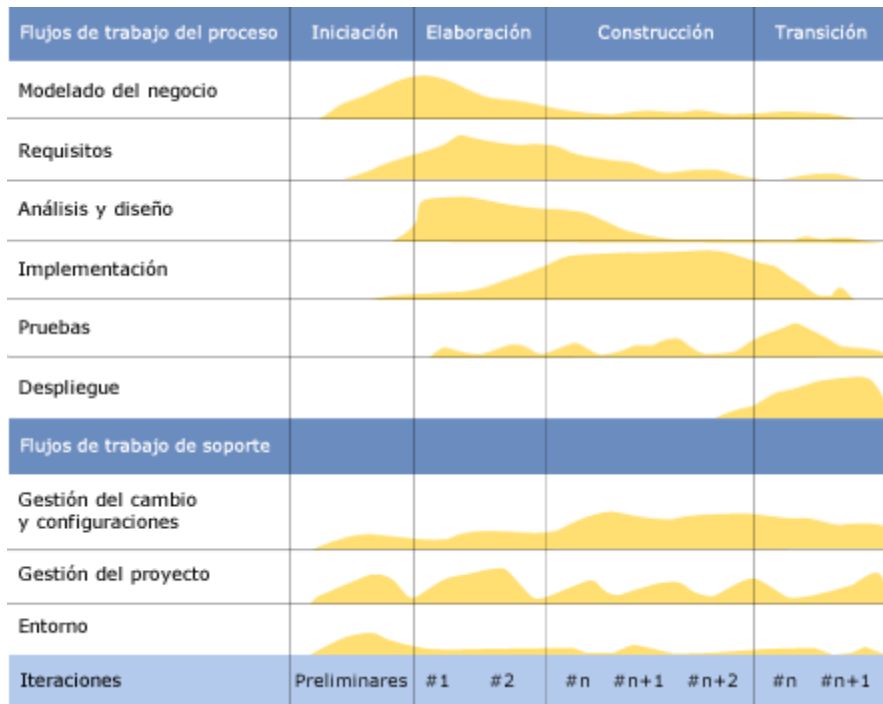


Figura 3: Fases e Iteraciones de la Metodología RUP

Las características de RUP, permiten que esta metodología sea adaptable a una gran variedad de sistemas para diferentes áreas de aplicación, diferentes tipos de organización y diferentes tamaños de proyecto. La particularidad de que cada ciclo de iteración exige el uso de artefactos, es el motivo que hace que sea una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Por parte de la dirección del proyecto, se decide optar por RUP como metodología de desarrollo, pues a pesar de ser una metodología adaptable, disminuye los riesgos, además de que la documentación generada es de gran utilidad para futuras actualizaciones del sistema.

2.8 Entorno de Desarrollo NetBeans

NetBeans 6.8 es una herramienta de desarrollo para aplicaciones, escrita puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores. Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

En una era en la cual la arquitectura orientada al servicio (SOA) requiere servicios con cierta relación que manejen procesos específicos del negocio, este satisface los requisitos con conjuntos de herramientas independientes de la plataforma, modulares y orientadas al objeto (autores, 2010).

Es un IDE de desarrollo maduro y con soporte a gran cantidad de librerías, lo cual lo convierte en un entorno de desarrollo ideal, por lo que permite a los programadores el uso de las mismas de una forma fácil e intuitiva. Es sencillo de utilizar y exige poco conocimiento por parte del usuario para su uso. Es completamente software libre y tiene versiones para los principales sistemas operativos como: Windows, MacOS y LINUX que es el sistema operativo donde se va a implementar.

2.9 Lenguajes de Programación

Teniendo en cuenta que la plataforma GeneSIG posee una arquitectura cliente-servidor, se hace necesario el uso de diferentes lenguajes de programación para el desarrollo del sistema. Por lo que se requerirá un lenguaje que gestione las peticiones realizadas al servidor, la lógica del negocio y la comunicación con la base de datos; y otro lenguaje que gestione la interfaz de usuario y el modo en que se presenta la información al usuario final.

2.9.1 Lenguajes de Programación del lado del Cliente

Estos son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento. Muestran la parte visual del sistema y permiten una interfaz amigable e intuitiva al usuario.

2.9.1.1 JavaScript

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (Torre, 2006).

Para el desarrollo del sistema propuesto se hace uso de este lenguaje de programación, ya que el mismo fue utilizado para la construcción de la interfaz gráfica de usuario en la plataforma GeneSIG.

2.9.2 Lenguajes de Programación del lado del Servidor

Son lenguajes reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por lo que se requerirá de un lenguaje que sea capaz de lograr equilibrio, eficiencia y rapidez, dando respuesta a todas las solicitudes que sean pedidas a través del cliente.

2.9.2.1 PHP

PHP 5.01 es un lenguaje de programación del lado del servidor, orientado a objetos de alto nivel. Puede ser interpretado por la mayoría de los servidores WEB y en casi todos los sistemas operativos. Es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es uno de los lenguajes más usados para la programación de aplicaciones del lado del servidor (Torre, 2006).

Para el desarrollo del plugin tematización se utiliza este lenguaje de programación debido a que este fue utilizado en el desarrollo del lado del servidor y es el lenguaje servidor utilizado en la implementación de la plataforma GeneSIG.

2.10 Librería de interfaz de usuario

ExtJS 3.0 es una librería JavaScript para el desarrollo de aplicaciones web interactivas, permitiendo recrear nuestro sitio con ventanas y menús similares a una aplicación de escritorio.

Esta librería incluye:

- Componentes UI⁹ del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API¹⁰ fácil de usar.
- Licencias Open Source y comerciales.

Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing. Gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno. Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros (Corzo).

⁹ User Interface

¹⁰ Application Programming Interface

2.11 Técnica de desarrollo web

AJAX es la unión de varias tecnologías que posibilitan hacer páginas Web más interactivas. La característica fundamental de AJAX es que permite actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De manera similar podemos enviar información al servidor. Esto se logra cargando el motor de AJAX cuando se carga el sitio, de modo que a partir de ese momento, él será el encargado de gestionar el intercambio de información entre el cliente y el servidor, refrescando en la página solo lo necesario.

2.12 Conclusiones parciales

Para darle solución al problema, el sistema de información geográfico se desarrolló sobre un ambiente WEB, aplicando tecnologías que permitan una correcta comunicación entre el cliente y el servidor. Se usó PHP 5.01 como lenguaje del lado del servidor junto con el framework CartoWeb 3.0 para facilitar el manejo de mapas, mientras que del lado del cliente se usó de ExtJS 3.0 y AJAX para lograr una correcta comunicación con el servidor.

El sistema en su conjunto estará ejecutándose a través del servidor de mapas MapServer 5.4.1. Como sistema gestor de base de datos se usó PostgreSQL 8.4 con el módulo PostGIS 1.5.1 para la gestión de los datos espaciales. Como metodología que guió todo el proceso de desarrollo se escogió RUP y como lenguaje que permita el modelado se eligió UML 6.4.

CAPÍTULO 3: Presentación de la solución propuesta

En el presente capítulo teniendo en cuenta que no se tienen bien definidos los procesos del negocio se realizará el modelo de dominio y se procederá a explicar cada uno de los conceptos que forman parte del mismo. Todo ello para tener una mejor comprensión de la estructura y dinámica de la organización. Se presenta la solución desde la perspectiva del sistema, se definen sus actores, se exhibe el Diagrama de Casos de Uso del Sistema (DCUS) y luego sus descripciones textuales.

3.1 Entorno donde trabajará el sistema

El Modelo de Dominio permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema. Representa las clases conceptuales del mundo real, no de componentes de software. Puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las oportunidades de los diagramas UML para representar conceptos, el Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión.

3.1.1 Eventos principales del entorno

Los mapas están compuestos por varias escalas representativas, leyendas que permiten un mejor entendimiento de los mismos y las capas que son aquellos mapas que irán conformando el mapa topográfico del sistema. También estos mapas contienen toda la información socioeconómica, tratada en los distintos tipos de datos referentes a todos los sectores de la economía y la sociedad. Los mapas a utilizar son proporcionados por el Cartógrafo el cual pertenece a GeoCuba, siendo la UJC la que proporcione toda la información socioeconómica que se le agregará posteriormente a dichos mapas en dependencia de las necesidades de sus trabajadores.

3.1.2 Diagrama de clases del modelo de dominio

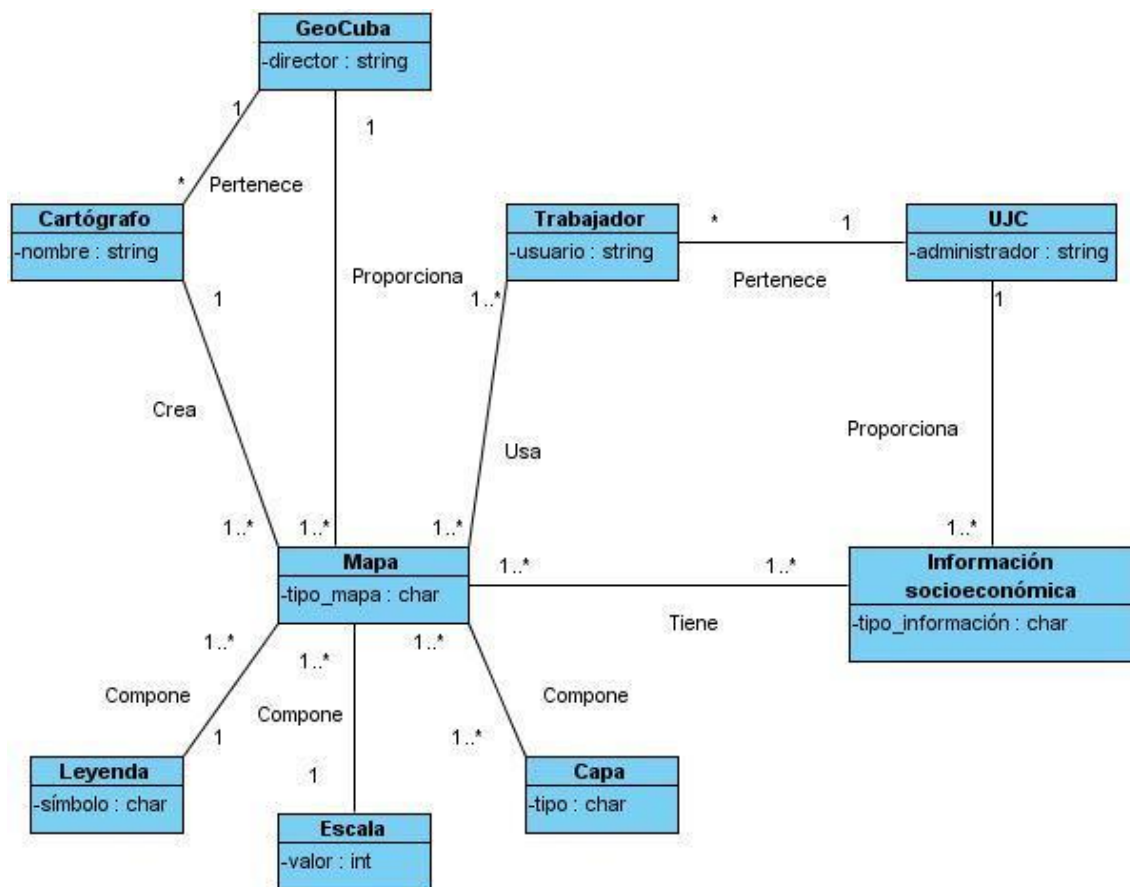


Figura 4: Diagrama de Clases del Modelo de Dominio

3.1.3 Glosarios de términos del dominio

Para una mejor comprensión del Modelo de Dominio a continuación se listan algunos de los términos utilizados en el dominio o específicos del entorno de trabajo.

- **Trabajador**: Persona que trabaja en la entidad y necesita consultar algún tipo de información incluida en un mapa.
- **UJC**: Entidad que solicita el servicio y que proporciona la información socioeconómica referente al mismo.
- **Cartógrafo**: Persona asignada para la realización de cartas geográficas, el estudio y elaboración de mapas.
- **GeoCuba**: Empresa a la que pertenece el Cartógrafo, la cual se dedica a los trabajos de cartografía digital, la aplicación de sistemas de información geográfica, servicios de geodesia y topografía, estudios y soluciones medioambientales, labores de artes gráficas, así como a la calibración de equipos geodésicos (Zerquera, 2009).
- **Información socioeconómica**: Datos relacionados a la economía y la sociedad conjuntamente.

- Mapa: Representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional.
- Leyenda: Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso.
- Escala: Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre.
- Zoom: Es un término inglés que se refiere al teleobjetivo especial de distancia focal variable, es decir, que permite acercar o alejar la imagen según su avance o retroceso (Colectivo de autores).

3.2 Requisitos funcionales

Los requisitos funcionales no son más que las condiciones o capacidades que el sistema debe cumplir. A continuación se enumeran los requisitos funcionales con los que el sistema debe cumplir.

RF1 Autenticar: Verifica que el usuario que va a entrar en el sistema tenga los permisos necesarios para acceder al mismo.

RF2 Realizar Navegación: Modifica la visualización inicial del mapa en la pantalla.

RF2.1 Acercar

RF2.2 Alejar

RF2.3 Centrar

RF2.4 Ver todo

RF2.5 Anterior

RF2.6 Siguiente

RF2.7 Mover

RF3 Realizar consulta geoespacial: Consulta un objeto o región del mapa.

RF4 Crear mapa temático: Visualiza un mapa temático de los locales pertenecientes a ministerios e instituciones.

RF5 Exportar mapa: Exporta un mapa o vista de éste a un fichero en formato pdf. Incluye la configuración de la página y demás elementos para garantizar la posterior impresión.

RF6 Listar entidades: Ubica, modifica o elimina la georreferencia de las entidades sobre el mapa.

RF7 Localizar: Localiza en el mapa una entidad determinada.

3.3 Requisitos no funcionales

No son más que las propiedades y características que el sistema debe tener. A continuación se mencionan los mismos.

Usabilidad

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.
- El sistema tendrá una correcta Arquitectura de la Información, a partir de un estudio de usuarios para su etiquetado e indexado.
- Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

Fiabilidad

- La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.
- La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas.
- La información y las funcionalidades del sistema estarán disponibles y el usuario podrá acceder a ellas las 24 horas de los 7 días de la semana.
- Los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

Eficiencia

- El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

Apariencia o interfaz externa

- El sistema debe tener una apariencia profesional y un diseño gráfico sencillo.
- Posibilitarle al usuario la configuración del entorno de trabajo.

- Ser intuitivo.

Interfaces de hardware

Para las PCs clientes:

- Se requiere tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los Servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas tenga como mínimo 2GB de RAM y 2GB de disco duro.
- El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

Interfaces de software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software:

Para las PCs clientes:

- Un Navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows y Mac OS.

Para los Servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

Interfaces de comunicación

El sistema garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, así como elementos para cambiar las vistas, las escalas y las capas que serán visibles en la interacción.

3.4 Descripción del sistema propuesto

El Modelo de Casos de Uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen distintos tipos de usuarios. Cada usuario se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso.

3.4.1 Descripción de los actores

Un actor es un agente externo que interactúa con el sistema en pos de obtener un resultado esperado. El sistema cuenta con los actores que se especifican a continuación:

Actores	Descripción
Usuario	Trabajador de la UJC que utiliza las funcionalidades del sistema.
Administrador	Personal autorizado, encargado de mantener actualizada la información manejada por el sistema.

3.4.2 Diagrama de casos de uso del sistema

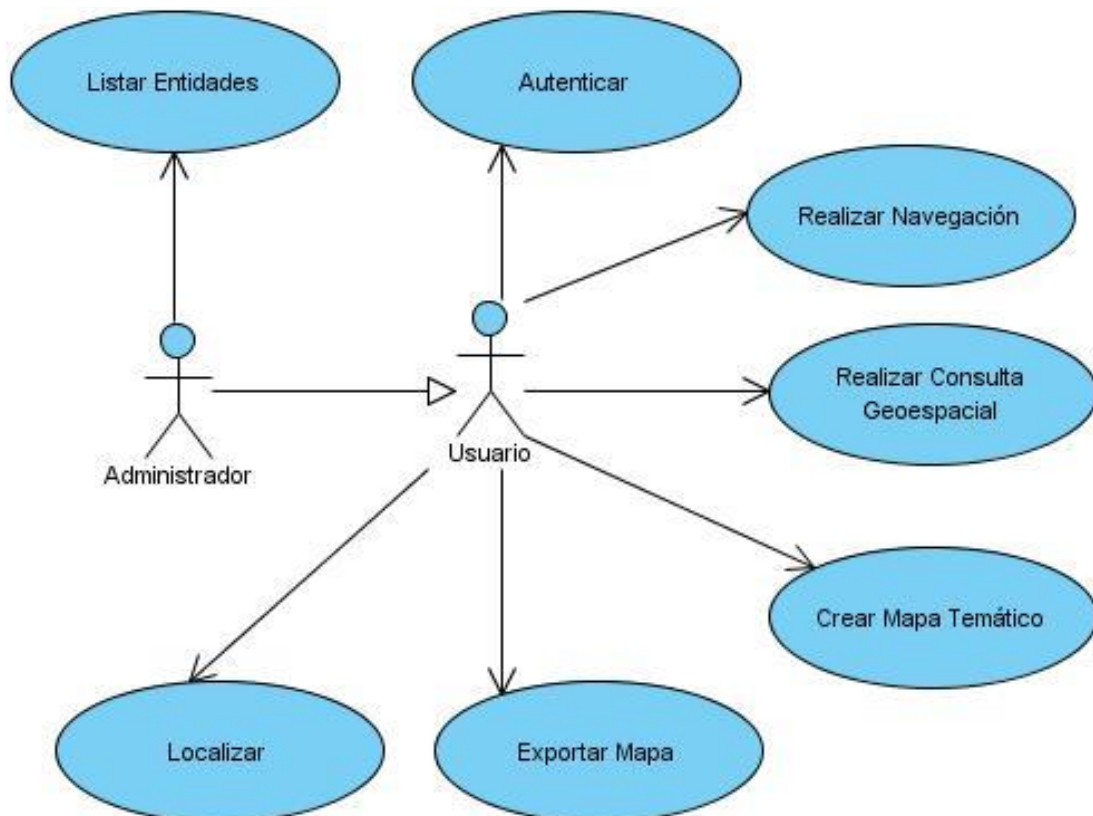


Figura 5: Diagrama de Casos de Uso del Sistema

3.4.3 Descripción textual de los casos de uso del sistema

La descripción textual de los casos de uso del sistema ayudará a un mayor entendimiento de la secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. A continuación se presenta la descripción del caso de uso Crear Mapa Temático, el resto de las descripciones se puede encontrar en los anexos #3.

3.4.3.1 Crear mapa temático

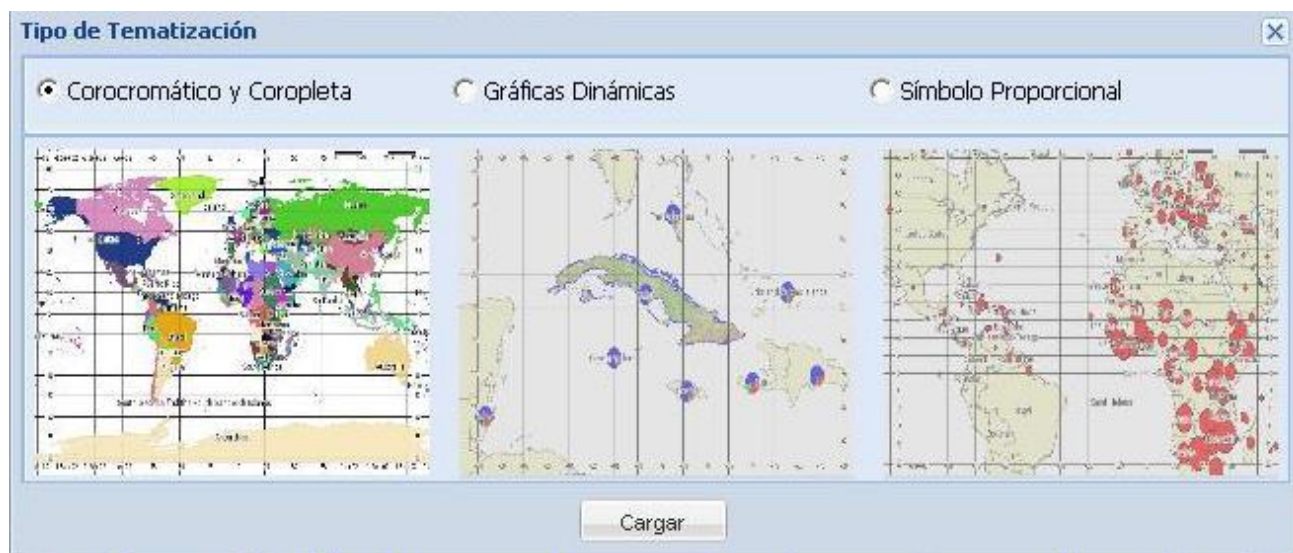
Caso de Uso:	Crear Mapa Temático.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el propósito de poder visualizar un mapa temático de los locales pertenecientes a ministerios e instituciones.
Resumen:	El caso de uso se inicia cuando el usuario desea realizar una tematización de los locales entrando los valores correspondientes a los criterios de análisis, y termina visualizándose el mapa temático.
Precondiciones:	
Referencias	RF 4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario selecciona la opción de Tematizar.(ver Interfaz 1)	<p>2. El sistema muestra la ventana “Tipo de tematización”, donde muestra tres opciones.</p> <ul style="list-style-type: none"> - Si seleccionó “Corocromático y Coropleta”, ver sección “Corocromático y Coropleta”. - Si seleccionó “Gráficas Dinámicas”, ver sección “Gráficas Dinámicas”. - Si seleccionó “Símbolo Proporcional”, ver sección “Símbolo Proporcional”. (ver Interfaz 2)
	3. Finalizando el caso de uso el sistema muestra el tipo de tematización según la selección del usuario.

Prototipo de Interfaz

Interfaz 1



Interfaz 2



Sección "Corocromático y Coropleta"

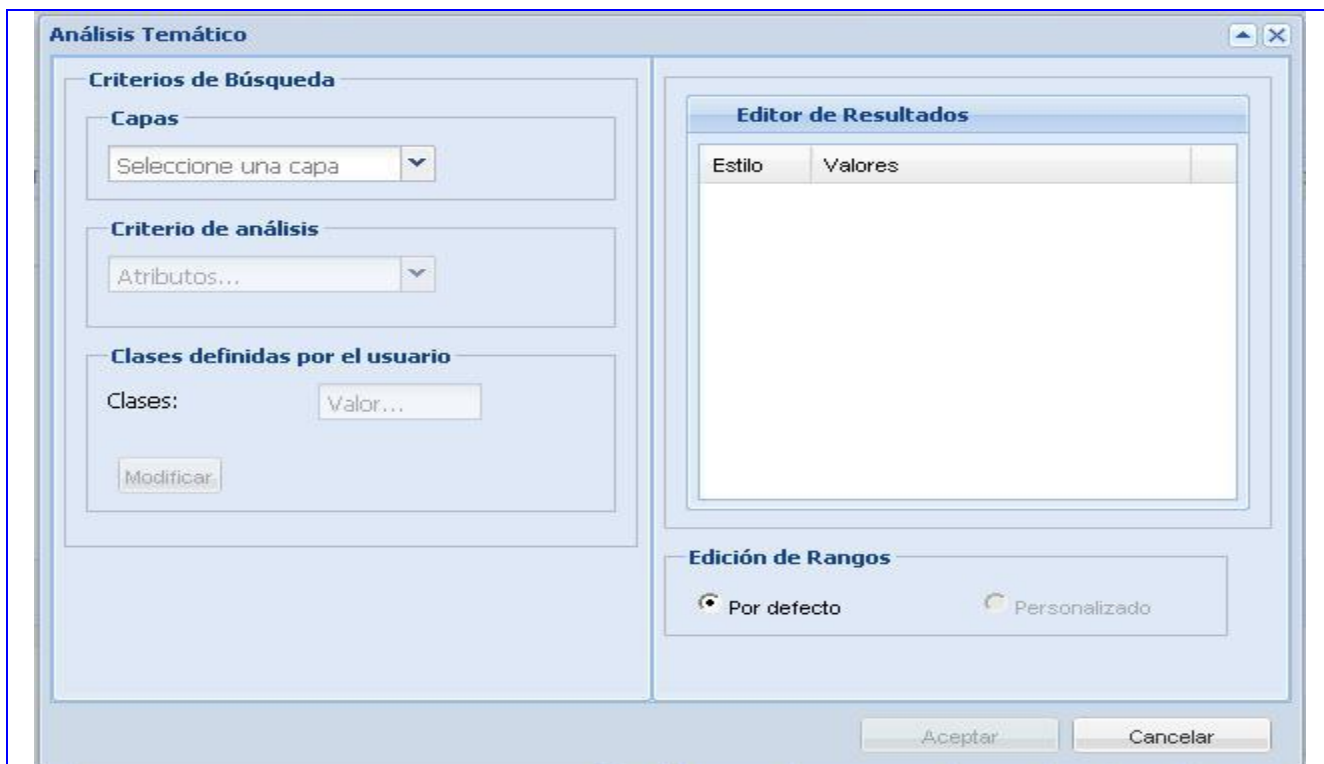
Acción del Actor

1. El usuario selecciona la opción Corocromático y Coropleta, y hace clic en el botón "Cargar".

Respuesta del Sistema

2.2. El sistema muestra la ventana "Análisis Temático" que posee un comboBox "Criterio de análisis" que le permite al usuario especificar el criterio que desea analizar. Posee además un "Editor de Resultados" donde se muestran los estilos y valores para realizar la tematización. (ver interfaz 3)

<p>3. El usuario especifica el criterio de análisis.</p>	<p>4. El sistema muestra el comboBox "Operadores" con los valores:</p> <p>"Mayor que "</p> <p>"Menor que"</p> <p>"Todos "</p>
<p>5. El usuario selecciona el operador "Mayor que" o "Menor que".</p>	<p>6. El sistema muestra el textbox "valor" para insertar el valor numérico a tener en cuenta en el criterio de análisis. Habilita dos radiobutton que permiten seleccionar si la edición de rangos será por defecto o personalizado.</p>
<p>5.1 El usuario selecciona la opción "Todos".</p>	<p>6.1 El sistema selecciona todos los valores existentes a tener en cuenta en el criterio de análisis. Habilita dos radiobutton que permiten seleccionar si la edición de rangos será por defecto o personalizado.</p>
<p>7. El usuario introduce los valores requeridos y hace clic en el botón "Generar".</p>	<p>8. Se muestran los resultados de la búsqueda en el "Editor de Resultados" en tres columnas: Estilo, Cuota Inferior y Cuota Superior.</p>
<p>9. El usuario presiona el botón "Aceptar".</p>	<p>10. El sistema muestra la tematización especificada.</p>
<p>Prototipo de Interfaz</p>	
<p>Interfaz 3</p>	



Flujos Alternos

Acción del Actor	Respuesta del Sistema
7.1 El usuario no introduce los valores requeridos.	8.1 El sistema señala el campo requerido en color rojo para que el usuario introduzca un valor.
7.2 El usuario desea variar la cantidad de intervalos a mostrar en el "Editor de Resultados" e introduce un valor en el campo "Clases" y presiona el botón "Modificar".	8.2 El sistema muestra en el "Editor de Resultados" la cantidad de intervalos establecidos por el usuario.

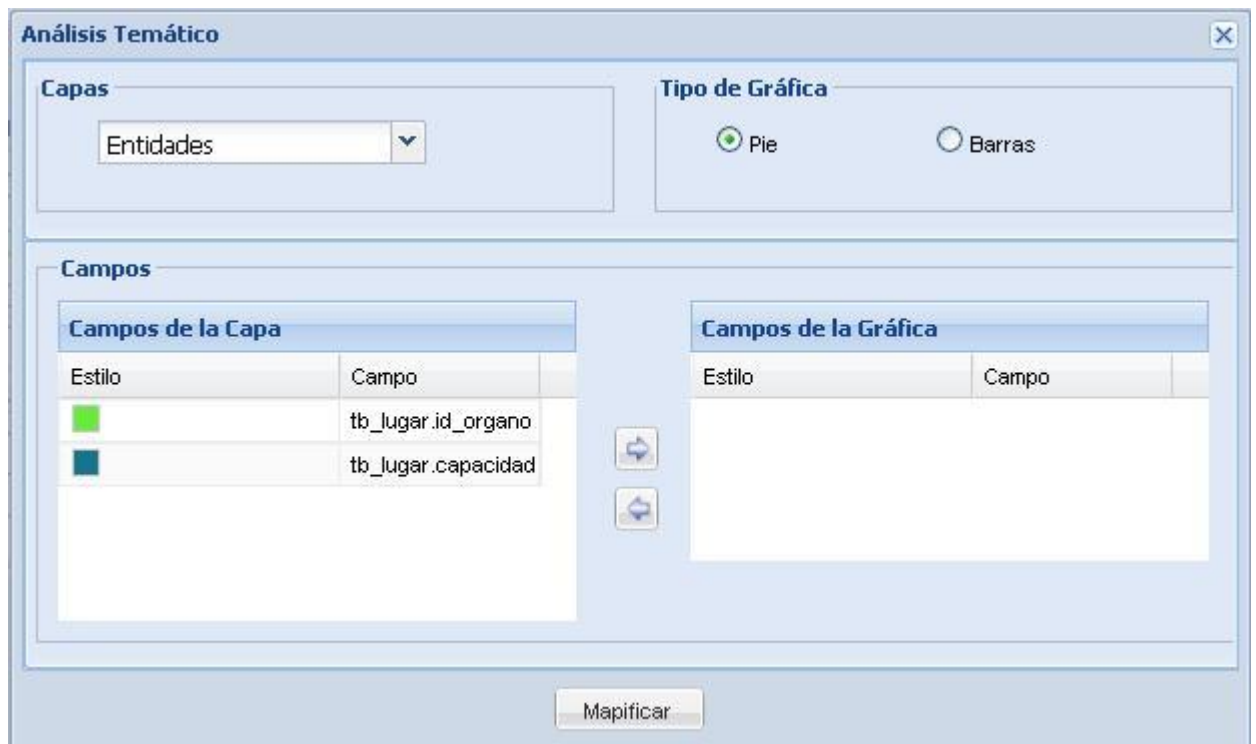
Sección "Gráficas Dinámicas"

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción "Gráfica Dinámica", y hace clic en el botón "Cargar".	2. El sistema muestra la ventana "Análisis Temático" que posee dos radiobutton que permiten seleccionar el tipo de gráfica: Pastel o Barra. Muestra además el DataGrid "Campos de la Capa" el cual muestra todos los campos que posee la capa, y el DataGrid "Campos de la

	gráfica” para que el usuario indique los campos que desea graficar. (ver interfaz 4)
3. El usuario selecciona la capa y realiza un clic sobre uno de los campos indicados en “Campos de la capa” y lo pasa hacia “Campos de la gráfica”. Esta acción puede repetirla mientras queden capas disponibles.	4. El sistema comprueba que el usuario haya indicado los campos que desea graficar
5. El usuario presiona el botón “Mapificar”.	6. Se obtiene la tematización con las características especificadas.

Prototipo de Interfaz

Interfaz 4



Flujos Alternos

Acción del Actor	Respuesta del Sistema
3.1 El usuario selecciona menos de dos capas.	4.1 El sistema muestra un aviso indicando que no ha seleccionado campos suficientes para graficar.

Sección “Símbolo Proporcional”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Símbolo Proporcional”, y hace clic en el botón “Cargar”.	2. El sistema muestra la ventana “Análisis Temático” que permite seleccionar los criterios de análisis mediante un comboBox “Atributos” y uno “Color”. Permite especificar el tipo de símbolo a utilizar. Círculo, Cuadrado, Triángulo y Compuesto. (ver interfaz 5)
3. El usuario selecciona el atributo que desee analizar.	4. El sistema habilita el botón “Mapificar”
5. El usuario selecciona el color de representación y presiona el botón “Mapificar”.	6. El sistema muestra la tematización especificada.

Prototipo de Interfaz

Interfaz 5



Flujos Alternos

Acción del Actor	Respuesta del Sistema
5.1 El usuario no especificó el color de representación. Presiona el botón “Mapificar”.	6.1 El sistema señala que el valor es requerido y no se produce la tematización.
Poscondiciones	Se crea un mapa temático a partir de los valores especificados por el usuario.

3.5 Conclusiones parciales

En el presente capítulo se presentó una descripción del modelo de dominio, detallando el entorno que lo conforma para comprender el negocio al cual el sistema va a servir. Se trazaron los requisitos funcionales entre los cuales se encuentran autenticar, realizar navegación, realizar consulta geoespacial, crear mapa temático, exportar mapa y listar entidades definiendo así la capacidad de acción del sistema. Se identificaron y describieron los actores y casos de usos del sistema, exponiéndose las relaciones entre cada uno de ellos en el diagrama de casos de uso del sistema para ilustrar la comunicación y el comportamiento del sistema mediante su interacción con los usuarios. De esta forma queda presentada la solución de la propuesta y se procede a la fase de construcción de la misma.

CAPÍTULO 4: Construcción de la solución propuesta

Realizar un diseño para una aplicación es importante, pues, con ello se puede desarrollar la programación del sistema de forma más eficiente, permitiendo un acercamiento al lenguaje de programación en el cual será desarrollada. En el presente capítulo se modela el diseño e implementación del sistema propuesto, con todos sus artefactos y diagramas basándose en distintos patrones de diseño y arquitectura.

4.1 Arquitectura Propuesta

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura lógica. Se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiar en el desarrollo de software dentro de un sistema informático.

Así, los programadores, diseñadores, ingenieros y analistas pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado (Guglielmetti, 2005).

Patrón de arquitectura orientada a objetos

Resumiendo las características de las arquitecturas Orientadas a Objetos (OO), los componentes del estilo se basan en principios OO: encapsulamiento, herencia y polimorfismo. Son las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. En cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases.

El sistema será modelado haciendo uso de este patrón arquitectónico, ya que el framework CartoWeb, que será utilizado para su desarrollo, posee una arquitectura orientada a objetos.

Patrón de arquitectura basada en componentes

Las características principales de este patrón son la modularidad, la reusabilidad y compatibilidad. En la arquitectura basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento.

El sistema será diseñado sobre la arquitectura basada en componentes, pues el framework CartoWeb también cumple con dicho patrón arquitectónico. La estructura del framework CartoWeb está compuesta por plugins lo cual permite que la arquitectura del sistema sea flexible y fácil de personalizar.

4.2 Modelo de diseño

El modelo de diseño está muy cercano al de implementación, en él se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se suponen.

El diseño de CartoWeb está conformado por paquetes, donde cada uno de ellos realiza una función determinante y la relación interactiva entre los mismos conlleva al funcionamiento óptimo del framework. Entre los paquetes que contiene CartoWeb se encuentra uno llamado "Project", donde se encuentran guardadas todas las aplicaciones que se construyan con esta herramienta, y es precisamente donde se encontrará ubicado el sistema.

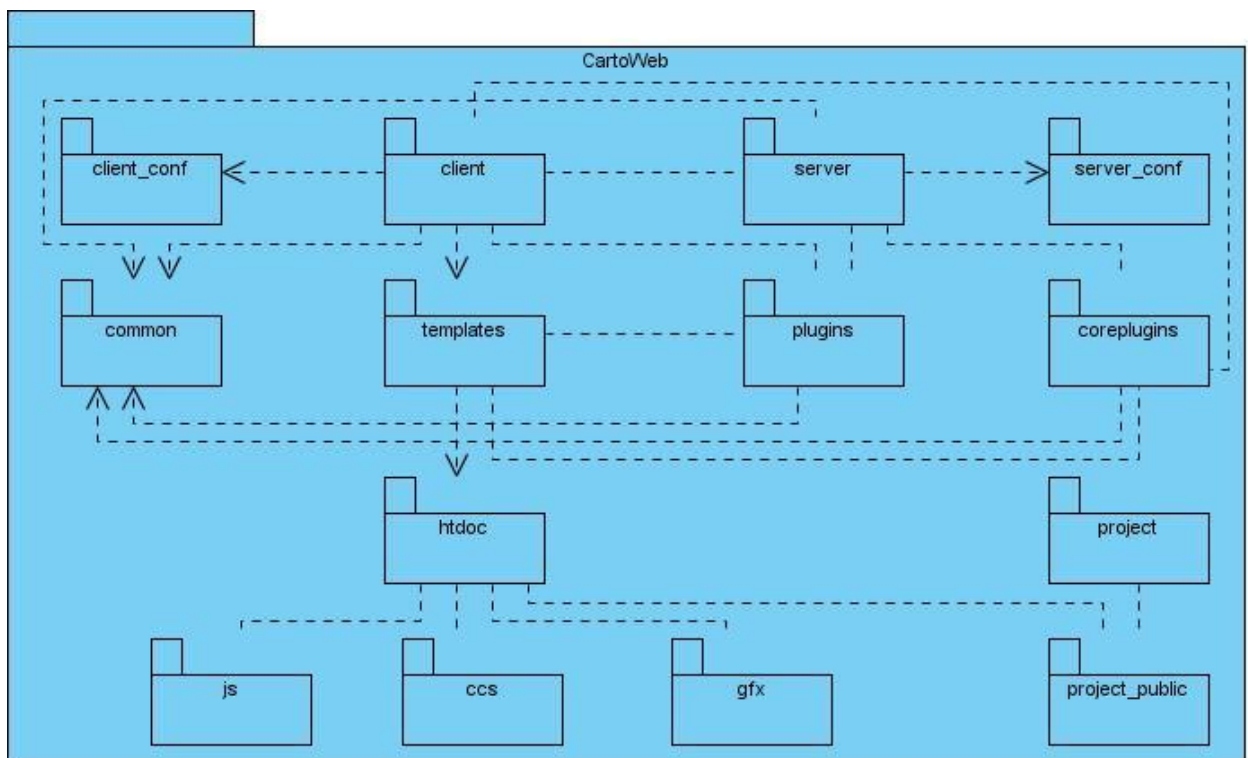


Figura 6: Estructura de CartoWeb.

Módulos	Descripción
Client	Contiene las clases clientes que utiliza CartoWeb.
Server	Contiene las clases servidoras que utiliza CartoWeb.
Coreplugins	Contiene los plugins básicos del CartoWeb.
Project	Contiene los proyectos publicados.
Common	Contiene las clases que son comunes para el funcionamiento del CartoWeb. Incluye las clases que servirán de puente para la comunicación entre cliente-servidor.
Plugins	Contiene los plugins de CartoWeb que pueden ser modificados.
Templates	Contiene las plantillas que utiliza CartoWeb, construidas en el generador de plantillas Smarty para el procesamiento de datos.
server_conf	Contiene las configuraciones de las clases servidoras.
client_conf	Contiene las configuraciones de las clases clientes.
Htdocs	Contiene los ficheros htdocs que utiliza el CartoWeb.
Js	Contiene los ficheros js de programación de la interfaz que utiliza CartoWeb.
Css	Contiene los estilos que utiliza CartoWeb.
Gfx	Contiene las imágenes que utiliza CartoWeb.
project_public	Contiene los vínculos de los proyectos publicados.

Tabla 1: Módulos de CartoWeb

4.2.1 Patrones de diseño

A lo largo de los años y a partir de experiencias adquiridas durante el desarrollo de sistemas informáticos, se han ido documentando una serie de patrones y buenas prácticas de programación. Con el objetivo de construir un software basándose en la experiencia colectiva de ingenieros de software.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan (Gracia, 2005).

Observer (Observador): Se aplica en la funcionalidad Listar Entidades; la filosofía de este patrón es que aparezcan en tiempo real los cambios efectuados en el mapa por un usuario. Estos cambios son observados por el alto mando, que son los encargados de la toma de decisiones. Este fenómeno trabaja internamente en la aplicación sin hacer uso del patrón ya que el sistema pide continuamente al cliente donde se está editando la cartografía digital y los cambios para mostrarlos, este mecanismo es

ineficiente debido a que tiene que refrescar continuamente en busca de cambios, esto trae consigo una sobrecarga en el sistema.

Para enmendar este problema presentado se aplica la solución brindada por el patrón seleccionado. La cual sería utilizar un Observador, que no es más que una clase con un atributo “actualizar” el cual es el encargado de notificar al sistema que ha ocurrido algún cambio y este lo representará al puesto de mando según el cambio notificado.

Singleton (Solitario): En el diseño de clases es necesario aplicar la solución del patrón Singleton que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. Se utiliza para modificar el framework CartoWeb, el objetivo del mismo es crear el objeto “mapa” para que no se cree cada vez que se hace una petición en la aplicación.

Command (Acción): Se utiliza en el proceso de petición mediante la Interfaz Gráfica de Usuario (GUI) al sistema de una información cualquiera por un cliente. Uno de los aspectos más importantes del sistema son las GUI, ya que el usuario interactúa constantemente con ellas y por eso se aplica la solución propuesta por este patrón de diseño.

4.2.2 Diagrama de clases de diseño

El diagrama de clases del diseño del caso de uso Crear mapa temático presentado a continuación está basado en la implementación que se realiza con el framework CartoWeb, además de estar diseñado bajo una arquitectura de componentes, el resto de los diagramas se puede encontrar en el anexo #1.

Caso de Uso: Crear mapa temático

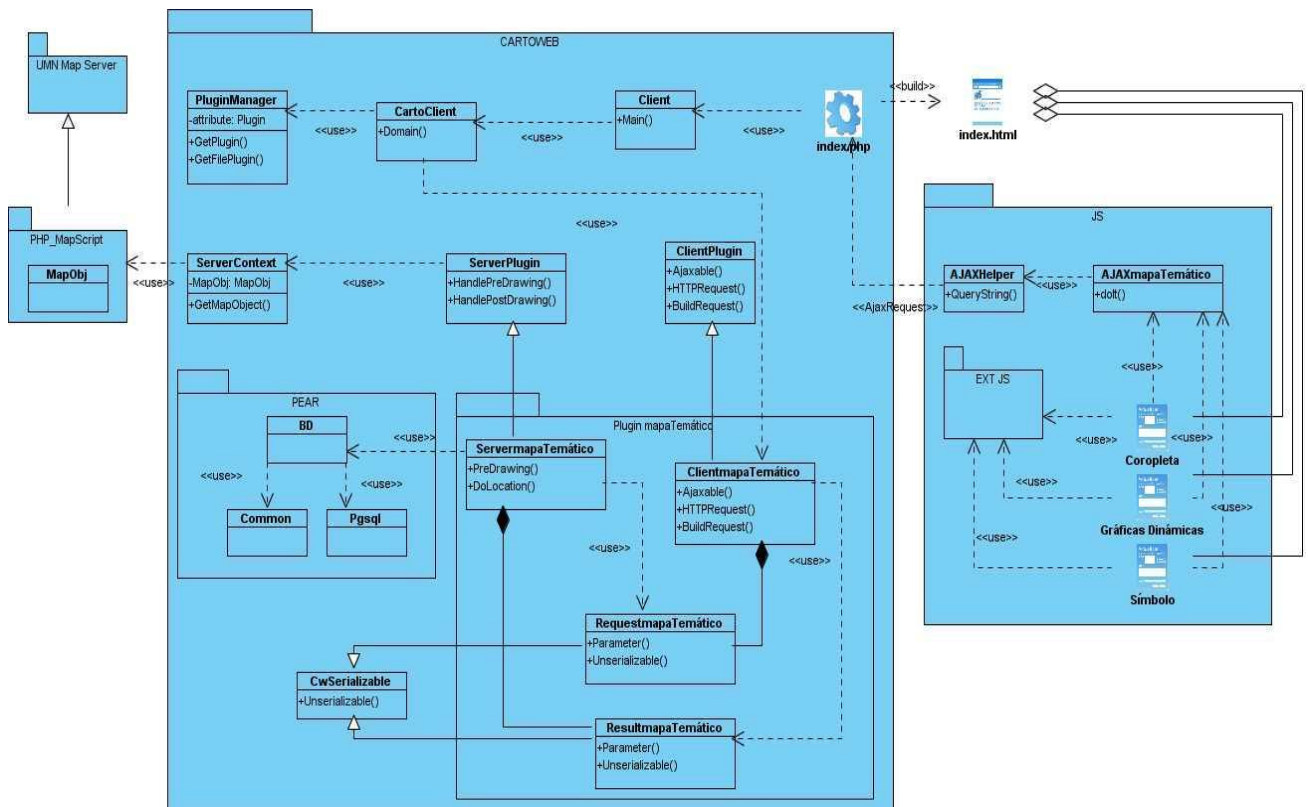


Figura 7: Diagrama de clases del diseño del CU Crear mapa temático

4.3 Principios de diseño

Para lograr una interfaz de usuario amigable y funcional para el usuario final, es necesario definir principios de diseño de interfaz de usuario, pues esta es una actividad de gran importancia dentro del diseño de un software. Se debe tener en cuenta que de la aceptación que tengan los usuarios depende, en gran medida el éxito del sistema.

4.3.1 Estándares de la interfaz de la aplicación

A continuación se definen los principios a tener en cuenta en el desarrollo de la interfaz de usuario de la aplicación:

- Mostrar al usuario toda la información y herramientas necesarias para cada etapa en su trabajo.
- Brindar una interfaz sencilla, de manera tal que cualquier persona con un mínimo dominio de la computación pueda trabajar con la aplicación.
- Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.

- Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.
- Los mensajes mostrados al usuario deben ser concisos y de fácil comprensión.
- Los menús y etiquetas de botones deben comenzar con la palabra más importante.
- Los eventos más importantes del sistema deben ser mostrados en una barra de estado.

4.4 Diseño de la Base de Datos

Durante el diseño de la base de datos se identifican todas las clases persistentes, específicamente en el diagrama de clases persistentes; este diagrama muestra la estructura lógica de la base de datos mediante clases. A continuación se muestra el diagrama de clases persistentes referentes al sistema y posteriormente se representa el diagrama entidad relación de la base de datos que se genera del diagrama anterior.

4.4.1 Diagrama de Clases Persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Por lo general las clases persistentes tienen como origen las clases identificadas como entidad porque ellas modelan la información y el comportamiento asociado de algún fenómeno o concepto. Las clases persistentes referencian directamente las entidades lógicas y sus atributos. A continuación se muestra el diagrama de clases persistentes del sistema.

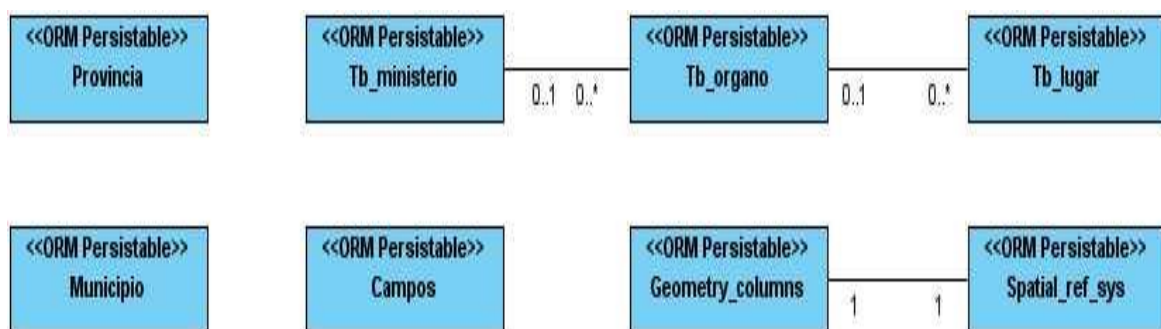


Figura 8: Diagrama de Clases Persistentes

4.4.2 Diagrama Entidad-Relación

Denominado por sus siglas como E-R este modelo representa la realidad a través de un esquema gráfico empleando la terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares

denominadas atributos. El enlace que rige la unión de las entidades está representado por la relación del modelo (Sofia, 2002).

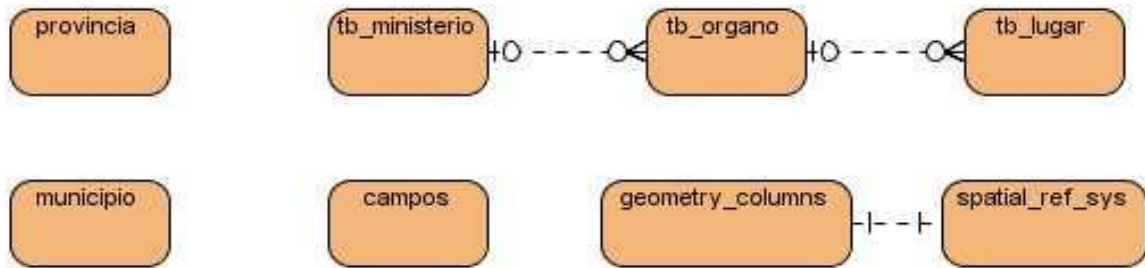


Figura 9: Diagrama Entidad-Relación

4.5 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. Es un conjunto de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos y procesos.

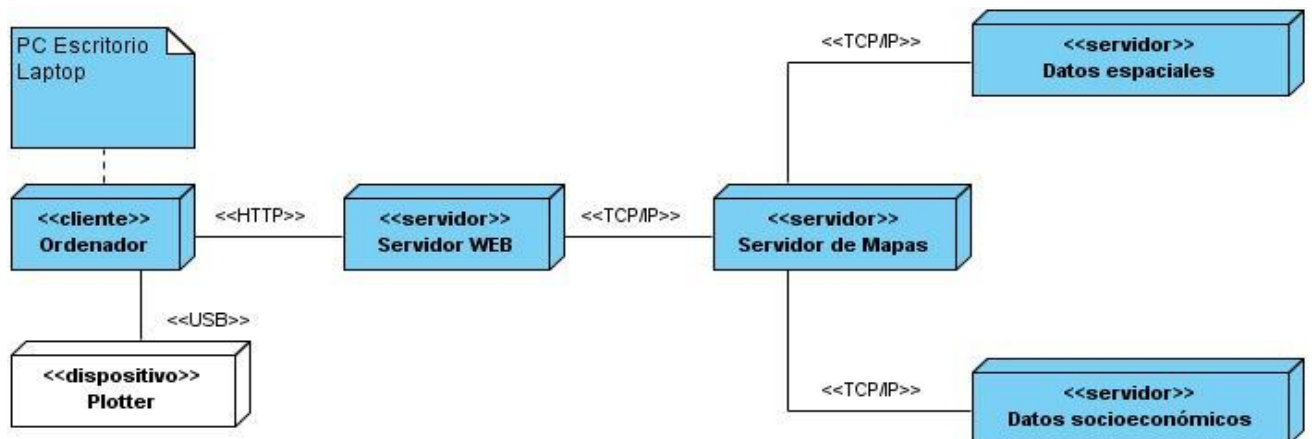


Figura 10: Diagrama de Despliegue.

4.6 Modelo de implementación

El Modelo de Implementación constituye una colección de componentes y los subsistemas o paquetes que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y otros tipos de ficheros necesarios para la implantación y el despliegue del sistema.

4.6.1 Diagrama de Componentes

El Diagrama de Componentes muestra un conjunto de componentes y sus relaciones. Modela los aspectos físicos del sistema, la vista de implementación estática del sistema y los elementos físicos que residen en un nodo, tales como ejecutables, tablas, librerías, archivos y documentos. A continuación se muestra el diagrama de componentes del caso de uso Crear mapa temático, los demás diagramas de componentes se pueden encontrar en el anexo #2.

Diagrama de componentes del CU: Crear mapa temático

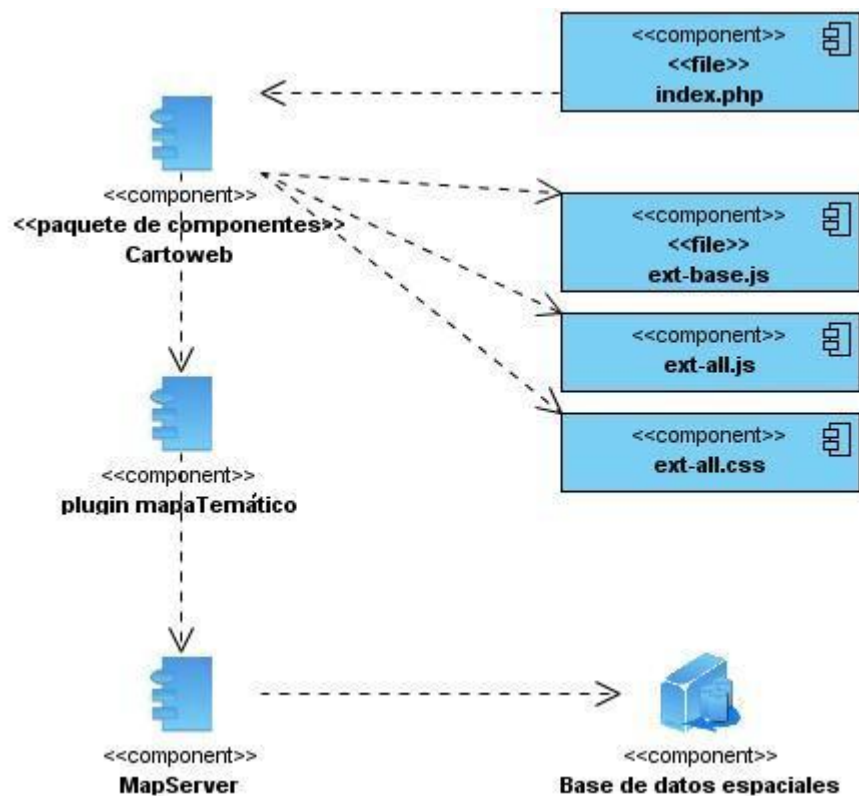


Figura 11: Diagrama de Componentes del CU Crear mapa temático.

4.7 Conclusiones parciales

Se concluye que para el desarrollo del sistema se utilizaron los patrones de arquitectura orientada a objetos y basada en componentes, los cuales se basan fundamentalmente en la estructura del framework CartoWeb, posibilitando la flexibilidad y la fácil personalización de la arquitectura del sistema. Se aplicaron los patrones de diseño Observador, Singleton y Command para ayudar en la solución de problemas particulares del sistema, utilizando un pequeño número de clases relacionadas de forma determinada.

Entre los principios a tener en cuenta en el desarrollo de la interfaz de usuario de la aplicación se encuentran brindar una interfaz sencilla y garantizar la legibilidad, de manera que exista contraste de colores de los textos con el fondo. Se realizó el modelo de despliegue a través del cual se muestra la distribución física del sistema y sus conexiones. Finalmente se expuso el modelo de implementación, donde se especificaron los distintos componentes creados para desarrollar la aplicación y su relación entre ellos, lográndose traducir el diseño en componentes ejecutables.

Conclusiones

Con el desarrollo de este sistema de información geográfica para la UJC Nacional, queda demostrado el cumplimiento los objetivos que dieron inicio al desarrollo de esta investigación y se arriban a las siguientes conclusiones:

- A partir del estudio realizado sobre el proceso de desarrollo de un plugin para CartoWeb, framework que se utilizó en el desarrollo de la plataforma GeneSIG, se logró el desarrollo del plugin tematización.
- Con el análisis de los requisitos funcionales solicitados por el cliente se logró la personalización de la Plataforma GeneSIG para la Oficina Nacional de la UJC.
- La oficina nacional de la UJC cuenta con un SIG mediante el cual se soluciona la imposibilidad de georreferenciar los distintos locales de los organismos e instituciones del país, destinados para la recreación.
- El SIG desarrollado incluye funcionalidades que permiten el análisis y representación de la información socioeconómica referente a los objetos georreferenciados. Facilitando a la oficina nacional de la UJC, realizar la recolección de estadísticas sobre cuáles son las ofertas con más demanda y mantener actualizada toda la información sobre los distintos locales de los organismos e instituciones del país, destinados para la recreación.
- El sistema esta desarrollado con una arquitectura web, lo cual permite que sea centralizado, evitando la duplicación de la información y haciendo más fácil la actualización y recogida de datos para su posterior análisis.
- Al sistema desarrollado se le realizaron siete pruebas de software, una por cada caso de uso implementado, basadas en las técnicas de caja negra. Los resultados obtenidos reflejan la correcta implementación de las funcionalidades dado que el 100% de las pruebas realizadas arrojaron resultados satisfactorios.

Recomendaciones

En correspondencia con los resultados obtenidos a lo largo del proceso investigativo y basados en la experiencia acumulada en el período se proponen las siguientes recomendaciones:

- Implementar una nueva versión utilizando OpenLayers con el fin de optimizar el rendimiento de la aplicación.

Bibliografía

[En línea] [Citado el: 12 de 11 de 2010.] <http://www.vmapas.com/>.

autores, Colectivo de. 2010. Ciberaula. [En línea] 2010. [Citado el: 20 de 2 de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

1997-2005. Cinfonet. [En línea] 1997-2005. [Citado el: 20 de 2 de 2011.] <http://hospedaje.cinfonet.com/?web=apache>.

2010. Fase_de_Inicio._Disciplina_de_Modelamiento_del_Negocio. 2010.

2010. Sun Microsystems. [En línea] 2010. [Citado el: 19 de 2 de 2011.] http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html.

Cano, José Ignacio Barredo. 1996. Sistemas de Información Geográfica y evolución multicriterio en la ordenación del territorio. 1996.

Colectivo de autores. baldalibre. baldalibre. [En línea] [Citado el: 20 de 1 de 2011.] http://www.trabalu.com/index.php?option=com_content&view=article&id=45&Itemid=55&lang=es&limitstart=8.

Definición.de. [En línea] [Citado el: 20 de 2 de 2011.] <http://definicion.de/zoom/>.

Elementos de UML. [En línea] [Citado el: 15 de 2 de 2011.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>.

Corzo, Giancarlo. Desarrollo en Web. [En línea] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

DAEDALUS. [En línea] [Citado el: 12 de 11 de 2010.] <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ciencia-de-sistemas/que-es-un-sistema/>.

Daniele, Marcela. 2007. [En línea] 2007. [Citado el: 2 de 4 de 2011.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.

2008. Definición de cartografía - Qué es, Significado y Concepto. [En línea] 2008. [Citado el: 12 de 11 de 2010.] <http://definicion.de/cartografia/>.

2008. Definición de mapa - Qué es, Significado y Conceptos. [En línea] 2008. [Citado el: 12 de 11 de 2010.] <http://definicion.de/mapa/>.

El Proceso Unificado de Desarrollo de Software.

Espinosa, Antonio Membrides. 2008. Como desarrollar plugins para CartoWeb. Como desarrollar plugins para CartoWeb. [En línea] 19 de 12 de 2008. [Citado el: 15 de 1 de 2011.]

ESRI. PDF. [En línea] <http://www.geotecnologias.com/Documentos/what-is-arcgis-spanish.pdf>.

ExtJS Framework. [En línea] [Citado el: 12 de 11 de 2010.] <http://groupbit.com/curso-extjs-framework>.

Gary E. Sherman, Tim Sutton, Radim Blazek, Stephan Holl, Otto Dassau, Tyler Mitchell, Brendan Morely, Lars Luthman, Godofredo Contreras, Magnus Homann, Martin Dobias. 2004 - 2007. Quantum GIS. Quantum GIS. [En línea] 2004 - 2007. http://download.osgeo.org/qgis/doc/manual/qgis-0.9.1_user_guide_es.pdf.

Gianfelici, Esteban. 2008. mapas y mapas. mapas y mapas. [En línea] 7 de 2008. [Citado el: 3 de 1 de 2011.] <http://www.mapasy mapas.com.ar/el%20dato%20geografico.php>.

Gracia, Joaquin. 2005. IngenieroSoftware. [En línea] 27 de 5 de 2005. [Citado el: 20 de 3 de 2011.] <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.

Guevara, Roberto Tinoco. 2007. Definición y algunas aplicaciones de los Sistemas de Información Geográfica. Sistemas de Información Geográfica para la Ingeniería Empresarial. [En línea] 2007.

Guglielmetti, Marcos. 2005. Mastermegazin. Mastermegazin. [En línea] 10 de 2 de 2005. [Citado el: 18 de 3 de 2011.] <http://www.mastermegazine.info/termino/3916.php>.

HERNANDEZ, JIMMY. Patrones de Diseño. [En línea] [Citado el: 25 de 2 de 2011.] <http://patronesdediseno.blogspot.com/2009/05/uml.html>.

Ing. Ramón Martínez, Lic. Manuel Vidaurre, Geog. Patricia Nájera, Dr. Enrique Loyola, Dr. Carlos Castillo-Salgado. 2001. SIGEpi: Sistema de Información Geográfica. [En línea] 03 de 09 de 2001. [Citado el: 12 de 11 de 2010.] http://www.paho.org/spanish/sha/be_v22n3-SIGEpi1.htm.

2010. Instituto de Investigación de Recursos Biológicos Alexander von Humboldt . [En línea] 2010. [Citado el: 12 de 11 de 2010.] <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001>.

2010. Instituto Nacional de Estadística y Geografía. [En línea] 2010. [Citado el: 12 de 11 de 2010.] <http://mapserver.inegi.gob.mx/geografia/espanol/prodyserv/marcoteo/mundo/mundo.cfm?c=234>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 1999. El proceso unificado de desarrollo de software. Madrid : s.n., 1999.

José H. Canós, Patricio Letelier y M^a Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. pág. 8.

José Luis Almazán Gárate, M. Carmen Palomino Monzón, Hilda Araceli Márquez Caba. 2009. Sistemas de Información Geográfica en la Gestion Integral del litoral. Universidad Politécnica de Madrid : s.n., 2009.

2006. Lenguajes del lado del cliente o servidor. [En línea] 2006. [Citado el: 12 de 11 de 2010.]

http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Lorenzo, Dra. Carmen Mosquera. LA INFORMACIÓN EN UN SIG.

Marcano. 2005. La Hidrología. La Hidrología. [En línea] 2005. [Citado el: 20 de Noviembre de 2010.] <http://jmarcano.topcities.com/ciencias/hidrologia.html>.

Martín, Manuel Martín. Manual PostGIS.

NCGIA. 1990. S.I.G. S.I.G. [En línea] 1990. [Citado el: 4 de 2 de 2011.] <http://juliangiraldo.wordpress.com/sig/>.

Obando, Carlos. ediciones especiales. ediciones especiales. [En línea] [Citado el: 15 de 1 de 2011.] <http://www.edicionesespeciales.elmercurio.com/destacadas/detalle/index.asp?idnoticia=20110609736631&idcuerpo=966>.

Parker. 1988. EduSIG. EduSIG. [En línea] 1988. [Citado el: 12 de 1 de 2011.] <http://www.edusig.com.ar/contenido/sig/definiciones-sobre-sig>.

Pores, Martín de. 2010. Informática de Hoy. [En línea] 2010. [Citado el: 12 de 11 de 2010.] <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-un-sistema-informatico.php>.

2010. Portal - Ministerio del Poder Popular para la Salud - Venezuela. [En línea] 17 de 06 de 2010. [Citado el: 12 de 11 de 2010.] <http://www.mpps.gob.ve/modules.php?name=Content&pa=showpage&pid=510>.

2004. Portal - Ministerio del Poder Popular para la Salud - Venezuela. [En línea] 17 de 06 de 2004. [Citado el: 12 de 11 de 2010.] <http://www.mpps.gob.ve/modules.php?name=Content&pa=showpage&pid=1>.

Pozo, Pedro. Clikear. [En línea] [Citado el: 15 de 2 de 2011.] <http://www.clikear.com/manuales/uml/diagramascasouso.aspx>.

Rico, Gabriel J.Ortiz. Topografía Global. [En línea] <http://www.topografiaglobal.com.ar/archivos/teoria/sig.html>.

1979-2010. Rister. [En línea] 1979-2010. [Citado el: 19 de 2 de 2011.] http://www.rister.com/es/Glosario_Soluciones_Tecnicas/0/s/.

Sanchez, María A. Mendoza. 2004. Metodologías De Desarrollo De Software. [En línea] 7 de 6 de 2004. [Citado el: 4 de 2 de 2011.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Scheltón Lima, Annabell, Ramírez Leal, Ramdy y López Cardoso, Osmani. Tecnologías y herramientas libres para el desarrollo de SIG en la WEB.

Silva, Dr. José Luis Batista. 2005. Aplicación de sistemas de información geográfica en Cuba. [En línea] 11 de 2005. [Citado el: 12 de 11 de 2010.]
http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051.

Sofia, Albert Anibal Osiris. 2002. Diagrama de Entidad-Relación. Diagrama de Entidad-Relación. [En línea] 2002. [Citado el: 1 de 4 de 2011.]
<http://capacitacion2010.googlecode.com/svn-history/r67/trunk/DERw.pdf>.

STIAN. [En línea] <http://www.stian.com.mx/STIAN/default.aspx?pg=073d881b-2d35-4ecb-b622-567973ce91bd>.

Torre, Aníbal de la. 2006. Lenguajes del lado servidor o cliente. [En línea] 2006. [Citado el: 22 de 1 de 2011.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Valle, José. 2005. monografias.com. [En línea] 2005. [Citado el: 12 de 1 de 2011.]
<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml#queson>.

Vaswani, Vikram. PHP and MySQL.

Wikipedia. 2010. Aplicación web - Wikipedia, la enciclopedia libre. [En línea] 08 de 11 de 2010. [Citado el: 12 de 11 de 2010.]
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web#cite_ref-ayudaappweb_0-0.

2010. MapServer - Wikipedia, la enciclopedia libre. [En línea] 30 de 09 de 2010. [Citado el: 12 de 11 de 2010.] <http://es.wikipedia.org/wiki/MapServer>.

Zerquera, Idalia Vázquez. 2009. Vanguardia. [En línea] 16 de 11 de 2009. [Citado el: 23 de 2 de 2011.]
http://www.vanguardia.co.cu/?tpl=design/secciones/lectura/portada.tpl.html&newsid_obj_id=18716.

Trabajos citados

autores, Colectivo de. 2010. Ciberaula. [En línea] 2010. [Citado el: 20 de 2 de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

1997-2005. Cinfonet. [En línea] 1997-2005. [Citado el: 20 de 2 de 2011.] <http://hospedaje.cinfonet.com/?web=apache>.

2010. Sun Microsystems. [En línea] 2010. [Citado el: 19 de 2 de 2011.] http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html.

Colectivo de autores. baldalibre. baldalibre. [En línea] [Citado el: 20 de 1 de 2011.] http://www.trabalu.com/index.php?option=com_content&view=article&id=45&Itemid=55&lang=es&limitstart=8.

Definición.de. [En línea] [Citado el: 20 de 2 de 2011.] <http://definicion.de/zoom/>.

Elementos de UML. [En línea] [Citado el: 15 de 2 de 2011.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>.

Corzo, Giancarlo. Desarrollo en Web. [En línea] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

Espinosa, Antonio Membrides. 2008. Como desarrollar plugins para CartoWeb. Como desarrollar plugins para CartoWeb. [En línea] 19 de 12 de 2008. [Citado el: 15 de 1 de 2011.]

Gianfelici, Esteban. 2008. mapas y mapas. mapas y mapas. [En línea] 7 de 2008. [Citado el: 3 de 1 de 2011.] <http://www.mapasy mapas.com.ar/el%20dato%20geografico.php>.

Gracia, Joaquin. 2005. IngenieroSoftware. [En línea] 27 de 5 de 2005. [Citado el: 20 de 3 de 2011.] <http://www.ingenierosoftware.com/analisisydiseño/patrones-diseño.php>.

Guglielmetti, Marcos. 2005. Mastermegazin. Mastermegazin. [En línea] 10 de 2 de 2005. [Citado el: 18 de 3 de 2011.] <http://www.mastermegazine.info/termino/3916.php>.

HERNANDEZ, JIMMY. Patrones de Diseño. [En línea] [Citado el: 25 de 2 de 2011.] <http://patronesdediseño.blogspot.com/2009/05/uml.html>.

NCGIA. 1990. S.I.G. S.I.G. [En línea] 1990. [Citado el: 4 de 2 de 2011.] <http://juliangiraldowordpress.com/sig/>.

Obando, Carlos. ediciones especiales. ediciones especiales. [En línea] [Citado el: 15 de 1 de 2011.] <http://www.edicionesespeciales.elmercurio.com/destacadas/detalle/index.asp?idnoticia=20110609736631&idcuerpo=966>.

Parker. 1988. EduSIG. EduSIG. [En línea] 1988. [Citado el: 12 de 1 de 2011.] <http://www.edusig.com.ar/contenido/sig/definiciones-sobre-sig>.

Pozo, Pedro. Clikear. [En línea] [Citado el: 15 de 2 de 2011.]
<http://www.clikear.com/manuales/uml/diagramascasouso.aspx>.

1979-2010. Rister. [En línea] 1979-2010. [Citado el: 19 de 2 de 2011.]
http://www.rister.com/es/Glosario_Soluciones_Tecnicas/0/s/.

Sanchez, María A. Mendoza. 2004. Metodologías De Desarrollo De Software. [En línea] 7 de 6 de 2004. [Citado el: 4 de 2 de 2011.]
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Sofia, Albert Anibal Osiris. 2002. Diagrama de Entidad-Relación. Diagrama de Entidad-Relación. [En línea] 2002. [Citado el: 1 de 4 de 2011.]
<http://capacitacion2010.googlecode.com/svn-history/r67/trunk/DERw.pdf>.

Torre, Aníbal de la. 2006. Lenguajes del lado servidor o cliente. [En línea] 2006. [Citado el: 22 de 1 de 2011.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Valle, José. 2005. monografias.com. [En línea] 2005. [Citado el: 12 de 1 de 2011.]
<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml#queson>.

Zerquera, Idalia Vázquez. 2009. Vanguardia. [En línea] 16 de 11 de 2009. [Citado el: 23 de 2 de 2011.]
http://www.vanguardia.co.cu/?tpl=design/secciones/lectura/portada.tpl.html&newsid_obj_id=18716.

Glosario de Términos

Actor: Es algo con comportamiento, ya sea una persona (identificada por un rol), un sistema informatizado u organización, que realiza algún tipo de interacción con el sistema (Pozo).

Caso de Uso (CU): Es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse (Pozo).

Diagrama de Casos de Uso (DCU): Muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa (Pozo).

Diagrama de Clases (DC): Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras (Colectivo de autores).

Clase: Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objetos tiene el suyo propio) (Colectivo de autores).