

UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS (UCI)

Facultad 10



**Título: MODELACION DEL SUBSISTEMA DE CONTROL DE LA
INFORMACION GEOESPACIAL EN LINEA.**

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor: Gretel Hidalgo Valdés.

Tutor: Ing. Wilber Linares Frómeta.

Co-tutor: Ing. Joaquín Quintas Santiago.

Ciudad de La Habana, Cuba

Junio del 2007

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Gretel Hidalgo Valdés

Ing. Wilber Linares Frómata

Ing. Joaquín Quintas Santiago



La confianza en sí mismo es el primer secreto del éxito.

Ralph Waldo Emerson.

AGRADECIMIENTOS

A todos los que me brindaron su apoyo incondicional,

A los que siempre estuvieron junto a mí sin importar la distancia,

A todos los que me ayudaron,

A los que me alentaron,

A los que creyeron en mí,

Sin ellos hubiera sido realmente difícil llegar hasta aquí.

[A mi tutor][A mis amigos][A mis padres][A mi hermana][Mis suegros][Mi novio]

DEDICATORIA

A mis padres...

RESUMEN

Actualmente en la entidad existen una serie de aplicaciones desarrolladas en software propietario y ligadas a la plataforma Windows que permiten determinadas facilidades para la representación geoespacial de la información. Con el inconveniente de que son aplicaciones de escritorio, sin orientación a una arquitectura cliente-servidor. Se puede (por desconocimiento) volver a crear una cartografía que ya había sido creada.

Se propone con esta herramienta lograr la gestión y control de la información de forma geoespacial por la web de forma tal que sea accesible desde cualquier parte del territorio y por toda aquella aplicación con permiso para acceder a la misma. Esta aplicación tendrá la característica de que se desarrollará en ambiente libre y multiplataforma de acuerdo a la estrategia de migración a software libre (SWL) de la organización.

Con este trabajo se da un paso de avance en la automatización del proceso de control de la representación gráfica de la información al permitir la creación de un sistema capaz de facilitar el proceso de tomas de decisiones en las instituciones vinculadas a la defensa del país.

ÍNDICE

INTRODUCCIÓN	1
FUNDAMENTACIÓN	4
1.1. INTRODUCCIÓN	4
1.2. SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG).....	4
1.3. SISTEMA DE INFORMACIÓN GEOGRÁFICA EN EL MUNDO.	5
1.3.1 Los SIG y el Software Libre	6
1.4. APLICACIÓN DE SISTEMA DE INFORMACIÓN GEOGRÁFICA EN CUBA.....	7
1.5. SISTEMA DE INFORMACIÓN GEOGRÁFICA EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS (UCI).	9
1.6. METODOLOGÍA A UTILIZAR... ..	10
1.7. TECNOLOGÍAS ACTUALES... ..	12
1.7.1. Servicios Web.....	12
1.8. ARQUITECTURA ORIENTADA A SERVICIOS	18
1.9. HERRAMIENTAS A UTILIZAR	20
1.10. LENGUAJES DE PROGRAMACIÓN WEB	21
1.10.1. PHP.....	21
1.10.2. JavaScript.	23
1.11. CONCLUSIONES	23
CARACTERÍSTICAS DEL SISTEMA	24
2.1. INTRODUCCIÓN	24
2.2. OBJETO DE ESTUDIO.....	24

Índice

2.3. OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN.....	24
2.4. FLUJO ACTUAL DE LOS PROCESOS.....	24
2.5. ANÁLISIS CRÍTICO DE LA EJECUCIÓN DE LOS PROCESOS.....	25
2.6. PROCESOS OBJETO DE AUTOMATIZACIÓN	25
2.7. PROPUESTA DE SISTEMA.....	25
2.8. MODELO DE DOMINIO.....	26
2.8.1. Definición de las entidades y los conceptos principales.....	26
2.8.2. Representación del modelo de dominio.....	27
2.9. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE	28
2.10. DEFINICIÓN DE LOS CASOS DE USO.....	31
2.10.1 Definición de los actores.....	31
2.10.2. Diagramas de casos de uso del sistema a automatizar	32
2.10.3. Listado de casos de uso.	34
2.11. CONCLUSIONES	45
ANÁLISIS Y DISEÑO DEL SISTEMA	46
3.1. INTRODUCCIÓN	46
3.2. MODELO DE ANÁLISIS.....	46
3.2.1. Diagramas de Clases de Análisis	46
3.3. MODELO DE DISEÑO.....	50
3.3.1. Diagramas de Clases de diseño.....	50
3.5. PRINCIPIOS DE DISEÑO	63
3.7. CONCLUSIONES	65

IMPLEMENTACIÓN Y PRUEBA	66
4.1. INTRODUCCIÓN	66
4.2. MODELO DE DESPLIEGUE.	66
4.3. MODELO DE COMPONENTES.	68
4.3.1. Componente.....	68
4.4. CONCLUSIONES	71
CONCLUSIONES GENERALES	72
RECOMENDACIONES.....	73
BIBLIOGRAFÍA	74
GLOSARIO DE TÉRMINOS	76

INTRODUCCIÓN

En la Empresa la gestión y representación gráfica de la información se realiza mediante pequeñas aplicaciones de escritorio y en muchos casos de forma manual, lo que provoca que el flujo y actualización de la información sea muy lenta y en muchos casos se cometan errores. Estas aplicaciones de escritorio para la representación de algunos tipos de información de forma gráfica tiene muchas limitaciones: una de ellas es que no son multiplataforma, lo que provoca que si está hecha para Windows no se ejecute en Linux y viceversa; solo se puede trabajar con ellas de forma local, trayendo esto consigo que la distribución de la información se haga más compleja y se corra el riesgo de que ésta no esté del todo actualizada, no tiene orientación a una arquitectura cliente-servidor. Se puede (por desconocimiento) volver a crear una cartografía que ya había sido creada. Además, estas aplicaciones de escritorio son software propietario que representa un inconveniente ante la situación de la migración de la organización a software libre (SWL).

Este planteamiento nos conduce al siguiente **problema**: ¿Cómo lograr proveer a una aplicación de un servicio de directorio que permita conocer la información geoespacial en línea en la institución?

A tono con las nuevas estrategias de desarrollo de software en la empresa y la expansión de la conectividad en la red de la misma, se hace necesaria la modelación de un servicio de directorio que permita conocer la información geoespacial que se encuentra disponible en la red y la forma para acceder a la misma. De ahí la importancia del desarrollo de un sistema de este tipo, pues no existen antecedentes de ningún otro en nuestro país que aporte un servicio similar.

Con el desarrollo de este trabajo se pretende diseñar una aplicación que permita la gestión, control, de la información de forma geoespacial por la web de forma tal que sea accesible desde cualquier parte del territorio y por toda aquella aplicación con permiso para acceder a las mismas. Esta aplicación tendrá la característica de que se desarrollará en ambiente libre y multiplataforma de acuerdo a la estrategia de migración a SWL de la organización.

Analizando los aspectos señalados anteriormente, para el desarrollo de este trabajo se han definido como **Objeto de estudio** los procesos de gestión y control de la información geoespacial en línea en las grandes empresas, estos procesos aplicados a la Institución representan su **campo de acción**.

Apoyados en lo expuesto anteriormente se ha llegado a la siguiente **hipótesis**: si la entidad contara con un modelo de un subsistema de control de información geoespacial en línea entonces se lograría la implementación de una aplicación que permita el control de información geoespacial en línea.

Para lograr esto nos hemos trazado como **objetivos generales** la modelación de una herramienta libre y multiplataforma que permita la gestión y control de toda la información geoespacial disponible para los diferentes sistemas que necesiten hacer uso de la misma, mediante la utilización de Servicios Web, y como **objetivos específicos**:

- Diseñar Servicios Web accesibles a las aplicaciones que hacen uso de los mismos independientemente a la plataforma en que estén implementadas.
- Lograr la modelación de un software que contribuya al mejoramiento del proceso de toma de decisiones en la organización mediante el control de la información geoespacial en línea.
- Modelar y diseñar herramientas y servicios para brindar información sobre la base cartográfica distribuida en la red.

Tareas de la investigación:

- Estudio de los procesos gestión y control de la información geoespacial realizados hasta el momento en la organización.
- Estudio de las principales prácticas aceptadas de la representación geoespacial de la información en Cuba y el Mundo, así como el estado del arte de esta materia.
- Selección de la metodología de Análisis y Diseño de sistemas informáticos, que facilite y garantice la creación con calidad de la aplicación.
- Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación.

FUNDAMENTACIÓN

1.1. Introducción

En este capítulo se brinda una visión general de los aspectos relacionados con los Sistemas de Información Geográfica, sus disponibilidades a escala mundial y nacional, los conceptos necesarios para el estudio de los servicios Web, las características de cada tipo de herramienta, así como la descripción de los principales conceptos asociados al objeto de estudio, que son necesarios para entender los capítulos que le siguen.

1.2. Sistema de Información Geográfica (SIG)

Un SIG es un sistema de hardware, software y procedimientos diseñados para soportar la captura, administración, manipulación, análisis, modelamiento y graficación de datos u objetos referenciados espacialmente, para resolver problemas complejos de planeación y administración. Una definición más sencilla es: Un sistema de computador capaz de mantener y usar datos con localizaciones exactas en una superficie terrestre. Es una herramienta de análisis de información. La información debe tener una referencia espacial y debe conservar una inteligencia propia sobre la topología y representación. [1]

Entre los objetivos principales de un SIG es transformar datos geográficos en información confiable para la toma de decisiones, incrementar el conocimiento a partir de la relación datos-información.

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía. [1]

El SIG separa la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma y así realizar análisis multicriterio complejos.

Las principales cuestiones que puede resolver un Sistema de Información Geográfica son:

Localización: preguntar por las características de un lugar concreto.

Condición: el cumplimiento o no de unas condiciones impuestas al sistema.

Tendencia: comparación entre situaciones temporales o espaciales distintas de alguna característica.

Rutas: cálculo de rutas óptimas entre dos o más puntos.

Pautas: detección de pautas espaciales.

Modelos: generación de modelos a partir de fenómenos o actuaciones simuladas.

Por ser tan versátiles los sistemas de información geográfica, su campo de aplicación es muy amplio, pudiendo utilizarse en la mayoría de las actividades con un componente espacial. La profunda revolución que han provocado las nuevas tecnologías ha incidido de manera decisiva en su evolución.

1.3. Sistema de Información Geográfica en el mundo.

Atendiendo a todas las características que poseen y las funcionalidades que estos brindan, existen actualmente diversidad de SIG que se aplican en todo el mundo en diferentes ramas de la vida de un país, por ejemplo la medicina, el transporte y la agricultura.

A continuación les presentamos un listado de software SIG teniendo en cuenta cuáles de ellos son software libre y cuáles no.

Software propietario: ArcGIS (ArcView, ArcInfo), Mapinfo, Maptitude, Geomedia, Geoconcept, GenaMap, Autodesk Map, MicroStation Geographics, Bentley PowerMap, GeoWeb Publisher, SmallWorld, Manifold, Idrisi, MapPoint, TatukGIS, TNT mips, MiraMon.

Software libre (SWL): GRASS GIS, JUMP, MapServer, Quantum GIS, gvSIG, SAGA GIS, MapWindow GIS, Kosmo.

Hemos querido tratar brevemente las características principales de algunas de estas tecnologías teniendo en cuenta que solamente nos interesan aquellas que sean SWL debido a las estrategias de migración que está llevando a cabo la empresa con la que estamos trabajando.

1.3.1 Los SIG y el Software Libre

La revolución que ha supuesto la filosofía del software libre ha llegado también a los SIG y hoy en día existen ya varios proyectos serios que pretenden ser una alternativa viable a programas comerciales como ArcView, Geomedia o IDRISI.

Se puede considerar Sistemas de Información Geográfica de sobremesa como aquellas aplicaciones SIG orientadas al trabajo normal de visualización, análisis, edición y salidas gráficas de información geográfica. [2]

Como una de estas aplicaciones de sobremesa podemos mencionar a **GRASS GIS**. GRASS (*Geographic Resources Analysis Support System*) fue inicialmente concebido y desarrollado por el laboratorio de investigación del cuerpo de ingenieros del ejército de los Estados Unidos (USA-CERL) para la gestión del territorio y la gestión medioambiental. GRASS comenzó a difundirse en ámbitos educativos y de instituciones públicas y se desarrollaron numerosas aplicaciones alrededor de dicho sistema, hasta que en 1999 pasó a tener licencia del tipo GNU GPL. Este SIG es uno de los de más rodaje, por lo que el número de herramientas y utilidades que presenta es muy elevado, pero presenta como inconveniente que está diseñado para entornos UNIX; esto, unido a su complejidad de uso y a su interfaz poco amigable, ha frenado su expansión para el público en general.

También existe **Jump**, que fue uno de los primeros SIG gratuitos y por lo tanto ha servido de base a otros desarrollos, tanto públicos como desarrollados por empresas de programación. Se trata de un SIG modular programado en Java y que basa su funcionalidad en módulos (*plugins*). Es posible conectarse a servidores de cartografía WMS y existen plugins para numerosos de formatos tanto de archivo como de servidores. Una de las cosas más interesantes son las herramientas de edición de que se disponen para modificar datos vectoriales, así como herramientas básicas de geoprocesamiento (zonas de influencia,

uniones, etc.). Actualmente han aparecido versiones internacionalizadas y varias páginas que albergan proyectos relativos a Jump, tanto para la creación de nuevos plugins como proyectos que basándose en Jump pretenden generar nuevos programas con funcionalidades más específicas. Cabe señalar como una debilidad de Jump el hecho de que se echan en falta algunas funcionalidades básicas como por ejemplo la impresión de cartografía, cuadrículas, etc., además de que existe cierta descoordinación en la generación de versiones.[2]

SAGA (*System for Automated Geoscientific Analyses*) es un SIG que está programado en C++. Está más orientado al tratamiento de datos matriciales que a la información vectorial. Tiene sus orígenes en el Instituto Geográfico de Göttingen (Alemania). SAGA también se basa en una filosofía modular para la ampliación de funcionalidades y formatos soportados. Tiene como puntos fuertes el poseer una interfaz gráfica realmente madura y conseguida y la posibilidad de programación de módulos por parte del usuario. Cabe señalar que este SIG está desarrollado para Windows, no está basado en estándares del OGC, ni soporta datos en servidores. [2]

El principal problema de estas aplicaciones es que al contrario de las aplicaciones comerciales, en estos programas es posible que se echen en falta algunas funciones básicas que si nos brindan las comerciales. Pero queda claro que teniendo en cuenta la situación de bloqueo económico que se vive en nuestro país, que nos obliga a prescindir de muchos productos de software a nivel mundial por la prohibición de ventas de licencias a Cuba, es el software libre nuestra mejor elección.

1.4. Aplicación de Sistema de Información Geográfica en Cuba.

El tratamiento geográfico de la información en los últimos decenios ha cobrado un auge vertiginoso a escala mundial, cada día con mayores posibilidades de aplicación gracias al desarrollo de la aparición de los Sistemas de Información Geográfica (SIG). Las expectativas creadas sobre SIG están también presentes en Cuba con sus correspondientes limitaciones y paradojas.

A pesar de que existen SIG creados y desarrollados por empresas españolas, holandesas, brasileñas, etc., que podrían adquirirse con sus licencias, una gran parte de los especialistas cubanos prefieren los SIG desarrollados por compañías de EE.UU., con la desventaja de que estos no tienen licencia de utilización de usuario debido a que las compañías estadounidenses no están autorizadas a vender este software a los cubanos. [3]

Ante la imposibilidad de obtener SIG comerciales, en Cuba se crea Digicapt (un módulo del Sistema de Información Geográfica de Cuba (SIGC)), pero ante la existencia de potentes Sistemas de Información Geográfica el primer SIG cubano quedó obsoleto. En la década de los noventa llegó a Cuba otro SIG de formato raster, cuando se inició el auge de las computadoras personales. El IDRISI, en su primera versión para D.O.S, fue utilizado por algunos pocos especialistas del Instituto de Ecología y Sistemática (Academia de Ciencias de Cuba). Ya en el año 2005 este SIG era poco empleado en nuestro país. [3]

En esa misma época la Facultad de Geografía de la Universidad de La Habana recibió la donación de un SIG llamado ILWIS, desarrollado en Holanda. En el presente, el ILWIS es utilizado por unos pocos especialistas cubanos que han recibido adiestramiento en el ITC (Institute for Aerospace Survey and Earth Sciences) de Holanda y que incluso tienen sus licencias de operación. [3]

Cuba recibió además otros SIG como el ATLAS GIS, proveniente de México a mediados de los noventa. La versión 3.01 del Atlas GIS para Windows (AGISW) también fue una donación de científicos foráneos amigos aunque se utilizó realmente poco pues ya a mediados de los noventa se introdujo y se diseminó rápidamente entre los especialistas el MapInfo, que no es un SIG potente, pero es muy amigable y relativamente fácil de manejar. [3]

En la década del noventa los cubanos deseaban tener su propio Sistema de Información Geográfica; de esta forma surgió el TELEMAP, el único SIG diseñado, creado y producido en Cuba por especialistas del antiguo Instituto Cubano de Hidrografía, una institución que hoy forma parte de GEOCUBA. En su momento fue muy utilizado, fue un logro de los técnicos cubanos, pero con el inconveniente de que requería una llave de usuario para su operación y además era poco compatible

con otros SIG, por lo que ya no se trabaja con él. Actualmente los cubanos ya utilizan el ArcGIS, que al parecer ha tomado lo mejor del AtlasGIS y del ArcView para lograr un mejor y potente Sistema de Información Geográfica. [3]

1.5. Sistema de Información Geográfica en la Universidad de las Ciencias Informáticas (UCI).

Como hemos podido ver, el desarrollo de los SIG a escala mundial es cada vez mayor, y la Universidad de las Ciencias Informáticas no ha dejado pasar por alto este hecho. Teniendo en cuenta que la extensión territorial del centro no ha dejado de crecer desde su creación, el dinamismo que lo caracteriza, y que cada vez se hace más complejo el manejo y localización de la información, desde hace unos años se han venido realizando estudios sobre la creación de sistemas que se encarguen de gestionar la información geográfica. Han sido tres los Trabajos de Diploma que sobre este tema se han presentado: *Sistema de Información Geográfica sobre WEB de la UCI*, *Sistema de Información Geográfica de la UCI basado en tecnología OpenSource*, y *Sistema de Información Geográfico de la UCI. Gráficos Estadísticos y Mapas Temáticos*. Actualmente se encuentra en funcionamiento una aplicación que permite realizar localizaciones y brinda la información que haya disponible sobre el lugar localizado por el usuario.

1.6. Metodología a utilizar...

RUP:

El Proceso Unificado de Desarrollo (RUP). Es un proceso de ingeniería de software basado en la modelación de sistemas informáticos usando tecnología orientada a objetos, provee un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización desarrolladora o cualquier proyecto de software. Entre sus características se encuentra el desarrollo del software de manera iterativa, usa arquitecturas basadas en componentes, verifica continuamente la calidad del producto, propone la creación de artefactos como herramientas visuales con el uso de UML. Siempre se conoce el estado del proyecto, las inconsistencias entre análisis, diseño e implementación se detectan tempranamente, el cliente obtiene resultados a corto plazo, y las pruebas se concentran en los aspectos de mayor riesgo.

En el desarrollo de este trabajo se utiliza RUP y como notación **UML** para su representación visual, ya que este es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables.

Son estas tres características las que distinguen precisamente a RUP de otras metodologías de desarrollo:

Proceso dirigido por Casos de Uso

Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema. En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo. Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

Proceso iterativo e incremental

El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo.

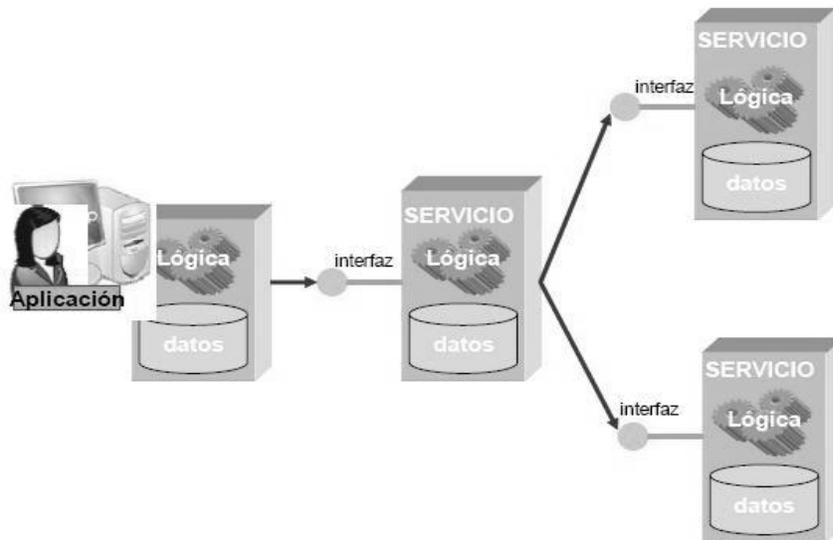
Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

Otros aspectos a tener en cuenta a la hora de elegir esta metodología de desarrollo son, en primer lugar, el hecho de que RUP es explícito en la definición de artefactos y su trazabilidad, contempla en relación causal de los artefactos creados desde los requerimientos hasta la implementación y las pruebas lo que facilita el seguimiento y el control del proyecto; como segundo aspecto importante podemos señalar que RUP identifica claramente a los profesionales (trabajadores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades, indicando qué trabajador es responsable de qué artefacto en cada actividad.

1.7. Tecnologías actuales...

1.7.1. Servicios Web

Los servicios web son componentes de software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, utilizando la Internet. Son aplicaciones independientes de la plataforma, que pueden ser fácilmente publicadas, localizadas o invocadas a través de protocolos Web estándar, como XML, WSDL, SOAP o UDDI. Los Servicios Web realizan funciones que pueden ser de todo tipo: desde simples peticiones hasta complejos procesos de negocio. Una vez puesto en marcha, otras aplicaciones pueden localizarlo e invocar el servicio. Los objetivos principales que persiguen los servicios web son la interoperabilidad y la integración. Esta última permite obtener la información solicitada en tiempo real, agilizando el proceso de toma de decisiones. [11]

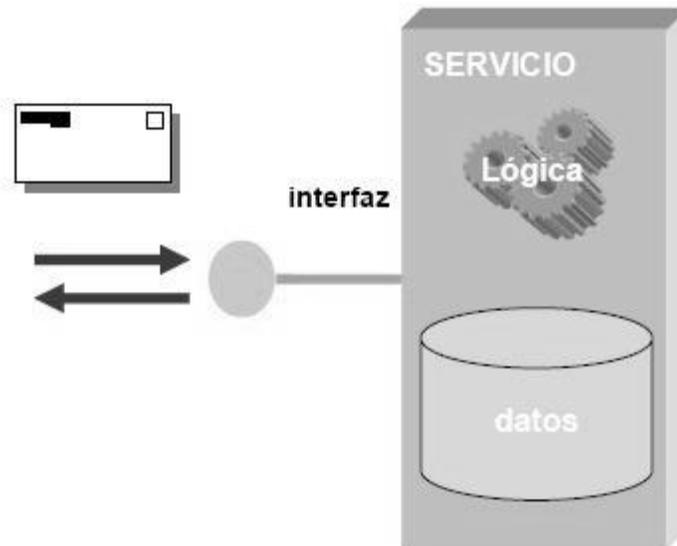


Sistema basado en servicios.

En la figura se representa un sistema sencillo basado en servicios. Una aplicación además de implementar sus propios componentes de negocio y datos, también puede reutilizar la funcionalidad de servicios existentes en la red Empresarial. [11]

Un servicio debe ser una aplicación completamente autónoma e independiente. A pesar de esto, expone una interfaz de llamado basada en mensajes, capaz de ser accedida a través de la red. Generalmente los servicios incluyen tanto la lógica de negocio como los datos relevantes a la solución del problema para el cual fueron diseñados. [11]

La comunicación hacia y desde el servicio, se realiza utilizando mensajes que deben contener o referenciar toda la información necesaria para entenderlo.



Visión interna de los servicios.

Ventajas de los servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

Protocolos que utilizan los servicios web

SOAP

Es un protocolo basado en XML para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Existen varios modelos de mensajes SOAP, pero el más común es el RPC (remote procedure calls), en donde un nodo de la red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y este último responde al mensaje del cliente. Estos mensajes son independientes del sistema operativo y pueden transportarse en protocolos como SMTP, MIME y HTTP. [8]

SOAP ha recibido un increíble apoyo por parte de la industria. SOAP es el primer protocolo de su tipo que ha sido aceptado prácticamente por todas las grandes compañías de software del mundo. Compañías que en raras ocasiones cooperan entre sí están ofreciendo su apoyo a este protocolo. Algunas de las mayores Compañías que soportan SOAP son Microsoft, IBM, SUN, Microsystems, SAP y Ariba. [8]

Algunas de las Ventajas de SOAP son:

- **No está asociado con ningún lenguaje:** los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.

- **No se encuentra fuertemente asociado a ningún protocolo de transporte:** La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.

- **No está atado a ninguna infraestructura de objeto distribuido:** La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.

- **Aprovecha los estándares existentes en la industria:** Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.

- **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrollo sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas.

WSDL

El lenguaje de descripción de servicios Web (WSDL, Web Service Description Language) es un dialecto basado en XML sobre el esquema que describe un servicio Web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML como DCOM sobre UDP. [7]

Los documentos WSDL definen los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Las especificaciones concretas del protocolo y del formato de datos para un tipo de puerto determinado constituyen un enlace reutilizable. Un puerto

se define por la asociación de una dirección de red y un enlace reutilizable; una colección de puertos define un servicio. [7]

Describe cuatro piezas principales:

- Información sobre la interfaz exportada.
- Tipo de datos de los mensajes intercambiados (peticiones, respuestas, errores).
- Información de “brindar” definido cómo trasportar los mensajes en la comunicación.
- Dirección para localizar los servidores.

XML

XML, eXtensible Markup Language o lenguaje de anotación extensible. Ya conocemos el lenguaje HTML (*hypertext markup language*), lenguaje de anotación para página webs que permite navegación tipo hipertexto; sin embargo, XML no es sólo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible. Todo lenguaje que se exprese de una forma determinada puede ser XML. Por lo tanto, XML *no es un lenguaje para hacer mejores páginas web*, sino un lenguaje para información auto-descrita, o al menos, auto-descrita si las etiquetas están bien puestas. No ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. [9]

UDDI

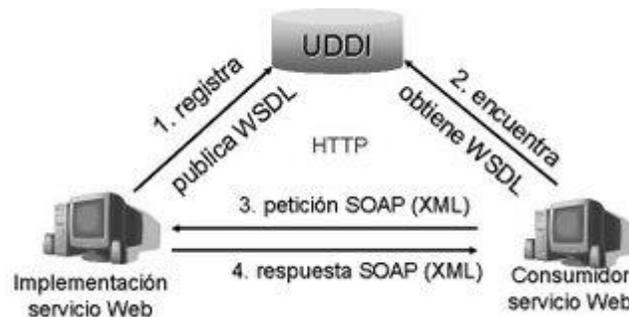
UDDI (Descripción Universal, Descubrimiento e Integración) es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. A

través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios. Se puede utilizar sistemas taxonómicos estándar para clasificar estos datos y poder encontrarlos posteriormente en función de la categorización. Lo más importante es que UDDI contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a API XML basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos.

Los servicios UDDI ofrecen beneficios estratégicos a las empresas que operan con servicios Web. Como pieza clave de la infraestructura de servicios Web dentro de Windows Server 2003, Servicios UDDI facilita el descubrimiento, compartición y reutilización de servicios Web. [10]

1.8. Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. Al contrario de las arquitecturas orientado a objetos, las SOAs están formadas por servicios de aplicación débilmente acoplados y altamente interoperables. Los Servicios Web se han convertido en el estandarte de SOA, ya que esta tecnología posee un conjunto de características que permiten cubrir todos los principios de la orientación a servicios. La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo. Con esta arquitectura, se pretende que los componentes software desarrollados sean muy reusables, ya que la interfaz se define siguiendo un estándar. En la figura, presentamos la estructura básica de funcionamiento de un SOA tradicional. [4]



En el gráfico anterior, se puede observar la existencia de tres roles claramente diferenciados:

- Cliente del servicio: Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.
- Proveedor del servicio: Es el encargado de implementar el servicio web y ofrecerlo a los clientes.
- Registro del servicio: Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

La secuencia de ejecución es la siguiente:

1. El proveedor del servicio da de alta el servicio web en el registro. Para realizar esto, el proveedor almacena en el registro el documento de descripción de este.
2. El solicitante del servicio busca en el registro un servicio web que pueda adaptarse a sus necesidades.
3. Una vez seleccionado el servicio, el solicitante lo invoca mediante el envío de un mensaje SOAP, en el cual se indica la acción a realizar y los datos de entrada.
4. El servicio web recibe la petición y ejecuta la funcionalidad. Para finalizar envía un mensaje SOAP al solicitante con los resultados obtenidos.

1.9. Herramientas a utilizar...

Visual Paradigm

Es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-uml también proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

Características:

- Producto de calidad.
- Soporta aplicaciones web.
- Las imágenes y reportes generados, no son de muy buena calidad.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Dreamweaver

Macromedia Dreamweaver es un editor de páginas web, creado por Macromedia (actualmente Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, por su soporte de los estándares del World Wide Web Consortium. Dreamweaver permite al usuario utilizar la mayoría de los navegadores. Con la llegada de la versión MX, Macromedia incorporó herramientas de creación de contenido dinámico en Dreamweaver. En lo fundamental de las herramientas HTML también permite la conexión a Bases de Datos como MySQL y Microsoft Access, para filtrar y mostrar el contenido utilizando tecnología de script como, por ejemplo, ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (JavaServer Pages), PHP sin necesidad de tener experiencia previa en programación. Tiene una arquitectura extensible. Es decir, permite el uso de "Extensiones". Las extensiones, tal y como se conocen (normalmente en HTML y Javascript). [12]

1.10. Lenguajes de Programación Web

1.10.1. PHP

Es un lenguaje de script incrustado dentro del HTML. Permite la programación de páginas web del lado del servidor, sus características principales son independencia de plataforma y gratuidad. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes. [5]

Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente.
[5]

Por qué utilizar PHP y no otras opciones:

- PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
 - Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
 - Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un comercio electrónico, XML, creación de PDF ...)
 - Está siendo utilizado con éxito en varios millones de sitios Web.
 - Es multiplataforma, funciona en todas las plataformas que soporten apache.
 - **Es software libre.** Se puede obtener en la Web y su código está disponible bajo la licencia GPL.
- Este es uno de los aspectos fundamentales para nuestra elección de este lenguaje.

Las cuatro grandes características de PHP son:

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. [6]

1.10.2. JavaScript.

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página Web. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Toda esta potencia de Javascript se pone a disposición del programador, que se convierte en el verdadero dueño y controlador de cada cosa que ocurre en la página.

1.11. Conclusiones

Con este capítulo queremos dar a conocer las herramientas y tecnología para darle solución a un problema que tiene alta prioridad dentro de la institución porque mediante esta aplicación se mejorara grandemente gestión, procesamiento y representación de la información geoespacial en línea para la toma de decisiones.

CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En este capítulo se hace una descripción de los procesos del negocio a través de un Modelo de Dominio, describiendo cada uno de los conceptos que lo conforman para que sea más entendible el contexto en que se ubica el sistema. Se hace una descripción de la propuesta del sistema, se enumeran los requisitos funcionales y los no funcionales que deben tener el sistema, lo que permite hacer una concepción general del sistema, e identificar mediante los diagramas de casos de uso y las descripciones de cada caso de uso las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.2. Objeto de estudio

Procesos de gestión y control de la información para su representación geoespacial en línea en las grandes empresas.

2.3. Objetivos estratégicos de la organización

El objetivo estratégico de la organización es la preparación del país para la defensa y la lucha armada ante una posible agresión extranjera. Dentro de este objetivo general nuestro trabajo se concentra en el aumento del control, mediante la informatización, de todos los recursos materiales, financieros y humanos de la entidad.

2.4. Flujo actual de los procesos

Actualmente en la empresa los procesos de gestión y representación de la información cartográfica se realizan manualmente en la mayoría de los casos. Los mapas requeridos son dibujados por el personal en dependencia de la información que sea necesaria representar. Si los datos que se necesitan no están disponibles en ese momento hay que contactar a la entidad que los posee, mediante la telefonía o utilizando cualquier otro medio. Existen casos en los que para representar se

utiliza MapInfo, esto ocurre solamente si existe en el sistema un mapa digitalizado con el que se pueda manejar la información. Si, por el contrario, lo que se quiere es obtener información a partir de un mapa, el personal encargado debe de buscar en los archivos el correspondiente y, de él se extraen los datos a procesar.

2.5. Análisis crítico de la ejecución de los procesos

Teniendo en cuenta lo dicho anteriormente, podemos deducir que la representación y procesamiento de la información en la entidad son muy lentos y poco confiables, puesto que el hecho de que la mayoría de estos procesos se realicen de forma manual trae como consecuencia que la información corra el riesgo de estar desactualizada.

2.6. Procesos objeto de automatización

Se automatizará el proceso de control de la información geoespacial disponible para los diferentes sistemas que necesiten hacer uso de la misma, mediante la utilización de Servicios Web y otras tecnologías que faciliten el proceso de toma de decisiones.

2.7. Propuesta de sistema.

El sistema que se quiere modelar será basado en Servicios Web y otras tecnologías que facilitarán el proceso de toma de decisiones. Este sistema brindará diversas funcionalidades que servirán para facilitar la gestión y control de la información geoespacial disponible para los diferentes sistemas que necesiten hacer uso de la misma. Será accesible desde cualquier parte del país por aquellas personas que tengan el permiso para acceder al mismo.

Actualmente existe una serie de aplicaciones para el manejo de la información geográfica en línea pero ninguna basada en servicios Web.

2.8. Modelo de dominio

Teniendo en cuenta las descripciones de los procesos en el epígrafe anterior, nos damos cuenta de que el negocio que estamos estudiando, tiene muy bajo nivel de estructuración. Por tanto trataremos de dar un enfoque a todo el proceso de toma de decisiones.

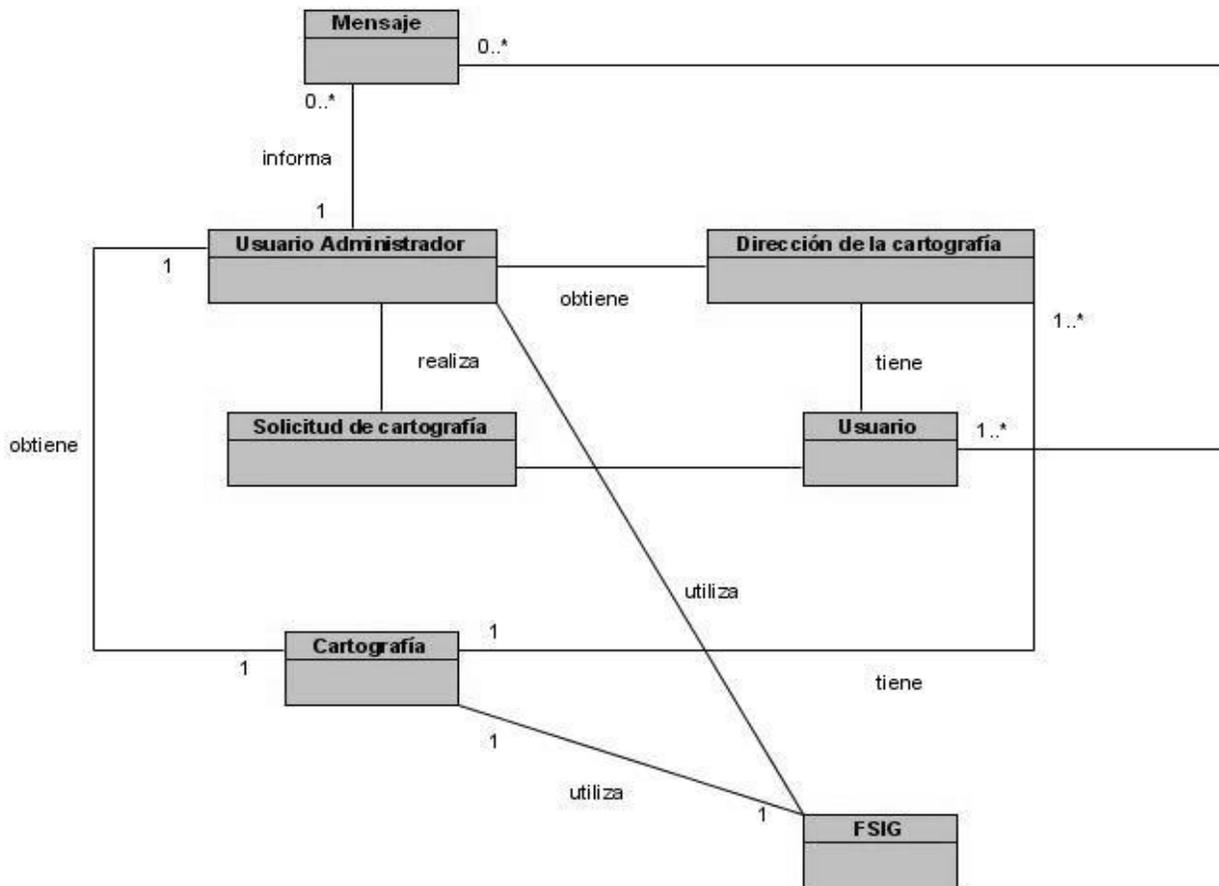
Para ello nos basaremos en un modelo del dominio, ya que nos permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Primeramente vamos a identificar todos los conceptos que se utilizarán en el diagrama, mediante las definiciones que se muestran a continuación.

2.8.1. Definición de las entidades y los conceptos principales.

Concepto	Descripción
Usuario Administrador	Representa a personas que realiza el proceso de gestión de la cartografía (solicitud, modificación).
Usuario	Representa a personas que contienen información de cartografía y se la brindan al administrador en caso de solicitud.
Dirección de la cartografía	Dirección donde se encuentran cartografía (web, ftp).
Solicitud de la cartografía	Representa la acción que realiza el administrador cuando desea obtener una cartografía.
Cartografía	Representa las capas que contienen información cartográfica.

FSIG	Aplicación externa, mediante la cual se modifican capas.
Mensaje	Acción que realiza el Usuario Administrador para informar a los usuarios de una nueva cartografía.

2.8.2. Representación del modelo de dominio.



2.9. Especificación de los requisitos de software.

Requisitos funcionales

Los requerimientos funcionales como bien nos sugiere, son capacidades o condiciones que el sistema debe cumplir, por lo que en nuestro caso este sistema que se está analizando debe tener la propiedad de satisfacer los requisitos siguientes:

R1. Configurar Sistemas

- Chequear que el usuario que solicita realizar esta operación está autorizado
- Permitir que el administrador configure un sistema.
- Permitir que el administrador modifique un sistema.
- Permitir que el administrador elimine un sistema.
- Guardar la información introducida o modificada.

R2. Registrar Sistemas

- Permitir que un sistema externo pueda ser registrado.
- Guardar la información introducida.

R3. Actualizar cartografía

- Permitir que el Sistema actualice información de las nuevas capas públicas.
- Permitir que el Sistema modifique información de las capas públicas.
- Permitir que el Sistema elimine información las capas que se dejaron de publicar
- Guardar la información introducida.

R4. Informar cambios en la base cartográfica

- Permitir que el sistema Informe que se ha publicado una capa.
- Permitir que el sistema Informe que se ha modificado una capa.

- Permitir que el sistema informe que se ha dejado de publicar una capa.

R5. Entregar información solicitada.

- Permitir que el sistema entregue información solicitada por un CIGEL Externo

R6. Reordenar componentes del sistema.

- Permitir que el sistema reordene los componentes de forma automática.

R7. Monitorear componente.

- Permitir que el sistema busque lista de los componentes que deben ser verificados
- Permitir que el sistema verifique cada uno de los componentes.
- Permitir que el sistema reintente la verificación según tenga configurado el componente.

Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa:

- Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- Empleo de los colores: verde, gris, blanco y azul principalmente, que son los definidos en los estándares del proyecto.

Usabilidad:

- El sistema podrá ser usado por un personal autorizado con el permiso para acceder al mismo.

Rendimiento:

- Tiempos de respuestas rápidos al igual que la velocidad de procesamiento de la información, no mayor a los 5 segundos en las actualizaciones y no mayor de 20 para las recuperaciones.

Soporte:

- Soporte para gran velocidad de procesamiento.
- Tiempo de respuesta rápido en accesos concurrentes.
- Versión de PHP 5.0.

Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

Portabilidad:

Necesidad de que el sistema sea multiplataformas.

Seguridad:

- Autenticación (contraseña de acceso)
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.

Funcionalidad:

Mínima cantidad de páginas para ejecutar todas las funciones posibles (preferentemente que estén relacionadas).

Software:

En secciones anteriores se ha mencionado que la construcción de nuestra aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto el servidor del usuario final debe tener como requerimientos mínimos de software:

Una computadora personal con plataforma del sistema operativo Windows Advanced Server 2000 o superior; o Linux.

Apache 2.0 o superior como servidor Web, con módulo PHP 5 disponible

Y la máquina cliente del usuario debe tener como requerimiento mínimo:

El navegador Mozilla FireFox.

Hardware

Partiendo del mismo supuesto que los requerimientos de software, nuestro modelo ideal (cliente/servidor), para los requerimientos mínimos de hardware, el usuario final debe tener un servidor con las siguientes características:

- Tarjeta de red.
- 128 Mb. de RAM o superior.
- 40 Gb. de disco duro o superior.
- Pentium II a 133 MHz de velocidad en su procesador o más.
- Una computadora que sirva de cliente:
- Pentium a 200 MHz. de velocidad de procesamiento o superior.
- 32 Mb. de memoria RAM superior.
- Tarjeta de red.

2.10. Definición de los casos de uso.

2.10.1 Definición de los actores.

Actores	Justificación
Administrador	Es el administrador del sistema, el que maneja toda la información de configuración del sistema.
CIGEL	Representa los sistemas externos que interactúan con el sistema CIGEL.
FSIG	Representa los sistemas externos que interactúan con el sistema CIGEL.

Sistema Externo.	Representa las aplicaciones que se registran en el sistema CIGEL y lo actualizan cuando hay cambios en la cartografía.
Reloj	Es el reloj el que inicia el proceso de monitorear componente, avisa cuando es el momento de hacer las verificaciones de cada uno de los componentes según tengan configurados.

Paquetes y sus relaciones



2.10.2. Diagramas de casos de uso del sistema a automatizar

A continuación se muestran los diagramas de Casos de Uso del sistema por cada uno de los paquetes.



Figura 1. Diagrama de los casos de uso del paquete: Configuración.

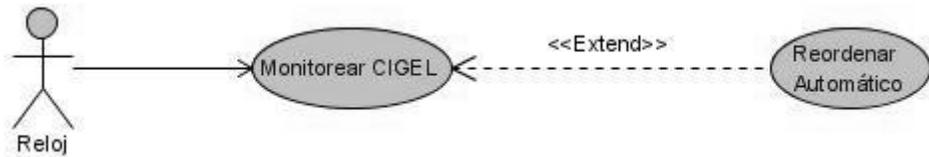


Figura 2. Diagrama de los casos de uso del paquete: *Gestionar Vitalidad*

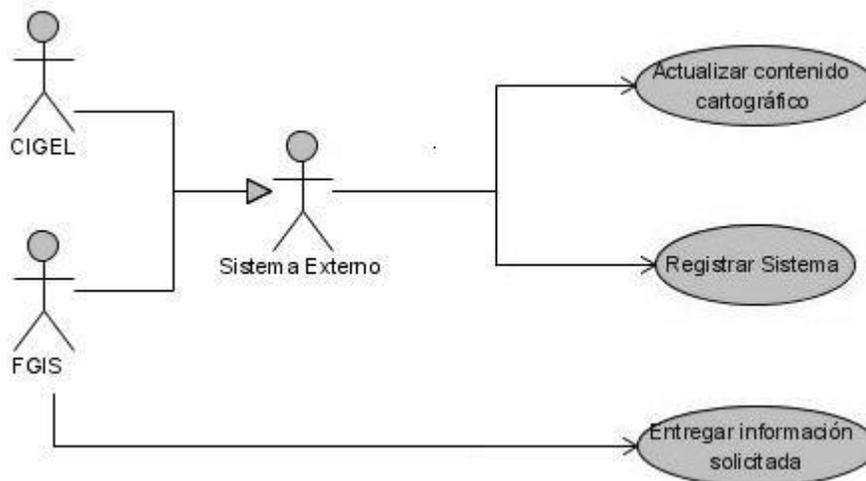


Figura 3. Diagrama de los casos de uso del paquete: *Gestionar Cartografía*.

2.10.3. Listado de casos de uso.

CU-1	Configurar Sistema
Actor	Administrador (inicia)
Descripción	El caso de uso se inicia cuando el administrador configura un CIGEL, modifica o elimina la configuración de un CIGEL. Este caso de uso termina cuando se han insertado, modificado o eliminado dicha configuración y la información ha sido actualizada.
Referencia	R1

CU-2	Registrar Sistema
Actor	Sistema externo (inicia), FGIS, CIGEL.
Descripción	Este caso de uso se inicia cuando el Sistema Externo solicita registrarse con un CIGEL al cual se va a conectar, enviándole los datos para sus datos y la referencia de su cartografía. El sistema almacena todos los datos enviados por el Sistema Externo. Este caso de uso termina cuando se ha almacenado toda la información.
Referencia	R2, R4

CU-3	Actualizar contenido cartográfico
Actor	FSIG (inicia)
Descripción	Este caso de uso se inicia cuando el FSIG informa actualizaciones en la cartografía, si hay una nueva capa pública, si se realizan cambios en alguna que ya está publicada, o si se ha dejado de publicar alguna capa. Este caso de uso termina cuando la información ha sido actualizada y el sistema informa a los demás sistemas que deben ser informados que ha habido cambios en la cartografía.

Referencia	R3,R4
-------------------	-------

CU-4	Entregar información solicitada
Actor	FSIG (inicia)
Descripción	Este caso de uso se inicia cuando el FGIS solicita información de una capa y el sistema le devuelve dicha información.
Referencia	R5

CU-5	Reordenar Automático
Actor	-
Descripción	Este caso de uso se inicia cuando ejecuta el caso de uso “Monitorear Componente”. Mediante este caso de uso el sistema realiza el reordenamiento de manera automática buscando reducir al mínimo las afectaciones que pueda provocar la salida inesperada de un componente del sistema.
Referencia	R6

CU-6	Monitorear Componente
Actor	Reloj (inicia).
Descripción	Este caso de uso se inicia cuando el Reloj avisa que debe realizar la verificación a los componentes. Mediante este caso de uso se detectan que elementos del sistema no están funcionando.
Referencia	R7

Casos de uso expandidos.

Caso de uso	
CU-1	Configurar Sistema
Propósito	Almacenar toda la información de configuración referente a un sistema.
Actores: Administrador (inicia)	
Resumen: El caso de uso se inicia cuando se crear, modificar o eliminar la configuración del sistema. Este caso de uso termina cuando se ha almacenado toda la información referente a un sistema.	
Referencias	R1

The image shows a dialog box titled "Configurar Sistema" with the following elements and callouts:

- 1: Botón Crear configuración (Create configuration button).
- 2: Botón Modificar configuración (Modify configuration button).
- 3: Botón Eliminar configuración (Delete configuration button).
- 4: Nombre de el CIGEL (Name of the CIGEL text input field).
- 5: Dirección del servicio CIGEL (CIGEL service address text input field).
- 6: Lista de selección. Tiempo de espera entre verificaciones (Dropdown menu for verification wait time, currently set to "5 min").
- 7: Nombre del CIGEL al que se va a conectar (Name of the CIGEL to connect to text input field).
- 8: Dirección del servicio CIGEL al que se va a conectar (CIGEL service address to connect to text input field).
- 9: Botón Aceptar (Accept button).
- 10: Botón Cancelar (Cancel button).

- (1) Botón Crear configuración.
- (2) Botón Modificar configuración.
- (3) Botón Eliminar configuración.
- (4) Nombre de el CIGEL
- (5) Dirección del servicio CIGEL
- (6) Lista de selección. Tiempo de espera entre verificaciones.
- (7) Nombre del CIGEL al que se va a conectar.
- (8) Dirección del servicio CIGEL al que se va a conectar.
- (9) Botón Aceptar
- (10) Botón Cancelar.

Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1.El administrador selecciona Configurar Sistema 3. Selecciona: Crear configuración Modificar configuración Eliminar configuración	2. El sistema muestra la interfaz “Configurar Sistema “. 4.Si el usuario selecciona: <ul style="list-style-type: none"> • ver sección “Crear configuración” • (2), ver sección “Modificar configuración” • (3), ver sección “Eliminar configuración”
Sección : Crear configuración	
Acción del actor	Respuesta del sistema
1.El administrador Introduce datos de configuración (4) , (5), (6), (7), (8) ,) luego presiona “(9)”	2. El sistema Verifica la integridad de los datos brindados 3. El sistema Guarda los datos de configuración. 4. El sistema Ejecuta el CU Registrar sistema.
Sección :Modificar configuración	
Acción del actor	Respuesta del sistema
1. El administrador introduce nuevos datos (4), (5), (6) luego presiona (9).	2. El sistema Verifica la integridad de los datos brindados por el usuario. 3. El sistema actualiza la configuración.
Sección :Eliminar configuración	
Acción del actor	Respuesta del sistema

1. El administrador presiona (3)	2. El sistema elimina la configuración del fichero de configuración.
Puntos de extensión.	
-	
Flujo alternativo sección Modificar Componente	
Acción del actor	Respuesta del sistema
1. Si el administrador modifica el parámetro (8).	2. El sistema ejecuta el caso de uso Reordenar Manual.

Caso de uso	
CU-2	Registrar Sistema
Propósito	Almacenar la referencia de la cartografía y la dirección para localizar al sistema que solicita el registro.
Actores: Sistema Externo (inicia), CIGEL, FGIS.	
Resumen: Este caso de uso se inicia cuando el sistema externo solicita registrarse con un CIGEL al cual se va a conectar. El sistema solicita datos para localizarlo. Este caso de uso termina cuando se ha almacenado toda la información y el resto de los componentes que deben ser informados del cambio han sido informados.	
Referencias	R6
Acción del actor	Respuesta del sistema
1. El Sistema Externo solicita: <ul style="list-style-type: none"> • Registrar • Eliminar el registro 	2. Si la opción solicitada por el Sistema Externo es: <p>Registrar ver sección “Registrar Componente”</p> <p>Eliminar registro, ver sección “Desregistrar</p>

	Sistema”
Sección :Registrar	
<p>1. El Sistema Externo envía su nombre, dirección, listado de cartografía.</p> <p>6. El Sistema externo recibe la información de vitalidad del sistema.</p>	<p>2. El sistema recibe los datos enviados por el Sistema Externo.</p> <p>3. El sistema guarda el nombre y la dirección en la configuración.</p> <p>4. El sistema guarda el listado de la cartografía.</p> <p>5. El sistema entrega al sistema externo su información de vitalidad.</p> <p>7. El sistema busca la lista de todos los componentes que deben ser informados del cambio.</p> <p>8. El sistema informa a los componentes que ha habido cambios en la base cartográfica, en este mensaje se envía los datos de esta cartografía.</p>
Sección :Des-registrar	
<p>1. El Sistema Externo envía su id.</p>	<p>2. El sistema verifica que el sistema existe.</p> <p>3. El sistema elimina al Sistema Externo de su configuración.</p>
Puntos de extensión.	
Obtener cartografía (incluido)	
Flujo alternativo sección Registrar	
Acción del actor	Respuesta del sistema
	<p>2. Si el nombre del componente es un CIGEL el sistema retorna su información de vitalidad.</p>

Caso de uso	
CU-3	Actualizar contenido cartográfico
Propósito	Recibir actualización de capas públicas.
Actores: Sistema Externo (inicia)	
Resumen: Este caso de uso se inicia cuando el FSIG informa actualizaciones en alguna de las capas, si hay una nueva capa pública, si se realizan cambios en alguna que ya está publicada, o si se ha dejado de publicar alguna capa. Este caso de uso termina cuando la información ha sido actualizada y el resto de los componentes que deben ser informados del cambio han sido informados.	
Referencias	R3, R4
Acción del actor	Respuesta del sistema
1. El FSIG informa actualización de capas.	2. El sistema recibe dicha información. 3. Si la información recibida es: Nueva capa pública ver sección "Agregar nueva capa". Capa pública modificada ver sección "Modificar capa". Eliminar capa pública ver sección "Eliminar capa".
Sección :Agregar nueva capa	
Acción del actor	Respuesta del sistema
1. El Sistema Externo especifica el nombre la referencia y descripción de la nueva capa pública.	2. El sistema verifica la integridad de los datos brindados. 3. El sistema procesa los datos. 4. El sistema busca la lista de todos los componentes que deben ser informados del cambio. 5. El sistema informa a los componentes que se hay una nueva capa en esta información se envían los datos de la nueva capa.

Sección :Modificar capa	
Acción del actor	Respuesta del sistema
1. El Sistema Externo envía específica el nombre de la capa y parámetros que fueron modificados.	2. El sistema verifica la integridad de los datos brindados. 3. El sistema guarda los cambios. 4. El sistema busca la lista de todos los componentes que deben ser informados del cambio. 5. El sistema informa a los componentes que se ha modificado una capa este mensaje contiene la modificación de la capa.
Sección :Eliminar capa pública	
Acción del actor	Respuesta del sistema
1. El FSIG especifica el nombre de la capa que ha dejado de ser pública.	2. El sistema busca la capa. 3. El sistema elimina la referencia de la capa de la 4. El sistema busca la lista de todos los componentes que deben ser informados del cambio. 5. El sistema informa a los componentes que se una capa a dejado de ser pública en este mensaje se envía el id de la capa.
Puntos de extensión.	
-	

Caso de uso	
CU-4	Entregar información solicitada.
Propósito	Entregar al sistema externos (FSIG) la información que solicite de alguna de las

	capas públicas.	
Actores: Sistema Externo (inicia)		
Resumen: Este caso de uso se inicia cuando el Sistema Externo solicita información de una capa o mapa y el sistema le devuelve dicha información.		
Referencias	R5	
Acción del actor	Respuesta del sistema	
1. El Sistema Externo solicita información referente a una capa y le pasa el id de la capa.	2. El sistema busca dicha información	
4. El Sistema Externo recibe información solicitada.	3. El sistema devuelve información solicitada.	
Puntos de extensión.		

Caso de uso		
CU-5	Reordenar Automático	
Propósito	Reordenar los nodos de forma automática.	
Actores: -		
Resumen: Este caso de uso se inicia cuando ejecuta el caso de uso "Monitorear CIGEL". Mediante este caso de uso el sistema realiza el reordenamiento de manera automática buscando reducir al mínimo las afectaciones que pueda provocar la salida inesperada de un componente del sistema.		
Referencias	R6	
Acción del actor	Respuesta del sistema	
	1. El sistema obtiene nodo fuera de servicio.	
	2. El sistema verifica que el nodo fuera de servicio es su padre.	
	3. El sistema obtiene nombre y dirección del padre de su padre.	

	4. El sistema solicita registrarse con el padre de su padre.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	<p>Línea 2:</p> <p>2. El sistema verifica que el nodo fuera de servicio es un hijo.</p> <p>3. El sistema elimina al hijo de la lista.</p> <p>Línea 3:</p> <p>3. El sistema no encuentra la dirección del padre de su padre y busca el primero de los hijos de su padre.</p> <p>4. El sistema solicita registrarse con el primero de los hijos de su padre.</p>
Puntos de extensión.	
-	

Caso de uso	
CU-6	Monitorear Componente.
Propósito	Detectar que elementos del sistema no están funcionando.
Actores: Reloj (inicia)	
Resumen: Este caso de uso se inicia cuando el Reloj avisa que debe realizar la verificación a los componentes. Mediante este caso de uso se detectan que elementos del sistema no están funcionando.	
Referencias	R7
Acción del actor	Respuesta del sistema
1. El Reloj avisa al sistema que debe hacer verificación de los componentes.	2. El sistema obtiene lista de todos los nodos que deben ser verificados.

	<ol style="list-style-type: none">3. El sistema verifica cada uno de los elementos de la lista.4. El sistema crea una lista con los elementos que no están activos.5. El sistema reintentar la cantidad de veces que tienen configurado.
Puntos de extensión.	
Reordenar Automático (Extensión)	

2.11. Conclusiones

En este capítulo se representó el modelo del dominio o modelo conceptual, el cual nos permite mostrar al usuario de forma visual los principales conceptos que se manejan en el dominio del sistema que se modela, de esta forma se puede comprender mejor cada uno de los conceptos que se manejan en la entidad. Se realizó la captura de los requisitos funcionales y no funcionales, descripciones y diagramas de casos de usos, los cuales permiten estar en condiciones de pasar a la próxima fase de propuesta de solución del problema, puesto que se han analizado y recopilado todos los elementos necesarios para la realización del proceso de realización de la aplicación.

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En el presente capítulo se realiza el diseño de nuestra propuesta. Se presentan los diagramas de clases del análisis y los del diseño los cuales permiten una mayor comprensión del problema para modelar la solución.

3.2. Modelo de Análisis

Contiene clases del análisis y sus objetos organizados en paquetes que colaboran. Mediante este flujo se profundiza en el dominio de la aplicación. Nos permite una mayor comprensión del problema para modelar la solución. Además, ofrece un mayor poder expresivo y una mayor formalización. Define una estructura que es una entrada esencial para modelar el sistema, incluyendo la creación del modelo de diseño.

3.2.1. Diagramas de Clases de Análisis

Artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada. Las clases de análisis están centradas en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Dichas clases tienen atributos y existen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos entre ellas. Estas clases se clasifican en:

- Entidad: Modelan información que posee una larga vida y que a menudo es persistente y fenómenos, conceptos y sucesos que ocurren en el mundo real.
- Interfaz: Modelan la interacción entre el sistema y sus actores.

- Control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

Diagrama de clases del análisis: *Configurar Sistema*

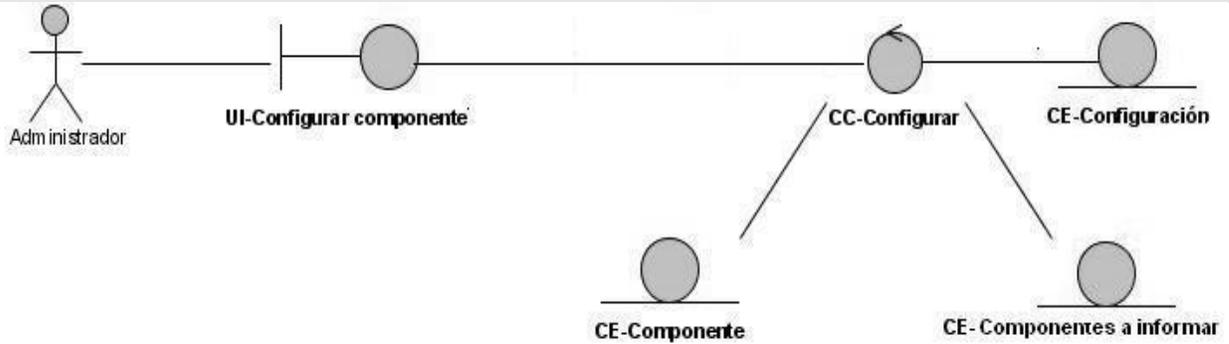


Figura 4. Diagrama de clases del análisis: *Configurar Sistema*

Diagrama de clases del análisis: *Registrar sistema*

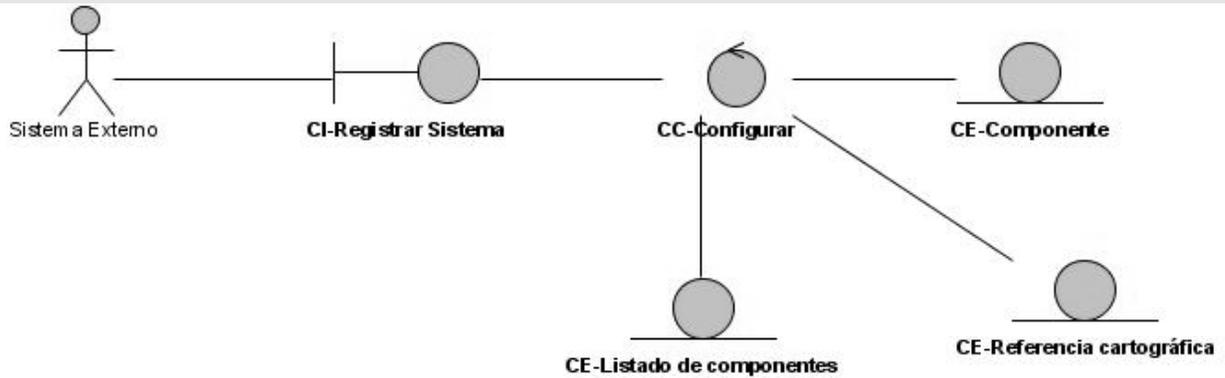


Figura 5. Diagrama de clases del análisis: *Registrar sistema*

Diagrama de clases del análisis: *Actualizar contenido cartográfico*

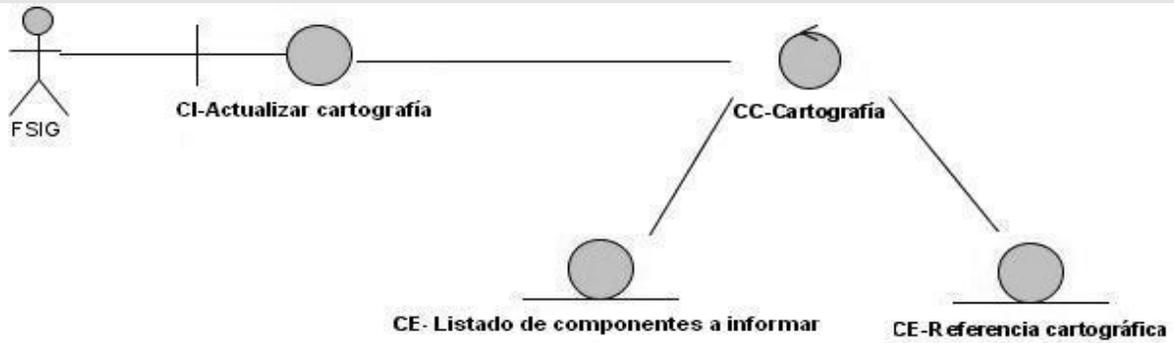


Figura 6. Diagrama de clases del análisis: *Actualizar contenido cartográfico*

Diagrama de clases del análisis: *Entregar información solicitada*



Figura 7. Diagrama de clases del análisis: *Entregar información solicitada*

Diagrama de clases del análisis: *Reordenar automático*

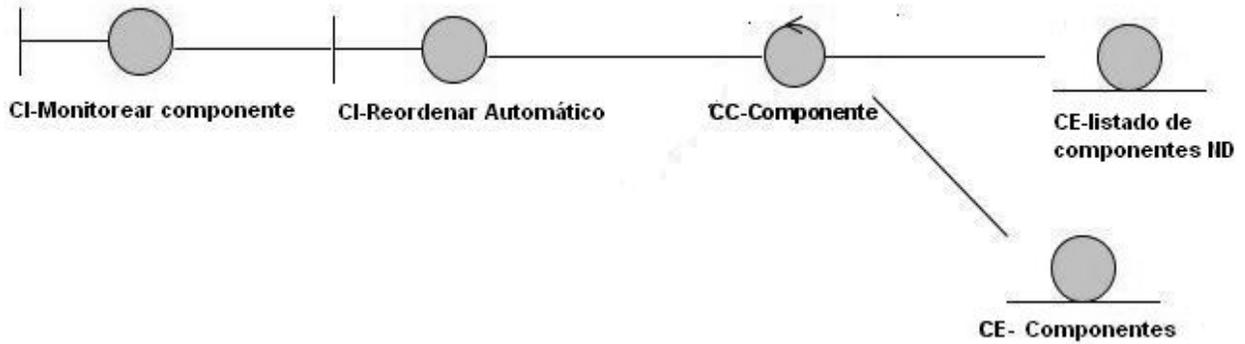


Figura 8. Diagrama de clases del análisis: *Reordenar automático*

Diagrama de clases del análisis: *Monitorear componente*

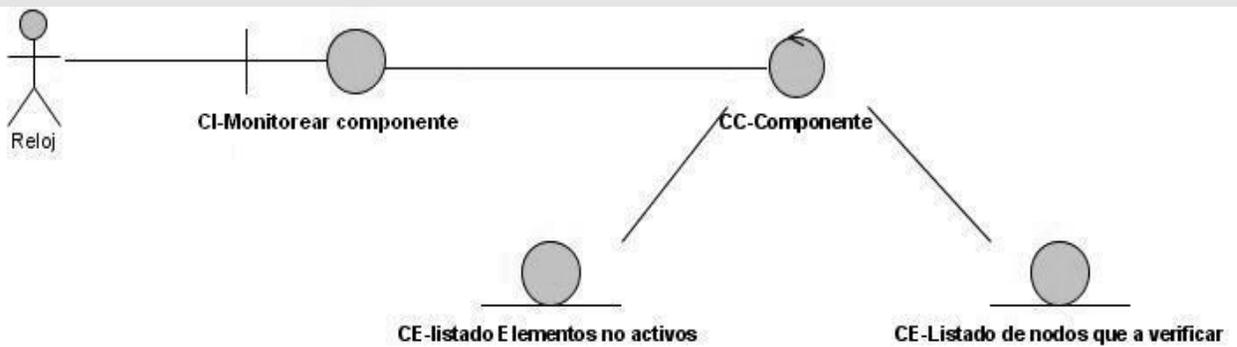


Figura 9. Diagrama de clases del análisis: *Monitorear componente*

3.3. Modelo de Diseño.

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales. La esencia de esta fase es la elaboración de diagramas de interacción, que muestran gráficamente como los objetos se comunican entre ellos a fin de cumplir con los requerimientos. Estos diagramas permiten la realización de los diagramas de clases del diseño, los cuales resumen la definición de las clases que se pueden implementar en el software. El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software.

3.3.1. Diagramas de Clases de diseño.

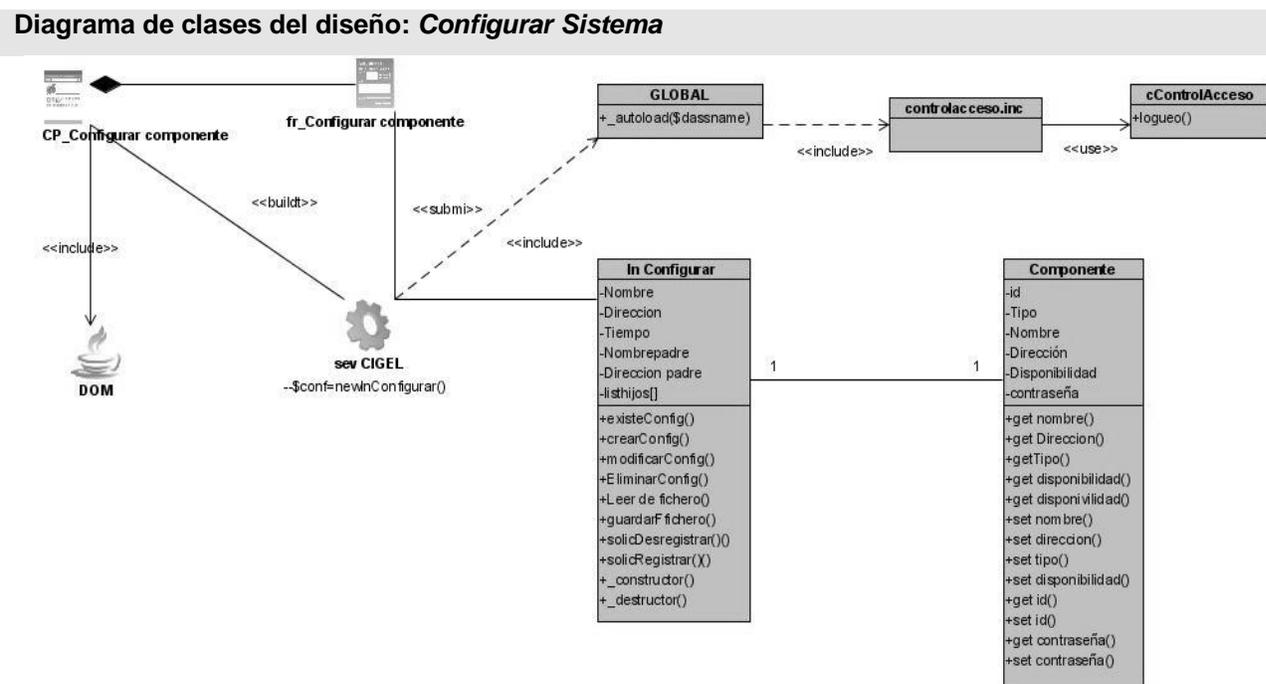


Figura 10. Diagrama de clases del diseño: Configurar Sistema

Diagrama de clases del diseño: Registrar sistema

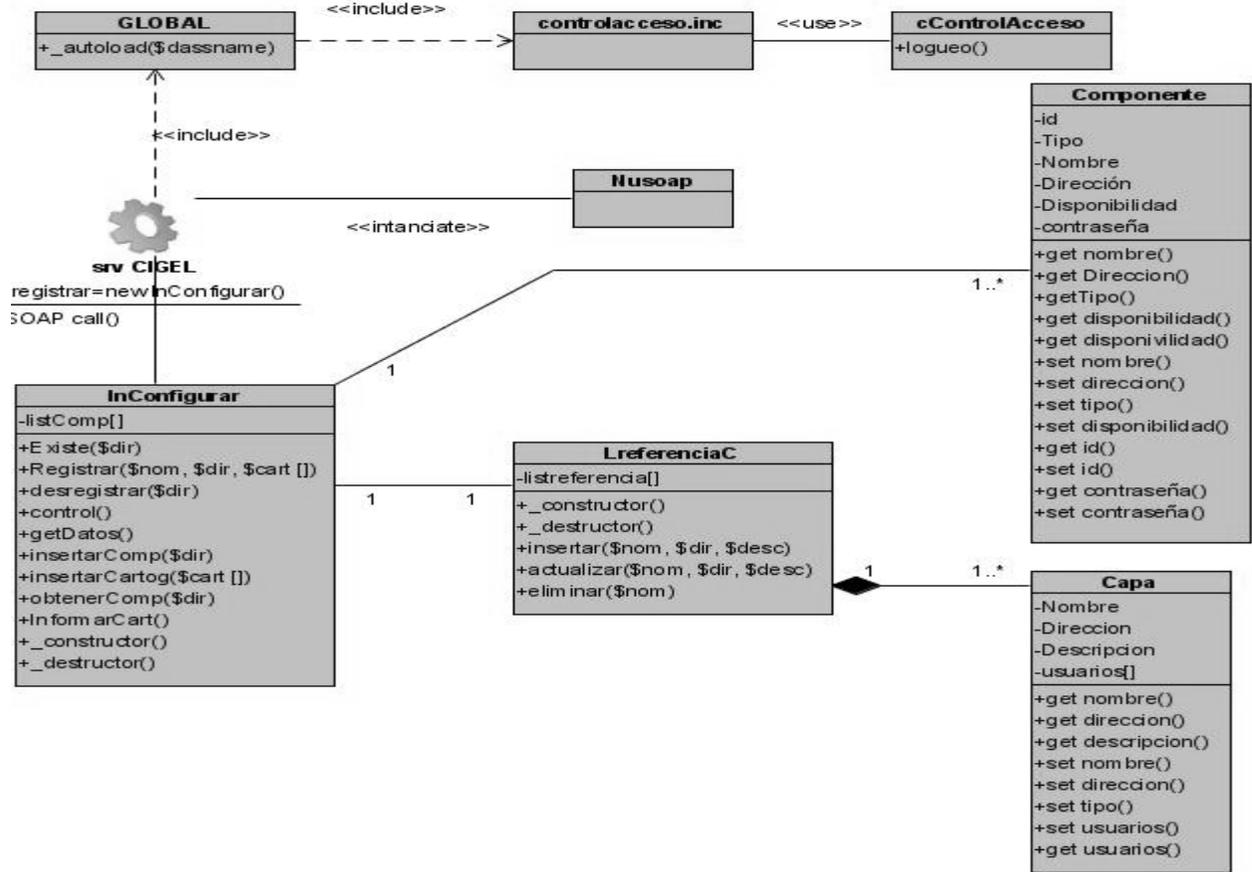


Figura 11. Diagrama de clases del diseño: Registrar sistema.

Diagrama de clases del diseño: *Actualizar contenido cartográfico*

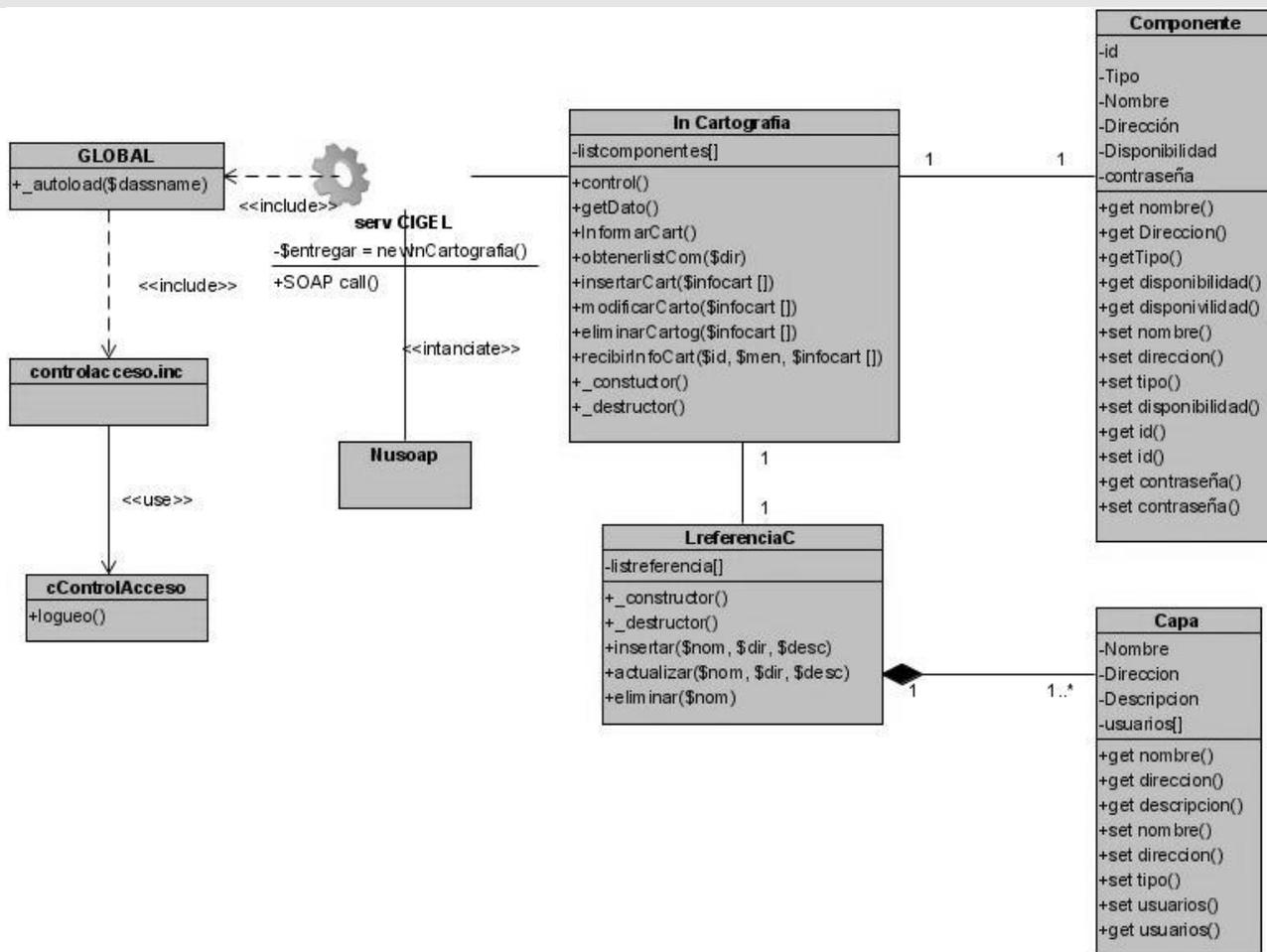


Figura 12. Diagrama de clases del diseño: *Actualizar contenido cartográfico*

Diagrama de clases del diseño: *Entregar información solicitada*

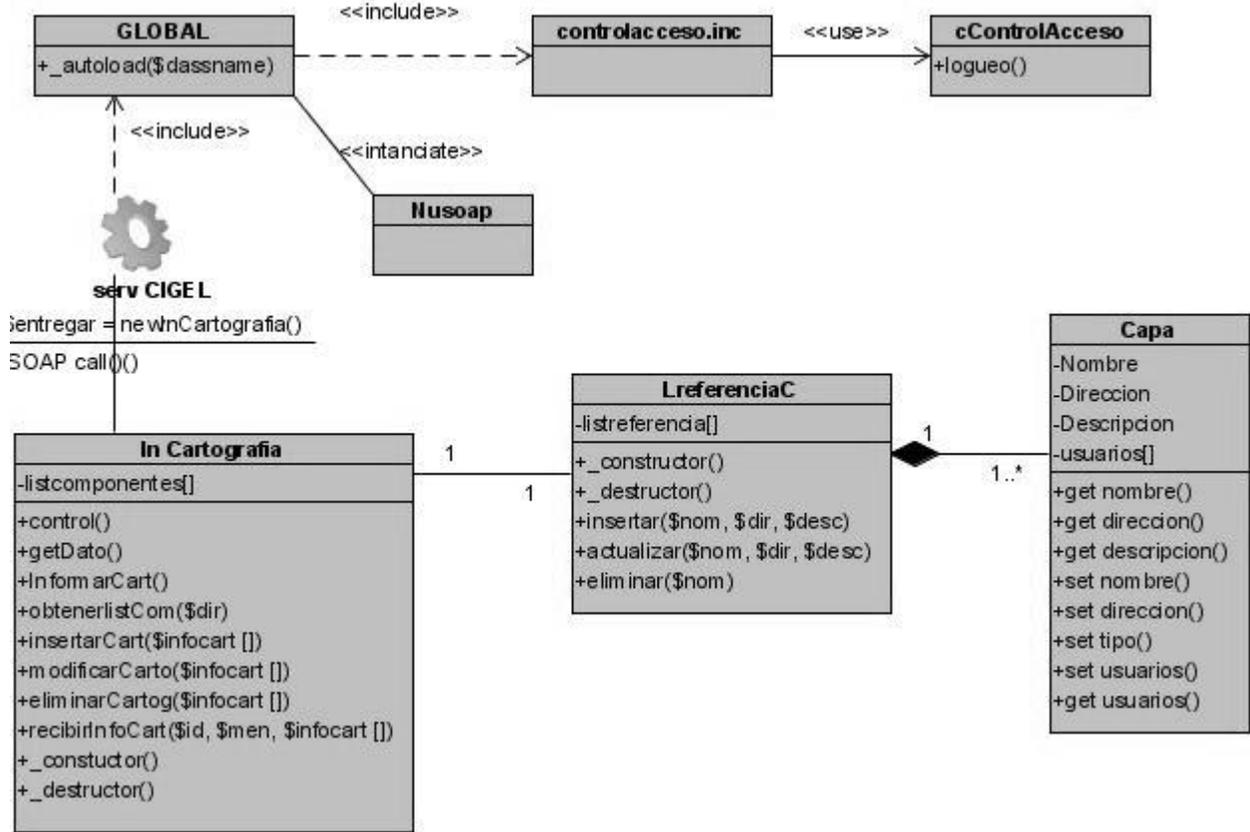


Figura 13. Diagrama de clases del diseño: *Entregar información solicitada*

Diagrama de clases del diseño: *Reordenar automático*

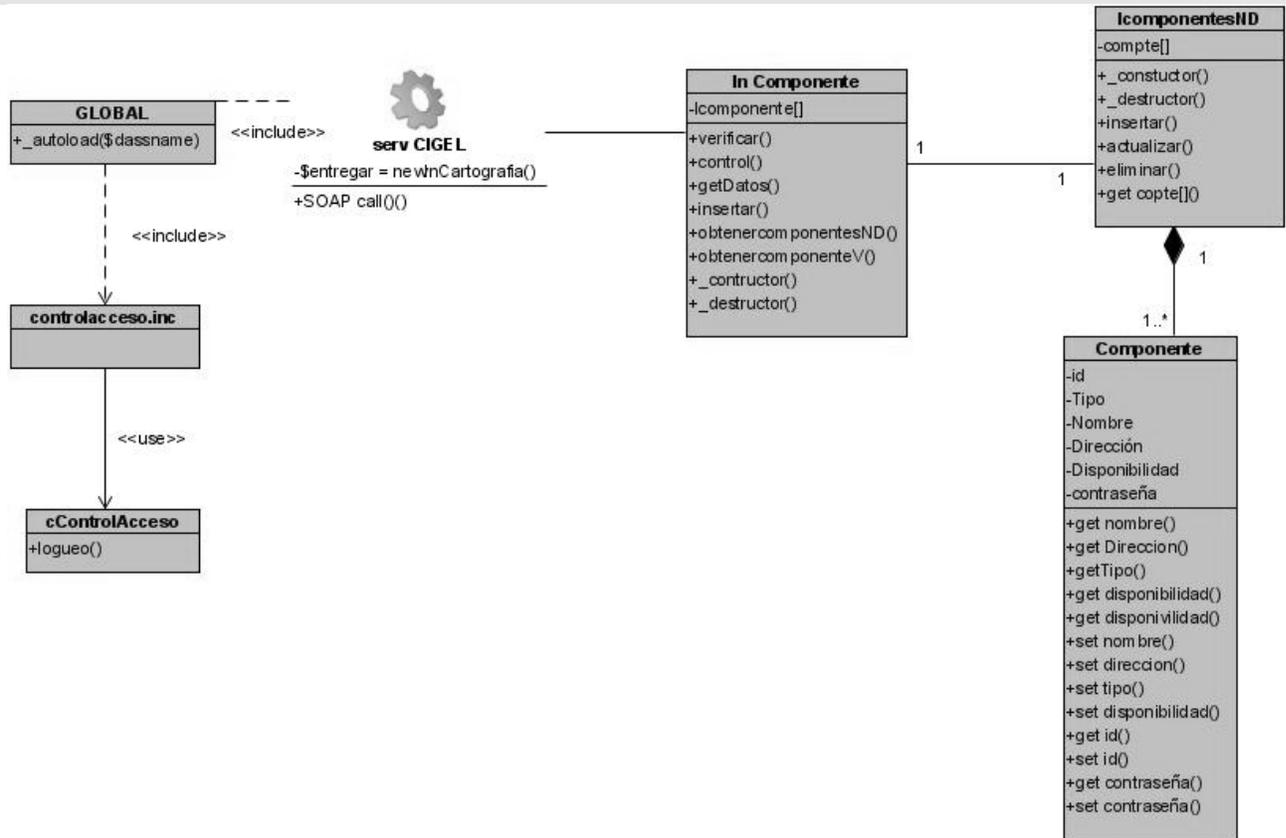


Figura 14. Diagrama de clases del diseño: *Reordenar automático*

Diagrama de clases del diseño: *Monitorear componente*

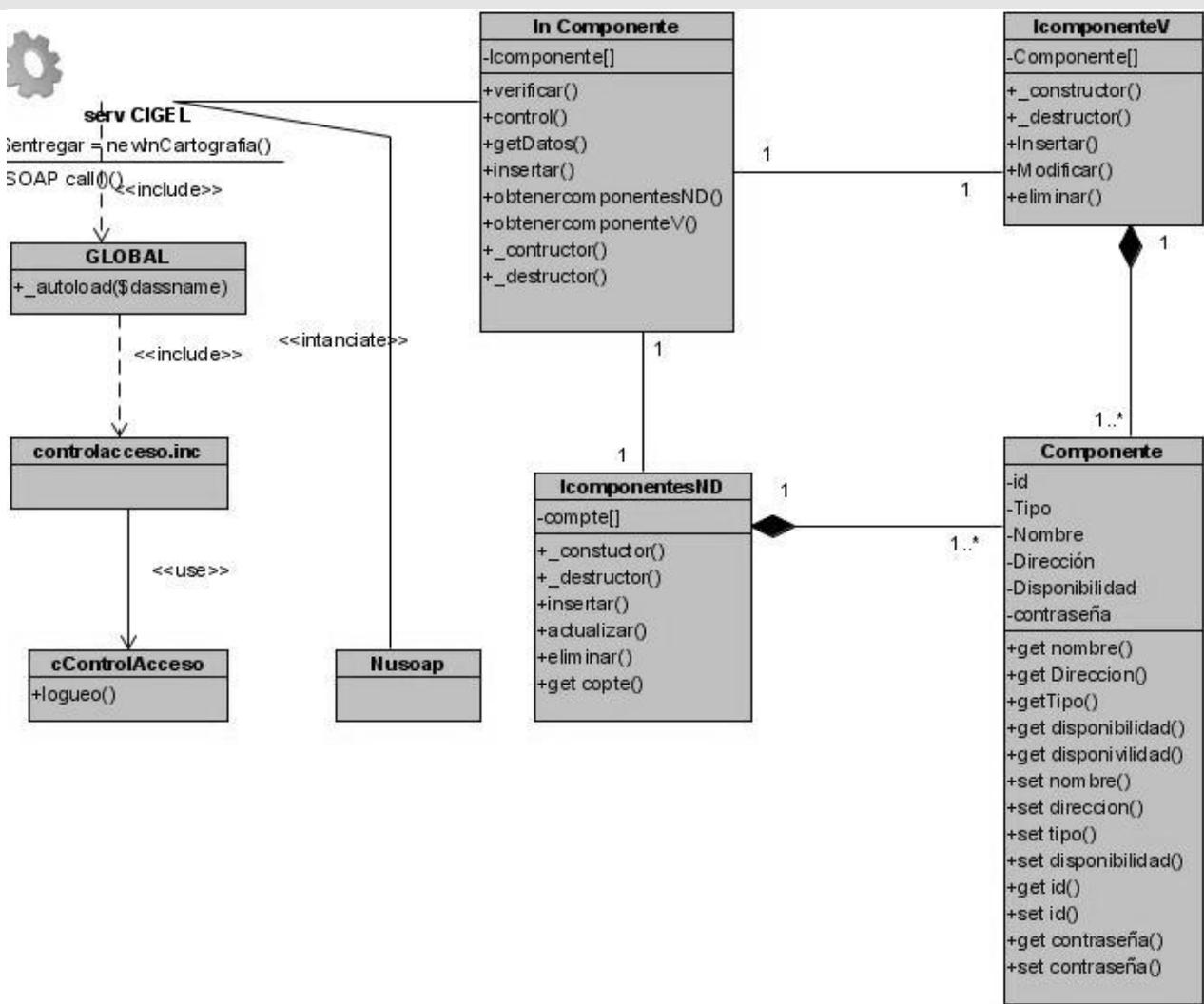


Figura 15. Diagrama de clases del diseño: *Monitorear componente*

3.4. Diagramas de secuencia.

Un diagrama de secuencia destaca la ordenación temporal de los mensajes. un diagrama de secuencia se forma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, alo largo del eje X. Normalmente, se coloca a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la derecha. A continuación, se colocan los mensajes que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo. Esto ofrece al lector una señal visual clara del flujo de control a lo largo del tiempo.

Diagrama de clases del diseño: Configurar Componente

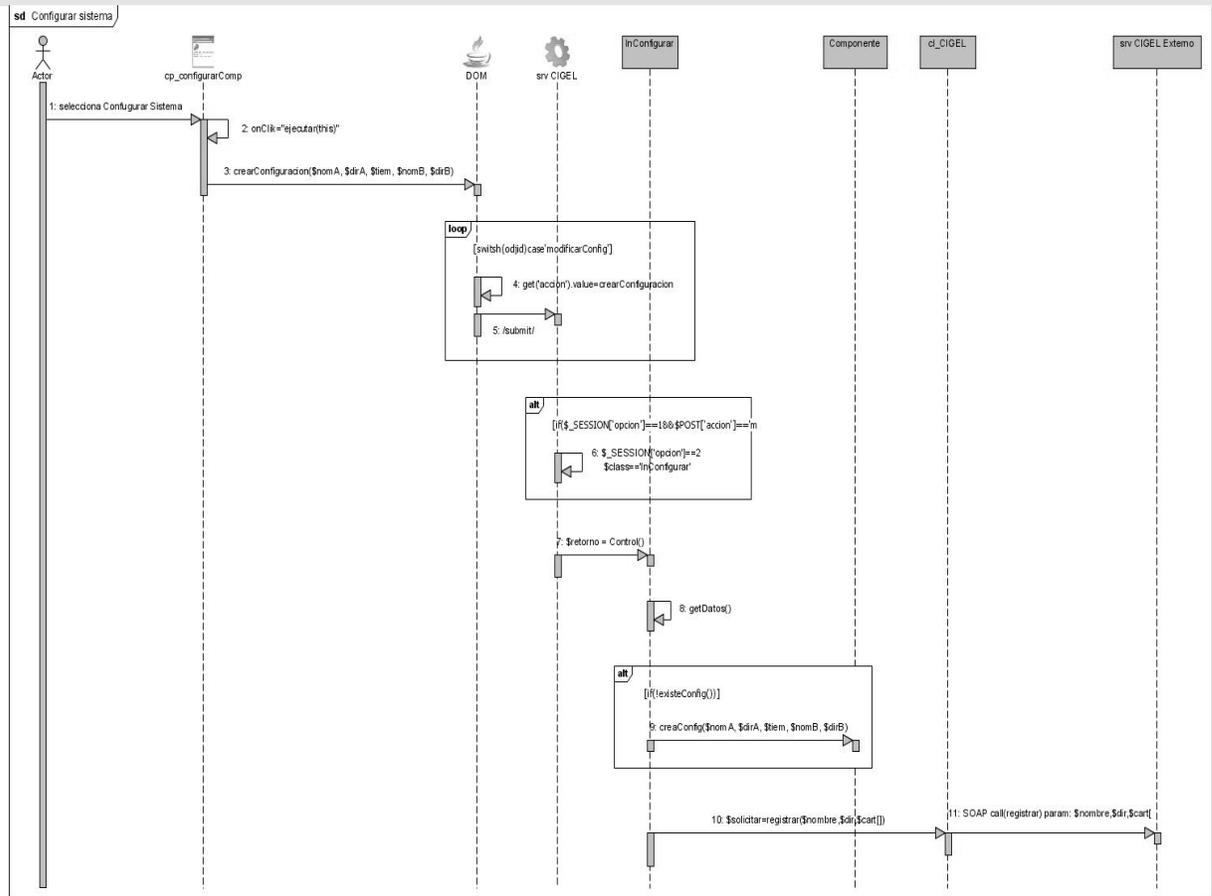


Figura 16. Diagrama de clases del diseño: Configurar Componente

Diagrama de clases del diseño: *Modificar Configuración*

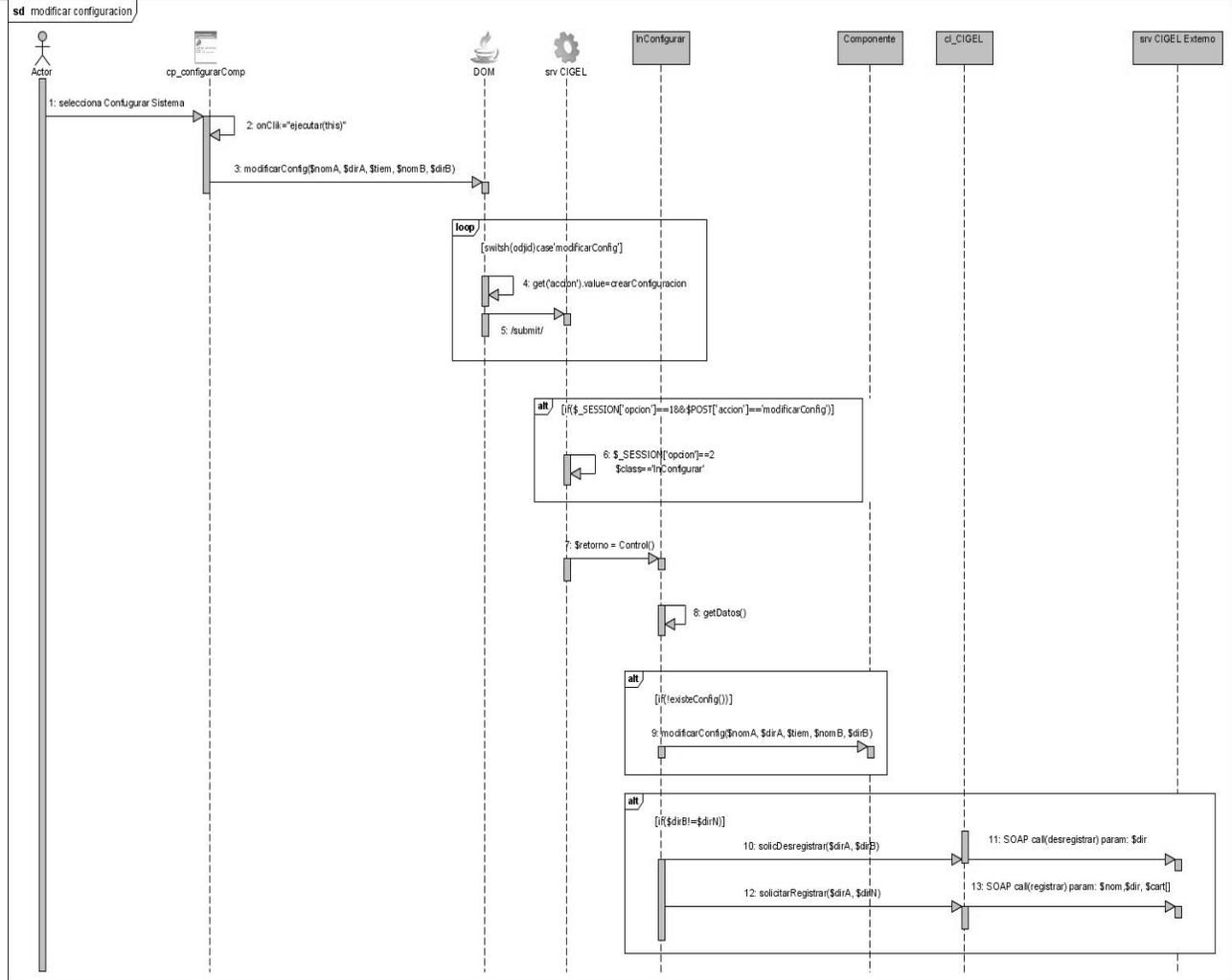


Figura 17. Diagrama de clases del diseño: *Modificar Configuración*

Diagrama de clases del diseño: Registrar Sistema

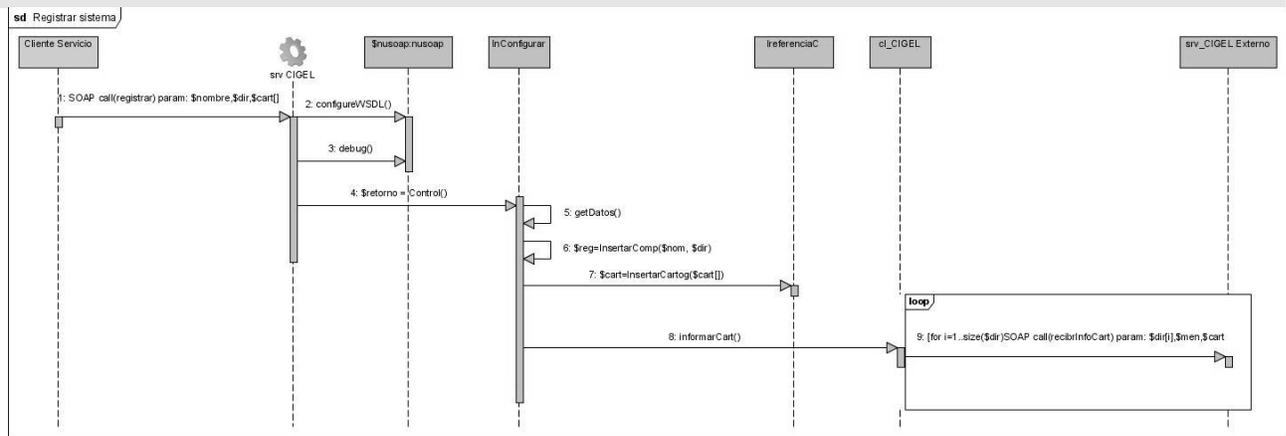


Figura 18. Diagrama de clases del diseño: Registrar Sistema

Diagrama de clases del diseño: Desregistrar Sistema

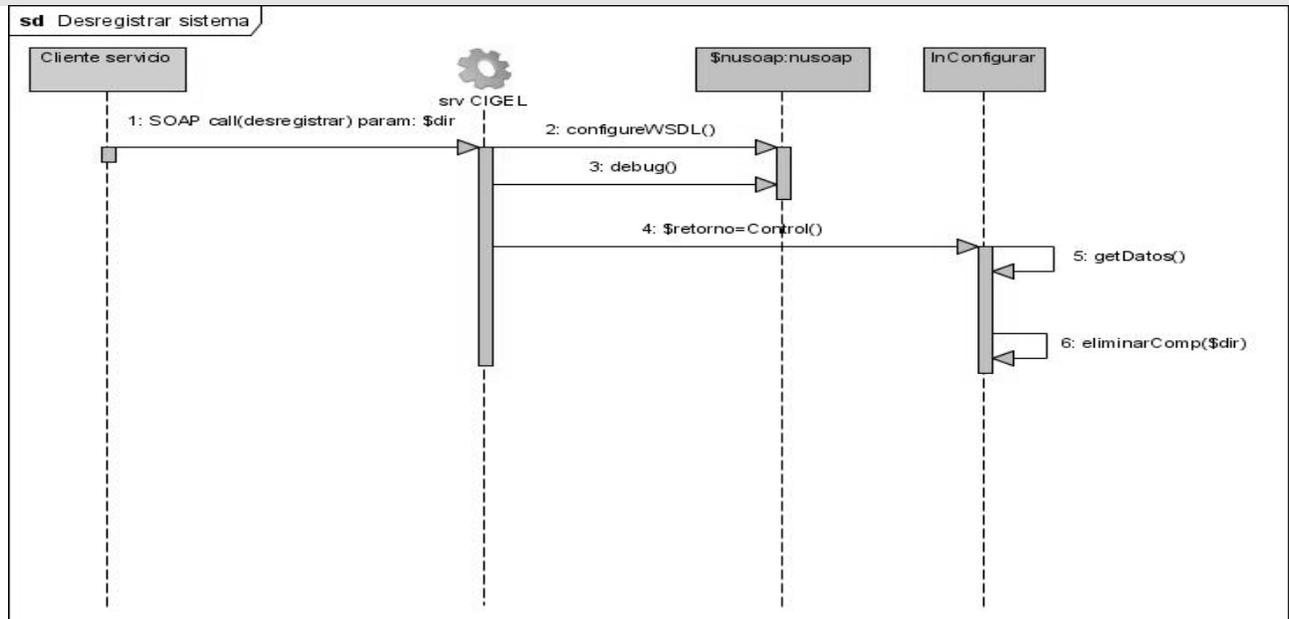


Figura 19. Diagrama de clases del diseño: Desregistrar Sistema

Diagrama de clases del diseño: Agregar nueva referencia cartográfica

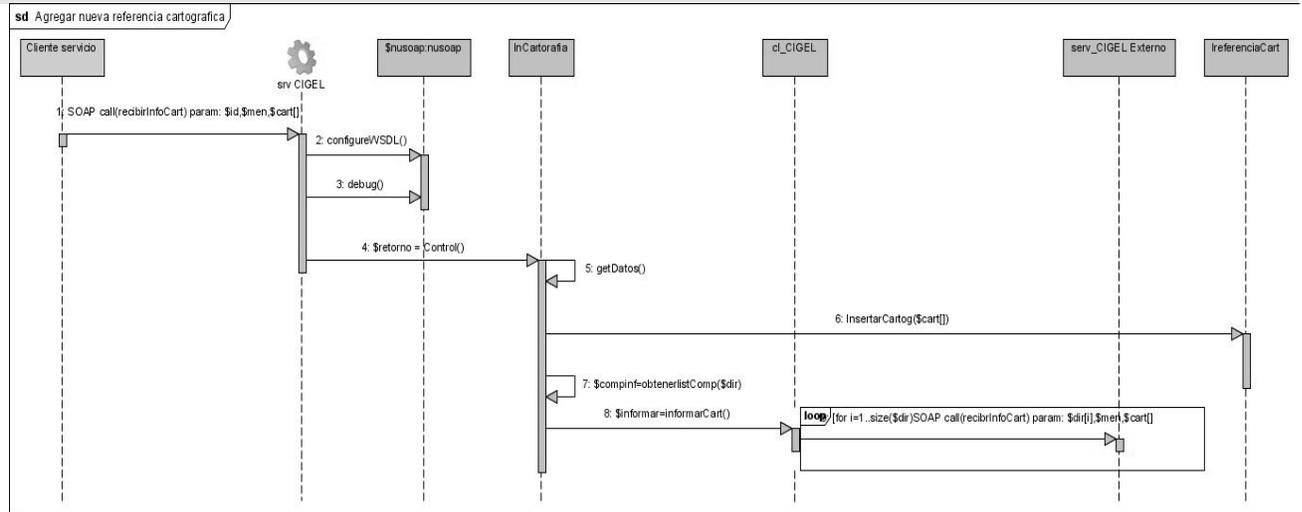


Figura 20. Diagrama de clases del diseño: Agregar nueva referencia cartográfica

Diagrama de clases del diseño: Modificar referencia cartográfica

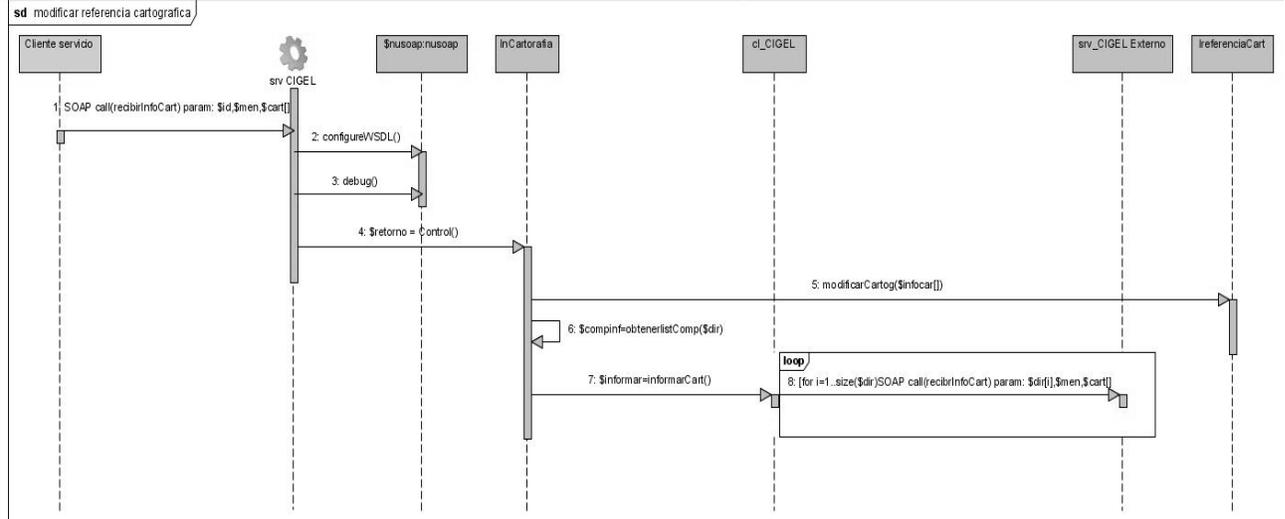


Figura 21. Diagrama de clases del diseño: Modificar referencia cartográfica

Diagrama de clases del diseño: *Eliminar capa pública*

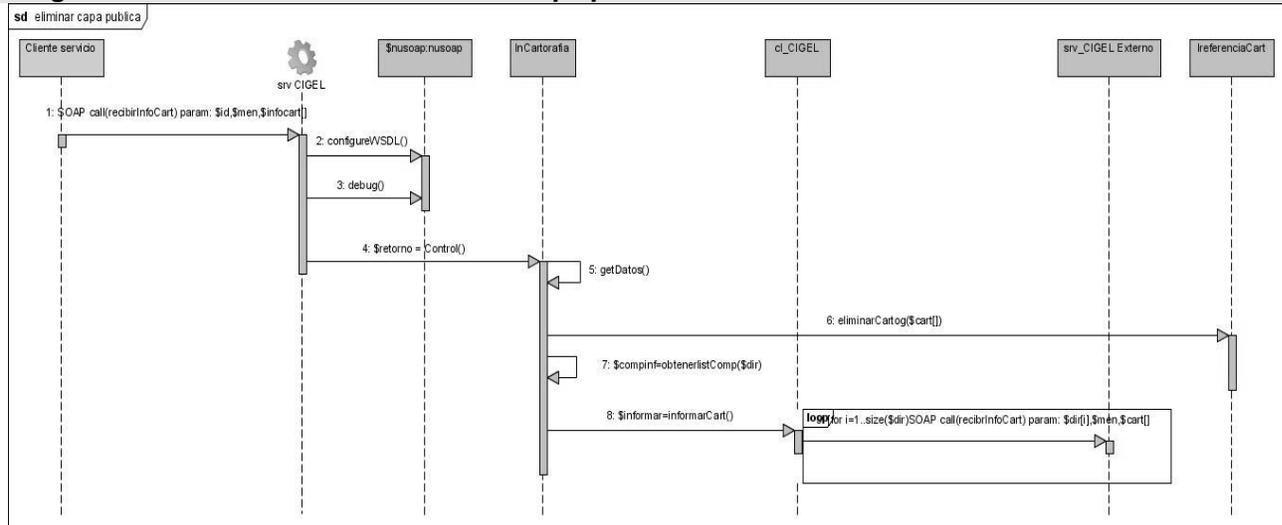


Figura 22. Diagrama de clases del diseño: *Eliminar capa pública*

Diagrama de clases del diseño: *Entregar información solicitada*

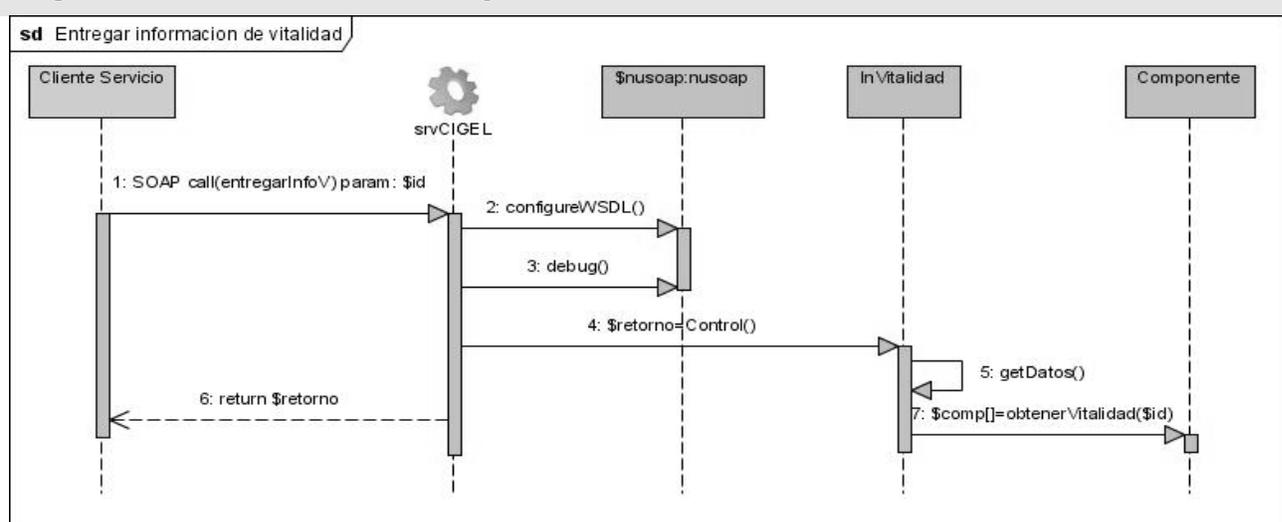


Figura 23. Diagrama de clases del diseño: *Entregar información solicitada*

Diagrama de clases del diseño: *Monitorear componente*

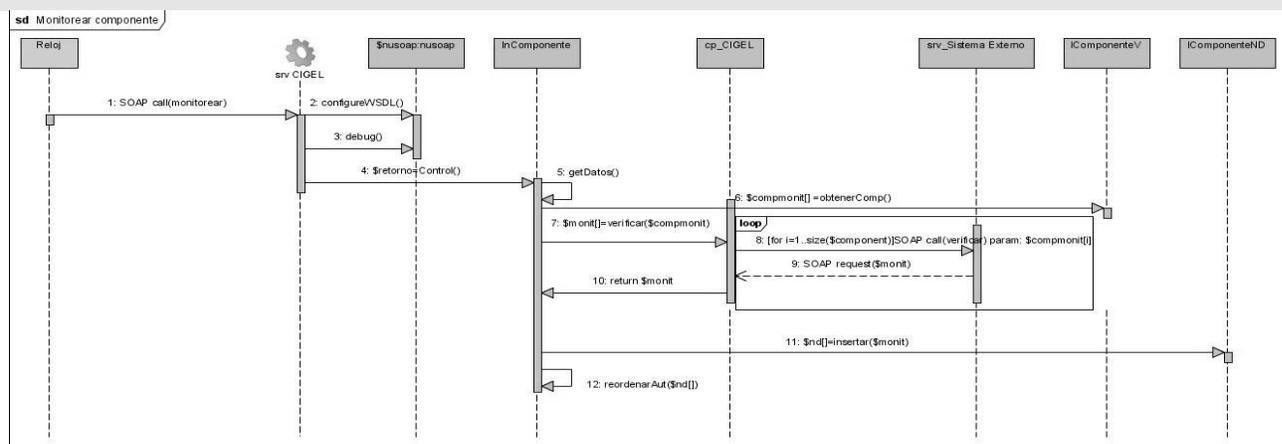


Figura 24. Diagrama de clases del diseño: *Monitorear componente*

Diagrama de clases del diseño: *Reordenar Automático*

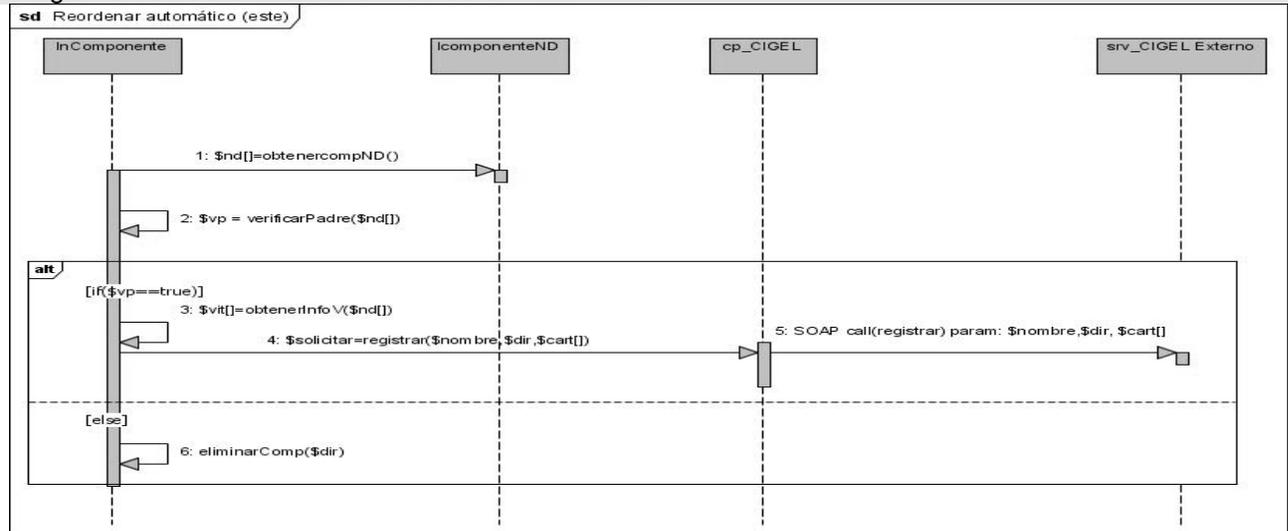


Figura 25. Diagrama de clases del diseño: *Reordenar Automático*

3.5. Principios de diseño

Descripción de las clases utilizadas

Clases	Descripción
[Plantillas]	Son las clases que contienen en su estructura todo el contenido estático de una página Web. Estas serán empleadas para generar las páginas clientes.
[Lógica del negocio]	Son las clases con prefijo In y encapsulan en sus métodos todas las operaciones específicas para cada caso de uso del sistema.
[Global]	Fichero con código PHP que contiene instancias y funciones que serán empleadas por todas las páginas servidoras.
[NuSOAP]	Es un kit de herramientas (ToolKit) para desarrollar Web Services bajo el lenguaje PHP, licenciada bajo LGPL. Está compuesto por una serie de clases que nos harán mucho más fácil el desarrollo de Web Services. NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1
[CL CIGEL]	Esta clase es la parte cliente del servicio Web, que se comunica directamente con la parte servidora y permite el acceso a los servicios disponibles.
[srv Servicio]	Es la clase base del servicio, que implementa todas las funcionalidades del lado del servidor. Publica todas las funciones disponibles como servicio y brinda la posibilidad al usuario de saber cuáles de las funciones están disponibles y sus características mediante la lectura de un fichero generado en XML (WSDL).

3.6. Mecanismos de diseño

Estos mecanismos se identifican a partir de comportamientos comunes; con su elaboración, teniendo en cuenta las tecnologías de diseño e implementación disponibles, se modela un conocimiento que ayuda al desarrollo de la futura aplicación.

Con el propósito de hacer el diseño de un sistema más objetivo, claro y eficiente para la futura implementación proponemos la utilización de los mecanismos de diseño para la seguridad del servicio de directorio que permita conocer la información geoespacial en línea en la institución.

Seguridad

Este ERP cuenta con un conjunto de subsistemas dentro de él se encuentra el destinado a la seguridad del proyecto. Este subsistema, que se comunica con una interfaz con la clase de control de Acceso, recibe las autenticaciones de los usuarios al ingresar al sistema, y chequea su existencia dentro del mismo, así como los privilegios que le han sido asignados, de ahí se derivan el conjunto de acciones que puede ejecutar dentro del módulo al que ha accedido y al que lo han redireccionado después haber confirmado la autenticidad de los datos entrados por el usuario.

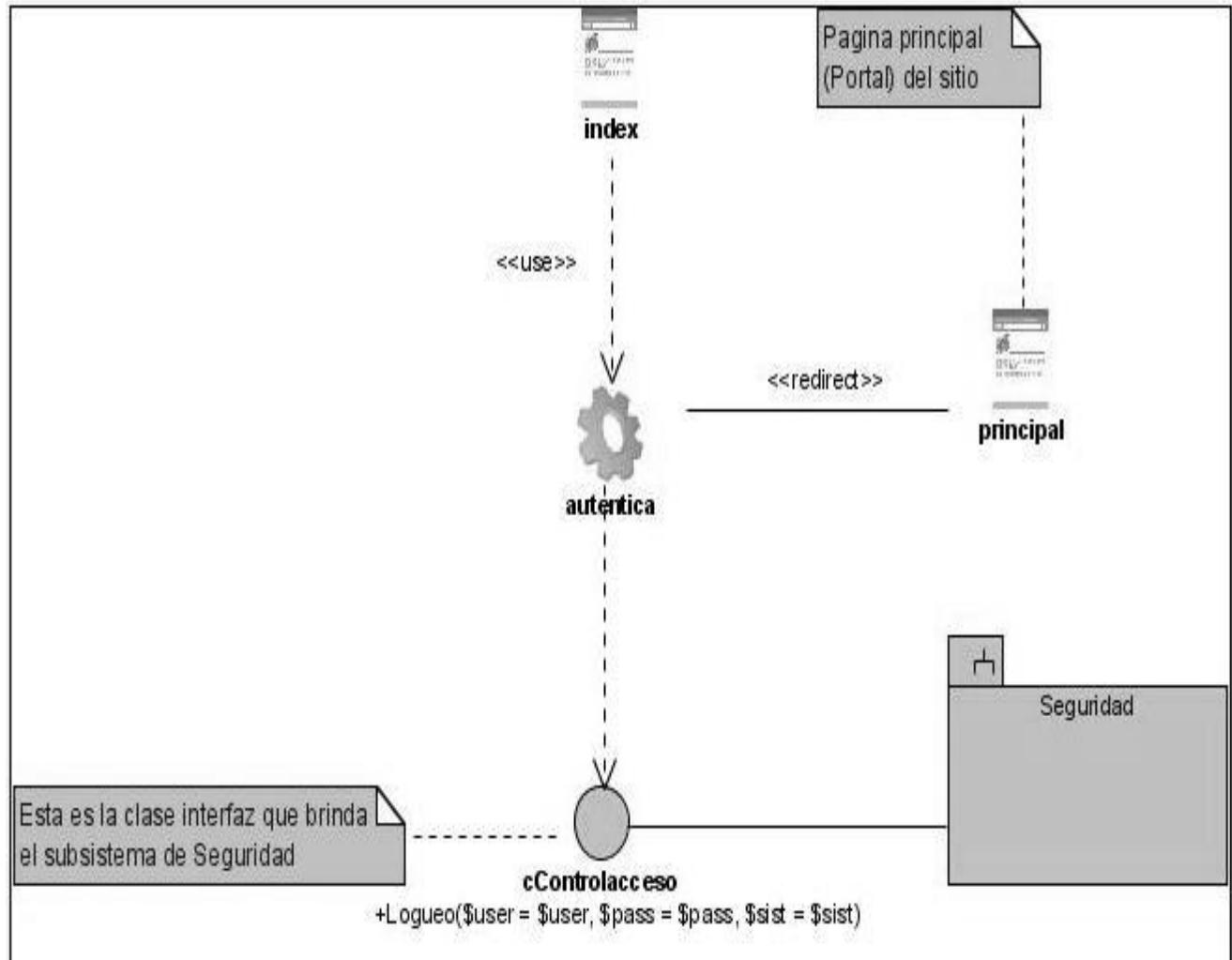


Figura 26. Mecanismo de diseño de seguridad

3.7. Conclusiones

En este capítulo se representaron las diferentes clases del análisis y diseño, en los cuales se han recopilado todos los elementos necesarios para la realización del proceso de realización de la aplicación.

IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

En este capítulo se representan los diagramas del despliegue que nos muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo, así como una mejor comprensión de la arquitectura del software y la arquitectura del hardware. También se representa el modelo de componentes para saber la distribución de los componentes necesarios para el desarrollo de la aplicación.

4.2. Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

Diagrama de despliegue.

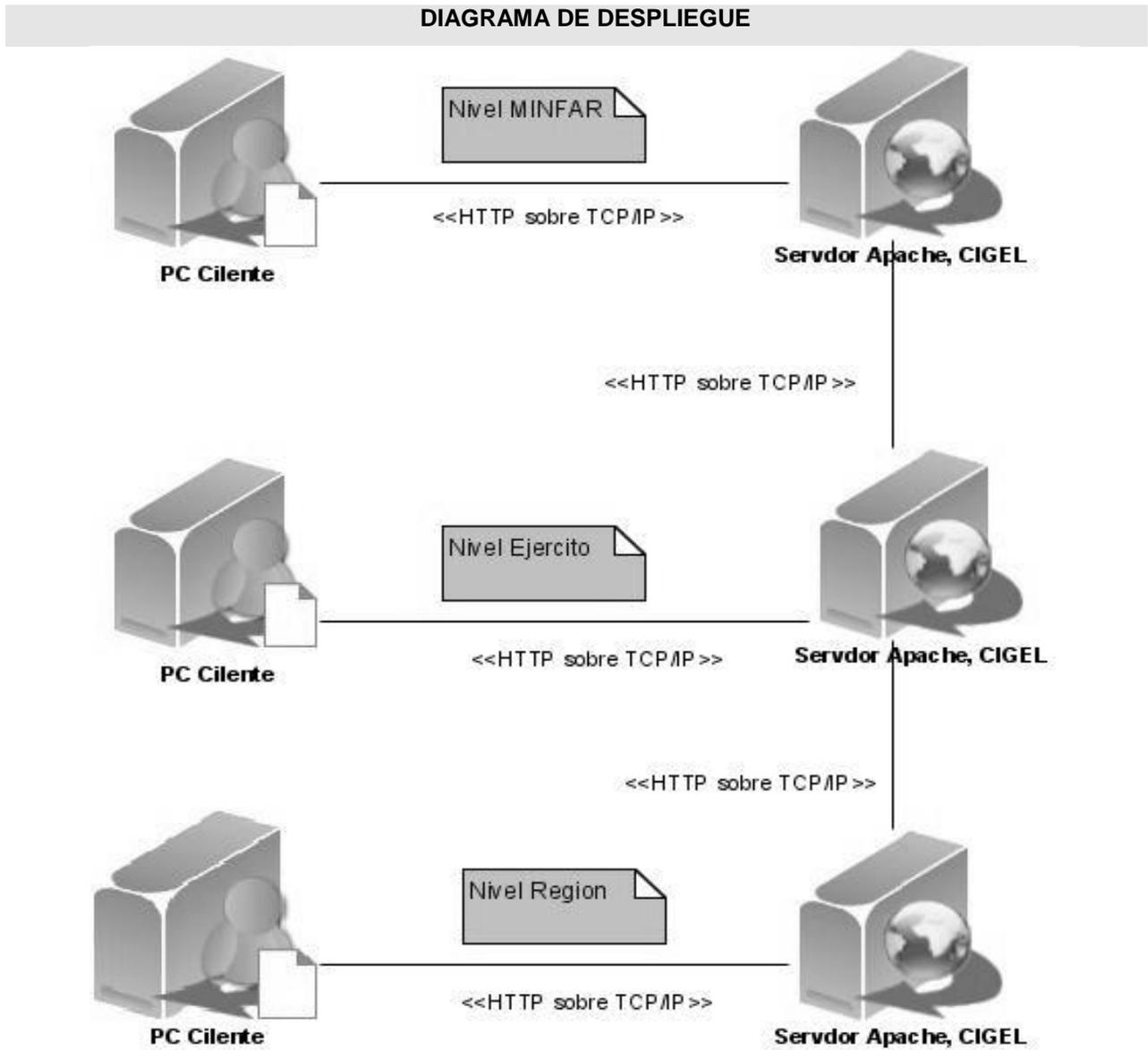


Figura 27. Diagrama de despliegue

4.3. Modelo de Componentes.

Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfases que estos soporten. Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

4.3.1. Componente

Representa una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces.

Diagrama de componentes.

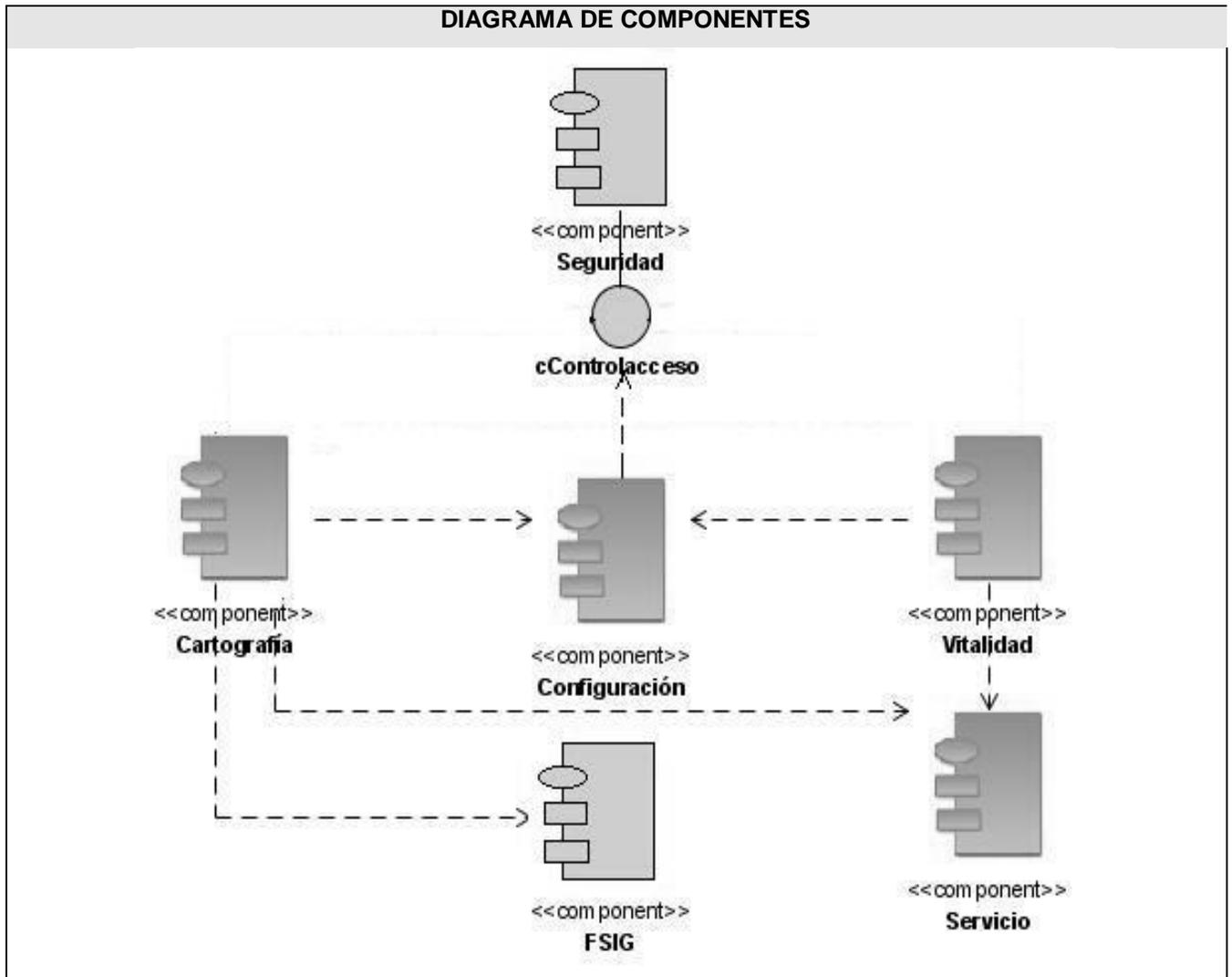


Figura 28. Diagrama de componentes general

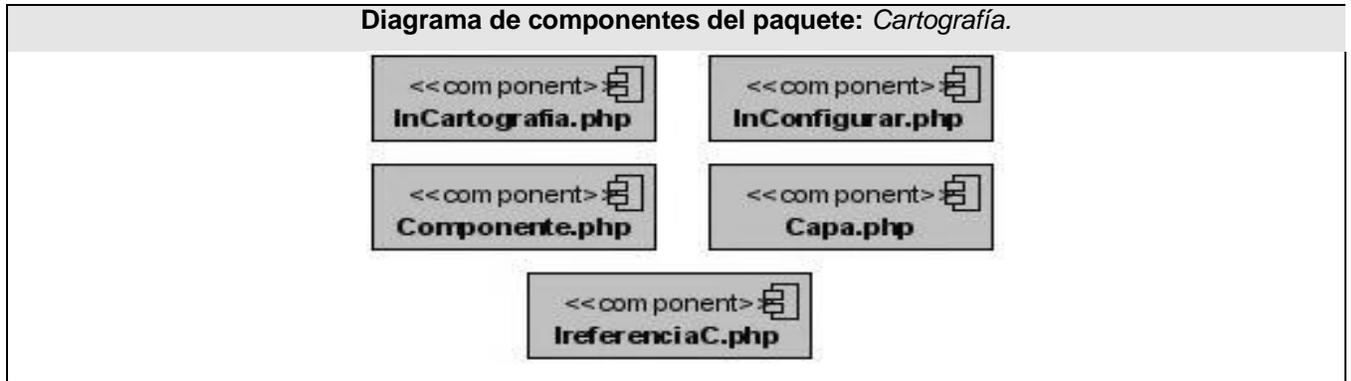


Figura 28. Diagrama de componentes del paquete: *Cartografía*

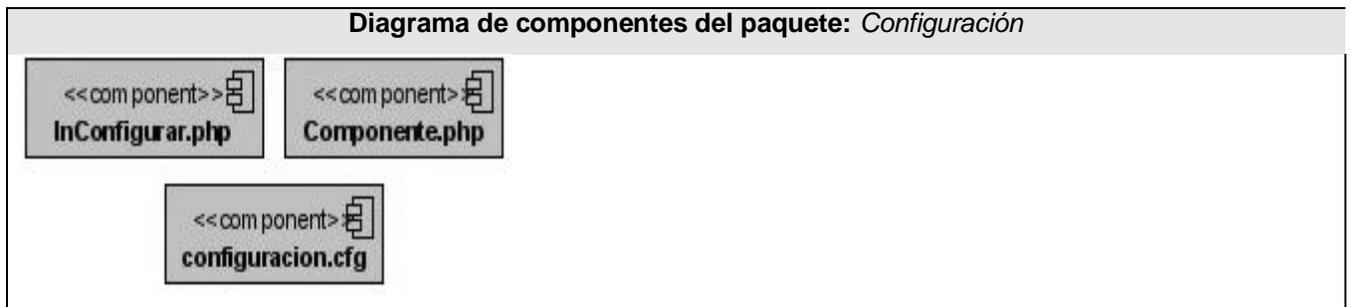


Figura 29. Diagrama de componentes del paquete: *Configuración*

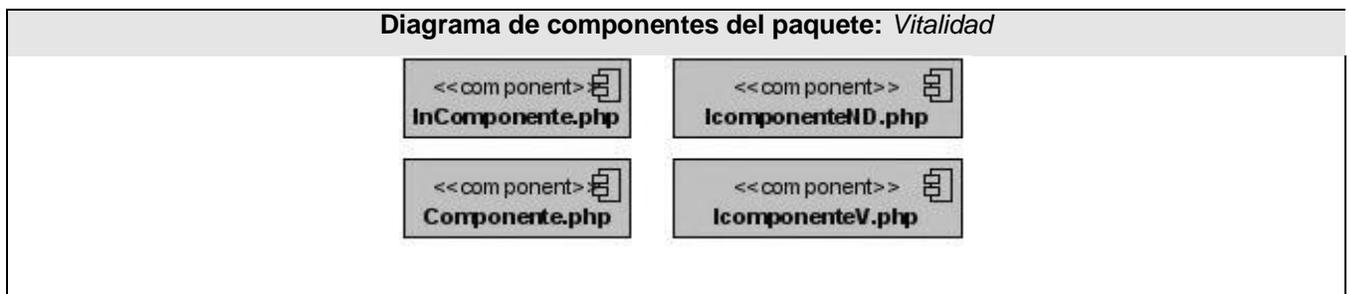


Figura 30. Diagrama de componentes del paquete: *Vitalidad*

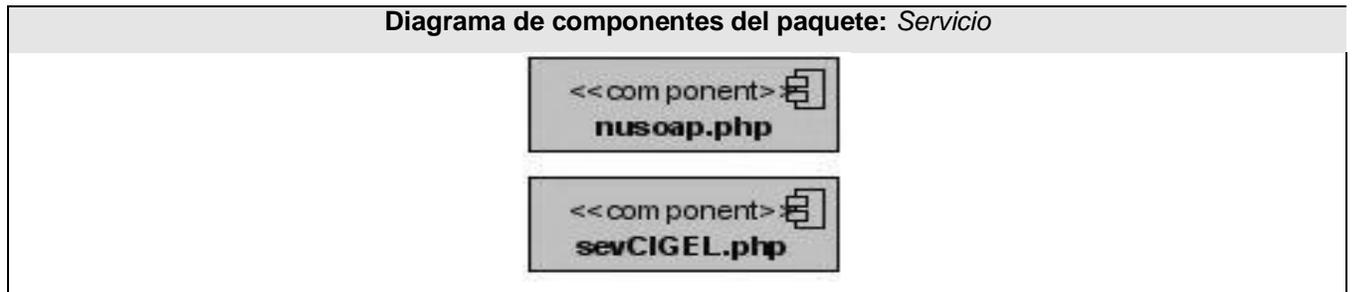


Figura 31. Diagrama de componentes del paquete: *Servicio*

4.4. Conclusiones

En este capítulo se explicó cómo estará estructurada la aplicación físicamente, mediante los modelos de despliegue y de componentes.

CONCLUSIONES GENERALES

Con la modelación del subsistema de control para la representación de la información geográfica en línea facilitará la creación de un servicio de directorio para conocer la información geoespacial que se encuentra disponible en la red y la forma de acceder a las mismas. Lo anterior es de suma importancia para el proceso de toma de decisiones de la entidad. Además se garantiza la accesibilidad desde cualquier parte territorio con conexión a la red.

El desarrollo de esta aplicación constituye un aporte práctico muy importante, debido a que es una novedad tecnológica que marca un proceso de avance en el desarrollo de software multiplataforma e interoperable dentro de la entidad.

RECOMENDACIONES

- Se recomienda el estudio de esta tesis para, perfeccionar sus funcionalidades.
- Se exhorta a que sea mejore la modelación de la aplicación en todo lo que se crea útil y necesario para el desarrollo del sistema.
- Se recomienda que se desarrolle la implementación del sistema para su uso por los usuarios finales.
- Que se cree un mecanismo para que el CIGEL gestione réplica de cartografía entre diferentes FSIG.

BIBLIOGRAFÍA

BIBLIOGRAFÍA CITADA

- [1] CORDÓN, Nuria. Los Sistemas de Información Geográfica se hacen populares en la empresa. Computerworld, 2007 [consultado en: mayo 2007] Disponible en: <http://www.idg.es/computerworld/articulo.asp?id=300536762>
- [2] JIMENEZ, José A.; AGUILERA, Ma. Jesús; MEROÑO, José E. ALTERNATIVAS DE SOFTWARE LIBRE A LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA COMERCIALES. [consultado en: Mayo 2007] Disponible en: <http://www.cartesia.org/geodoc/ingegraf2005/gis10.pdf>
- [3] BATISTA, José Luis. APLICACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA EN CUBA. [consultado en: enero 2007] Disponible en: http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051
- [4] GUINEA, Alejandro; JORRÍN, Sergio. Arquitectura SOA para la integración entre software libre y software propietario en entornos mixtos. [consultado en: abril 2007] Disponible en: <http://www.sigte.udg.es/JornadasSIGLibre/comun/1pdf/13.pdf>
- [5] VAN DER HENST, Christian. ¿Qué es el PHP? [consultado en: abril 2007] Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro/>
- [6] ALFARO, Ricardo. ¿Por qué elegir PHP? [consultado en: mayo 2007] Disponible en: <http://www.mmug.cl/articulos.php?id=283>
- [7] ANÓNIMO. WSDL para la documentación de Servicios Web. [consultado en: abril 2007] Disponible en: <http://www.desarrolloweb.com/articulos/1581.php>
- [8] ANÓNIMO. SOAP (Simple Object Access Protocol) [consultado en: abril 2007] Disponible en: <http://www.desarrolloweb.com/articulos/1557.php>
- [9] ANÓNIMO. XML: el lenguaje de los Servicios Web. [consultado en: abril 2007] Disponible en: <http://www.desarrolloweb.com/articulos/1574.php>
- [10] ANÓNIMO. UDDI (Universal Description Discovery and Integration). [consultado en: abril 2007] Disponible en: <http://www.desarrolloweb.com/articulos/1589.php>
- [11] MUÑOZ, O. Servicios Web. [consultado en: abril 2007] Disponible en: <http://www.it.uc3.es/mario>
- [12] ANÓNIMO. Cursos de empresa Dremweaver. [consultado en: abril 2007] Disponible en: <http://www.aulafacil.com/AulaDream/Dream/Lecc-01.htm>

BIBLIOGRAFÍA CONSULTADA

- Jacobson, Ivar; Booch, Grady; Rumbaugh James. El proceso unificado de software. Volumen I. La Habana, 2004. Editorial Félix Varela
- Jacobson, Ivar; Booch, Grady; Rumbaugh James. El proceso unificado de software. Volumen II. La Habana, 2004. Editorial Félix Varela
- Pressman, Rogers. Ingeniería del software. Parte 1. La Habana, 2005. Editorial Félix Varela.
- Pressman, Rogers. Ingeniería del software. Parte 2. La Habana, 2005. Editorial Félix Varela.
- Carmona, Alvaro de J., Monsalve, Jhon Jairo. [consultado en: enero 2007] Sistemas de Información Geográficos. Disponible en: <http://www.monografias.com/trabajos/gis/gis.shtml>

BIBLIOGRAFIA CONSULTADA

- [Álvarez] - Álvarez, Sofía, Hernández Anaisa. Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML. La Habana, 2000
- [ASP] - ¿Qué es eso de asp? <http://www.pobladores.com/territorios/informatica/asp>. (17/01/2002)
- [CocomoII] - Boehm, B. et al 2000 "Software Cost Estimation with COCOMO II"
- [Dealer] - Dealer World, la revista de referencia para el canal. <http://www.idg.es/dealer> (15/01/2002)
- [e-Contact] - Las Mesas de Ayuda. Revista e-Contact. Julio-Agosto. 2002.
<http://www.imt.com.mx/revista/numero19>. (15/01/2002)
- [Farell] - Tourniaire, Françoise y Farell, Richard. The Art of Software Support. Prentice Hall. 1998.
- [Febles-a] - MSc. Febles Estrada, Aylin. Presentación en Power Point. Clase de Soporte de Software. Asignatura ADS. 5to Año. Curso 2001-2002.
- [Febles-b] - MSc. Febles Estrada, Aylin. "Case Corporativo para el proceso de control de cambios" Tesis presentada en opción al título de Master en Informática Aplicada, Ciudad de la Habana, 2001.
- [Hernández] - Hernández, Yanko e Banderas, Ideal. "Case para la planificación y control de configuración de software", Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2001.
- [Herrera] - Herrera J., Lizka Johany. "Ingeniería de Requerimientos – Ingeniería de Software", <http://www.monografias.com/trabajos6/resof> (04/01/2002)
- [Huidobro] - Huidobro Moya, José Manuel. Redes y Servicios de Telecomunicaciones. Segunda Edición.
- [Internet] - Colectivo de autores. Secretos de Internet. Editorial Academia, La Habana 2001.
- [Jones] - JONES, Capers. Programming Language Table Release 8.2, March 1996.
<http://www.spr.com/library> (17/01/2002)
- [JS-a] - Guía de JavaScript <http://developer.netscape.com/docs/manuals/communicator/jsguide4>. (17/01/2002)
- [JS-b] - Manual de JavaScript <http://www.redestb.es/soporte/aula/jScript>. (17/01/2002)
- [JS-c] - Sitio de Macromedia <http://www.macromedia.cl/products>. (17/01/2002)
- [Matos] - Matos, Rosa María. Introducción al trabajo con Base de Datos. Asignatura de Sistemas de Gestión de Base de Datos.
- [Morales] - Morales Moliner, Alex. Conceptos básicos de ASP
http://www.programacion.com/asp/tutoriales/asp_basics. (17/01/2002)
- [PHP] - Introducción a php. <http://www.ciberteca.net/webmaster/php>. (17/01/2002)
- [Pressman] - Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Cuarta Edición
- [Rational] - "Lo nuevo de Rational Rose 2000" . Rational Corporation, 2000.
<http://www.abists.com.mf/Fabs/Rational/notasTK>. (17/01/2002)
- [Rodríguez] - Rodríguez, Daniel y Bravo, Joaquín. Tutorial de HTML <http://html.programacion.net>. (17/01/2002)
- [RUP] - Rational Unified Process, Rational Software Corporation, "Rational Unified Process", Version 2001A.04.00, Copyright 1987-2001.
- [Salas] - Salas, Juan Francisco. La satisfacción del cliente: una estrategia empresarial. Febrero 2001.
<http://www.cinterac.com.ar>. (15/01/2002)
- [Sánchez] - Sánchez Cavazos, Yolanda del Carmen. Satisfacción del cliente e inteligencia corporativa. Universidad de Monterrey. <http://www.mex-i-co.com/scip>. (15/01/2002)
- [Satisfacción] - Satisfacción del cliente. <http://www.ceocant.es/documentosvarios>. (15/01/2002)
- [UML-a] - Tutorial de UML <http://www.dcc.uchile.cl/~psalinas/uml>. (17/01/2002)

GLOSARIO DE TÉRMINOS

C

- **Cartografía:** Es una disciplina que integra ciencia, técnica y arte, que trata de la representación de la Tierra sobre un mapa o representación cartográfica.

D

- **Datos Alfanuméricos:** Son descripciones de las características de las entidades gráficas. Generalmente son almacenados en formatos convencionales para este tipo de información.
- **DCOM:** (Distributed Component Object Model) es una tecnología propietaria de Windows para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí.

I

- **Internet:** Red de computadoras alrededor de todo el mundo que comparten información unas con otras por medio de páginas o sitios.
- **Interoperabilidad:** Condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos

N

- **Nodo:** Elementos que forman parte de la red (FSIG o CIGEL)

P

- **PHP** (Personal Home Page) Es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor.

R

- **RUP (Rational Unified Process):** Proceso Unificado de desarrollo.

S

- **SIG:** Sistemas de Información Geográfica (**GIS** por sus siglas en inglés). Sistema de hardware, software y procedimientos diseñados para soportar la captura, administración, manipulación, análisis, modelamiento y graficación de datos u objetos referenciados espacialmente, para resolver problemas complejos de planeación y administración. Es un sistema de computador capaz de mantener y usar datos con localizaciones exactas en una superficie terrestre. Es una herramienta de análisis de información.
- **SOA:** Arquitectura Orientada a Servicios
- **SOAP (Simple Object Access Protocol):** Protocolo Simple de Acceso a datos. Creado por Microsoft para programar servicios en la Web que se basan en XML, con independencia del lenguaje de programación y del Sistema Operativo.
- **SWL:** Software Libre, aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan.
- **SWP:** Software propietario, software privativo. Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible o el acceso a este se encuentra restringido.

U

- **UML:** Lenguaje Unificado de Modelado. Es el lenguaje de modelado de sistema de software más conocido en la actualidad.
- **UDDI:** Servidor de Descripción Universal, Descubrimiento e Integración.

W

- **WSDL (Web Service Description Language):** lenguaje de descripción de servicios Web.
- **WMS (Web Map Services):** El servicio Web Map Service (WMS) definido por el OGC (Open Geospatial Consortium) produce mapas de datos espaciales referidos de forma dinámica a partir de información geográfica.

X

- **XML (Extensible Markup Language):** lenguaje de anotación extensible.