# Universidad de las Ciencias Informáticas

# FACULTAD 6


**Título: "Módulo de monitoreo para la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG".**

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

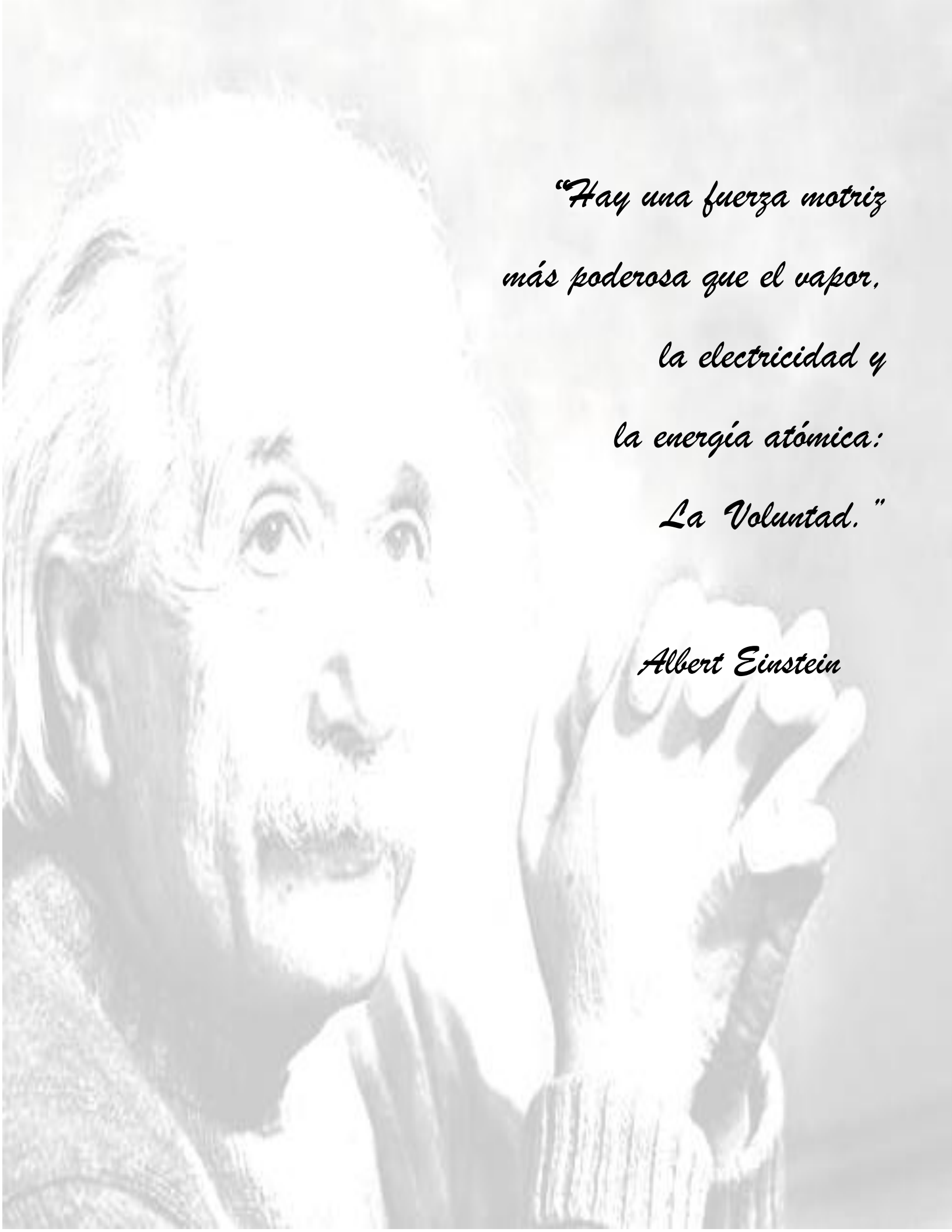
**Autora:** Celia María Guerra Fernández

**Tutores:** Ing. Yenier Jiménez Morales

Ing. Leiber Fornaris Ramírez

**Asesor:** Ing. Marcel Mesa Martínez

**La Habana, Junio del 2013**
**"Año 55 de la Revolución"**



*"Hay una fuerza motriz*

*más poderosa que el vapor,*

*la electricidad y*

*la energía atómica:*

*La Voluntad."*

*Albert Einstein*

# Declaración de autoría

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

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

**Celia María Guerra Fernández**

_____
Firma del Autor

**Ing. Yenier Jiménez Morales**

_____
Firma del tutor

**Ing. Leiber Fornaris Ramírez**

_____
Firma del tutor

# Datos de contacto

**Tutor:** Ing. Yenier Jiménez Morales

**Formación Académica:** Ingeniero en Ciencias Informáticas (Julio/2010).

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** yjmorales@uci.cu

**Tutor:** Ing. Leiber Fornaris Ramírez

**Formación Académica:** Ingeniero en Ciencias Informáticas (Julio/2011).

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** lfornaris@uci.cu

**Asesor:** Ing. Marcel Mesa Martínez

**Formación Académica:** Ingeniero en Ciencias Informáticas (Julio/2008).

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** mmesa@uci.cu

# Agradecimientos

*Ante todo a mis padres Milagros y Eusebio por ser los mejores del mundo, por siempre quererme y apoyarme en todas mis decisiones sin importar lo que fuese. A mis dos mejores amigas de toda la vida Ani y Ela mis hermanas de la vida. A mi novio René por ser mi bastón de apoyo y confidente, porque cuando empezaba a creer que no podía ya me estabas empujando para que siguiera adelante. A mis hermanos, mi sobrino, tíos, primos y en especial a mi abuelito Manolo, es decir a toda mi familia que aunque estén lejos siempre se preocupan por mí.*

*Quisiera agradecer especialmente mis amigos y a todos aquellos que en mi trayecto por la universidad fueron mi familia aquí en la Habana y me brindaron un pedacito de su hogar, gracias a ellos no sentí todo este tiempo el gran vacío que significaba estar lejos de mis padres y amigos de toda la vida.*

*...Celia*

# Dedicatoria

*A mis padres Milagros y Eusebio*
*por su amor y dedicación.*

# Resumen

Los avances en las Tecnologías de la Información y las Comunicaciones han propiciado el surgimiento de herramientas como los sistemas de información geográfica que son utilizadas para representar y analizar información geográfica relacionada a empresas e instituciones, además de apoyar el análisis de toma de deciciones. Perteneciente al Centro de Geoinformática y Señales Digitales de la facultad 6 en la Universidad de las Ciencias Informáticas, el proyecto Aplicativos_SIG tiene como base para el desarrollo de estos sistemas a la plataforma GeneSIG.

La presente investigación surge a partir de la necesidad de Aplicativos_SIG, de contar con un módulo para la plataforma GeneSIG que permita la visualización en tiempo real, de la información geográfica gestionada en los sistemas desarrollados por el proyecto. Su desarrollo propicia que toda la información gestionada sea actualizada en tiempo real, siendo esto muy beneficioso para lograr un mejoramiento en los procesos de análisis de toma de decisiones en las diferentes empresas o instituciones para las que se desarrollan sistemas de información geográfica. Para la construcción de esta solución se utilizó como guía de desarrollo a la metodología AUP, se definió como protocolo de comunicación a XMPP y en especial sus extensiones Publicación-Suscripción y BOSH. Se emplean los leguajes de programación PHP, JavaScript y las bibliotecas Extjs y Strophejs.

**Palabras clave**

BOSH, información geográfica, módulo, monitorear, Publicación-Suscripción, tiempo real, XMPP.

# Índice

# Índice de figuras

# Índice de tablas

## Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) han tenido un gran auge a nivel mundial debido a los innumerables avances que se han logrado en múltiples sectores de la sociedad. Las TIC han provocado cambios en la manera de realizar los negocios gracias la disposición de nuevos productos y servicios, y un incremento en los niveles de organización. Estas tecnologías marcaron el surgimiento de herramientas como los Sistemas de Información Geográfica (SIG) que integran "tecnología informática, personas e información geográfica[1], y cuya principal función es capturar, analizar, almacenar, editar y representar datos georreferenciados[2]". (Olaya, 2010) Dichos sistemas son utilizados por empresas o entidades para manejar información georreferenciada acerca de innumerables esferas científicas, tecnológicas, socio-económicas, medio-ambientales y de otros tipos, que facilitan resolver problemas complejos de planificación y gestión de datos.

A principios de los años 90 del siglo XX es creado el primer Sistema de Información Geográfica de Cuba, lo que marcó el inicio de un amplio desarrollo de SIG, con vista a extender la informatización del país en todas sus esferas. La Universidad de las Ciencias Informáticas (UCI), ha adquirido experiencia en el desarrollo de *software* y su estructura organizacional a través de centros de desarrollos ha contribuido a extender el progreso de estos a las distintas esferas de la sociedad. El Centro de Desarrollo Geoinformática y Señales Digitales tiene como misión desarrollar productos, servicios y soluciones informáticas en el campo del procesamiento de Señales Digitales y la Geoinformática. Perteneciente a este centro, el proyecto Aplicativos_SIG es una Línea de Productos de *Software* (LPS), que se especializa en realizar la personalización[3] de SIG para determinadas empresas o instituciones. Estas personalizaciones tienen como principal base para su desarrollo a la plataforma GeneSIG, la cual "es un producto libre y soberano que surge de la unión de los equipos de desarrollo de *software* Unidad

[1] Información que describe a entidades y fenómenos geográficos, identifica la localización geográfica y las características de los elementos naturales o construidos.
[2] Información a la cual puede asignársele una posición geográfica.
[3] Adaptar las funcionalidades de un SIG a los requerimientos de una entidad o empresa que solicite el desarrollo de estos sistemas.

de Compatibilización e Integración para la Defensa (UCID, MINFAR), GeoCuba[4] y la UCI" (Pantoja, 2010) y se especializa en la creación de SIG encaminados a realizar la representación y análisis de información geográfica.

Las personalizaciones desarrolladas en la LPS Aplicativos_SIG se emplean para representar y analizar información geográfica relacionada a la empresa o institución que necesite un sistema de este tipo. La actualización de dicha información representada, tras la ocurrencia de cambios generados por los usuarios es un factor importante para el proceso de análisis de toma de decisiones, ya que de ello depende que la información con la que se trabaje no sea obsoleta en correspondencia con el momento en que se analice. En este momento la actualización de la información no se realiza de manera correcta en concordancia con lo que se necesita y para explicar la situación planteada se detalla el siguiente ejemplo:

- Dos usuarios (*A* y *B*) están usando el SIG en lugares distintos al mismo tiempo. Si el usuario *A* modifica de alguna manera la información representada en él, el usuario *B* no podrá ver los cambios realizados hasta que no solicite manualmente la actualización de la interfaz de usuario en el navegador o que realice alguna acción que requiera su actualización. Esto ocurre de la misma manera si es el usuario *B* quien realiza las modificaciones.

Se desarrolló como estrategia para cambiar esta situación, la actualización frecuente de la interfaz de usuario del sistema automáticamente, en intervalos de tiempo definidos por el equipo de desarrollo, para mostrar las posibles modificaciones realizadas por los usuarios que trabajen simultáneamente en el SIG. Dicha estrategia resuelve el problema de cierta forma, pero no implica una solución adecuada porque se sobrecarga el sistema con búsquedas innecesarias si no se realizaron modificaciones.

Partiendo del análisis de la problemática planteada se define como **problema a resolver**: ¿Cómo visualizar en tiempo real las actualizaciones que se ejecutan relacionadas a la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG?

[4] Grupo empresarial cubano dedicado al desarrollo cartográfico.

# Introducción

Una vez determinado el problema de la investigación se hace necesario encontrar una solución, teniendo como **objeto de estudio** el proceso de visualización de información en tiempo real. Enmarcando como **campo de acción** la visualización en tiempo real de la información geográfica.

Como **objetivo general** de la investigación se pretende desarrollar un módulo para la plataforma GeneSIG que permita monitorear[5] la gestión de información geográfica en sus personalizaciones y así visualizar en tiempo real las modificaciones realizadas. De manera que se determina como **idea a defender:** El desarrollo de un módulo que posibilite el monitoreo de la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG permitirá visualizar en tiempo real las actualizaciones realizadas por los usuarios.

Con el fin de garantizar el cumplimiento del objetivo trazado se plantean las siguientes **tareas de la investigación**:

1. Caracterizar el flujo de información en las personalizaciones de la Plataforma GeneSIG.

2. Caracterizar el proceso de comunicación en tiempo real y los protocolos que lo posibilitan.

3. Fundamentar el uso de las tecnologías y herramientas a utilizar para el desarrollo del módulo que resuelve al problema planteado.

4. Elaboración de la documentación técnica asociada a la metodología de desarrollo seleccionada. Implementar los casos de uso definidos a partir de los requisitos funcionales.

5. Construcción del Módulo de monitoreo para la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG.

6. Validación del funcionamiento del módulo haciendo uso de pruebas ingenieriles.

Durante la evolución de la investigación se utilizaron diferentes métodos científicos con el objetivo de obtener variada información acerca del desarrollo del sistema, así como sintetizar dicha información

[5] Observar la gestión de la información geográfica, capturar los datos de las modificaciones realizadas y transmitirlos para que sean visualizados.

reconociendo los elementos más importantes para el desarrollo de la misma, los métodos anteriormente mencionados son:

**Métodos teóricos:**

**-Analítico-sintético:** Se utilizó en la consulta de diversas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Esto favorece el análisis de los elementos más importantes que aportan a la investigación.

**-Modelación:** Permitió crear abstracciones que explican la realidad, por ejemplo: todos los modelos y diagramas realizados.

**Métodos empíricos:**

**-Entrevista:** Se realizó con el fin de conocer las especificidades sobre las necesidades del equipo de desarrollo del proyecto LPS Aplicativos_SIG, así como los datos que podrían brindar para lograr el desarrollo del módulo que se propone. De un total de 12 profesores y 25 estudiantes pertenecientes al proyecto, obteniendo a una población[6] de 37 personas, se selecciona una muestra[7] de 3 profesores que incluye al líder del proyecto, el arquitecto y la analista principal, para aplicar la entrevista. Para seleccionar la muestra se utiliza la técnica de muestreo no probabilístico[8], específicamente el muestreo intencional[9], que permite elegir los integrantes de la muestra y permite escoger los elementos que son representativos y que pueden brindar mayor información. Las entrevistas realizadas fueron de carácter no formales, por tal motivo no se anexa la misma.

[6] La población es una agrupación de todos los individuos o elementos individuales de un tipo en particular.
[7] La muestra es una parte de la población que deberá representar adecuadamente las características que se desea analizar en el conjunto de estudio.
[8] Se seleccionan los elementos de la muestra de acuerdo a determinados criterios previamente establecidos, en este caso se tuvo en cuenta el rol que desempeñaban en la LPS y el nivel de experiencia en el desarrollo de SIG para lograr una mejor definición de las funcionalidades.
[9] Este tipo de muestra las selecciona una persona con experiencia y conocimiento amplio de la población en estudio, con el propósito de lograr una muestra lo más representativa posible de la población.

Con el desarrollo del trabajo de diploma se espera obtener los siguientes **resultados:**

1. Módulo de monitoreo para la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG.

2. La documentación asociada al proceso de desarrollo del módulo.

El presente trabajo está estructurado en cuatro capítulos:

**Capítulo 1. Fundamentación teórica.**

En este capítulo se realiza un análisis de la problemática presentada. Se realiza una fundamentación teórica de la investigación, en la cual se exponen, con el fin de lograr una mejor comprensión, los conceptos asociados al objeto de estudio Se expone una síntesis de algunas soluciones existentes para el problema identificado.

**Capítulo 2. Tendencias y tecnologías.**

En este capítulo se efectúa un estudio de las tecnologías y se argumentan las herramientas a utilizar en cuenta para el desarrollo del módulo que se desea implementar.

**Capítulo 3. Presentación de la solución propuesta.**

En este capítulo se realiza una presentación de la solución propuesta para ello se describe el dominio de la solución, identificando las funcionalidades con las que contará la solución. Se realiza una representación de la solución propuesta a desarrollar y se describe detalladamente.

**Capítulo 4. Construcción y prueba de la solución propuesta.**

En este capítulo se realiza la descripción del componente propuesto a partir de su diseño, y se muestran las relaciones entre sus componentes de implementación. Se realizan un conjunto de pruebas que permite al equipo de desarrollo verificar la correcta implementación de la solución propuesta.

# Capítulo 1. Fundamentación teórica

## 1.1. Introducción.

Este capítulo está dirigido a plantear todos los elementos teóricos que sirven de base al objeto de estudio y al objetivo de la investigación científica. Se identifican los conceptos asociados al dominio del problema y se realiza el estudio del arte donde se analizan los sistemas existentes que monitoreen información geográfica y que hagan uso de la mensajería instantánea. Además se argumentan las tecnologías y herramientas a utilizar en el desarrollo del módulo.

## 1.2. Conceptos asociados.

A continuación se enunciaran los conceptos que se consideran claves con el fin de lograr una mejor comprensión del presente trabajo:

**-Sistema:** "Es una colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico". (Graf, 2004) Un sistema es aquel que “puede reaccionar como un todo al recibir un estímulo dirigido a cualquiera de sus partes”. (AG, 1997)

**-Información geográfica:** “Se le denomina información geográfica al conjunto organizado de datos espaciales georreferenciados, que mediante símbolos y códigos genera el conocimiento acerca de las condiciones físico - ambientales, de los recursos naturales y de las obras de naturaleza entrópica del territorio nacional. La información es el resultado de un dato y una interpretación”. (Olaya, 2010).

**-Sistema de Información Geográfica (SIG):** ”Es una integración organizada de *hardware*, *software* y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión”. (Taylor, 1991)

**-Datos Geográficos:** “Entidades, espacio-temporales que describen o cuantifican la distribución, el estado y los vínculos de los distintos fenómenos naturales y sociales”. (IGAC, 1998)

**-Objetos geográficos:** “Los objetos geográficos son abstracciones de elementos del mundo real que están asociadas a una posición geográfica y temporal definida, respectivamente, en un sistema de referencia espacial y temporal”. (ICDE, 2013)

**-Monitoreo:** Enmarcado esta investigación monitorear es la observación del proceso de gestión de la información geográfica con el fin detectar las modificación realizada por los usuarios, capturar los datos referentes a estas modificaciones y transmitirlos para su posterior visualización.

**-*Plugin* (Complemento):** “Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Constituye un módulo de *hardware* o *software* que añade una característica o un servicio específico a un sistema más grande. En GeneSIG los *plugin* son paquetes modulares de archivos que se utilizan para llevar a cabo una acción especializada”. (Membrides, 2010)

**-Navegador web:** En el ámbito de la tecnología, un navegador es un programa informático que permite visualizar la información contenida en una página Web, ya sea alojada en Internet o en un servidor local. El navegador está en condiciones de interpretar los códigos de programación de la página y presentar el contenido en pantalla de modo que el usuario pueda interactuar con la información y navegar hacia otras páginas a través de enlaces. “Como un modelo cliente-servidor, el navegador es la ejecución del cliente en un equipo que en contacto con el servidor Web y solicita información. El servidor Web envía la información de vuelta al navegador Web que muestra los resultados en la computadora u otro dispositivo habilitado para Internet que soporte un navegador”. (IT Business Edge Network, 2013)

**-Comunicación asincrónica:** “Que no tiene un intervalo de tiempo constante entre cada evento. Característica de cualquier sistema de comunicación en el que el transmisor puede enviar datos sin previo aviso. El receptor debe estar preparado para aceptar datos en cualquier momento”. (Gutiérrez, 2009)

- ✓ **Independiente del lugar:** “La comunicación se produce entre comunicantes que pueden o no encontrarse físicamente ubicados en contextos distintos”. (Grupo 97 Catedra Unadista, 2011)

**-Comunicación sincrónica:** “En un sistema de comunicación, el transmisor debe coordinarse con el receptor antes del envío de datos”. (Gutiérrez, 2009) La comunicación sincrónica está caracterizada por ser:

- ✓ **Temporalmente dependiente:** “Para que este tipo de comunicación tenga lugar, es necesario que los comunicantes coincidan en un mismo tiempo”. (Grupo 97 Catedra Unadista, 2011)

***-Polling*** **(Encuestado):** “Se define como hacer constantes peticiones de datos desde un dispositivo a otro”. (IT Business Edge Network, 2013) Es una operación de consulta constante, generalmente hacia un dispositivo de *hardware*, para crear una actividad sincrónica sin el uso de interrupciones, aunque también puede suceder lo mismo para recursos de *software*, es generalmente generada en el cliente.

***-Long Polling:*** “No inmediatamente contesta para cada petición de HTTP, responder sólo cuando hay acontecimientos para entregar. De este modo, hay siempre una petición pendiente para la cual el servidor puede contestar con el objeto de entregar acontecimientos como ocurren” (Loreto, y otros, 2011)

***-Push*** **(Empujar):** “Enviar datos a un cliente sin que el cliente lo solicite”. (IT Business Edge Network, 2013)

**-Mensajería instantánea:** “La mensajería instantánea constituye una forma de comunicación en tiempo real entre dos o más personas basada en texto, que es enviado a través de dispositivos con acceso a la red. Implica algún método de envío de pequeños y simples mensajes que son inmediatamente enviados a los usuarios de un sistema informático conectados, con presencia activa en ese momento”. (García, 2008)

### 1.3. Descripción de la situación problemática.

La plataforma GeneSIG debe su surgimiento y evolución al amplio desarrollo que han tenido los SIG a nivel mundial, para la toma de decisiones a bajo costo y con un número apreciable de facilidades. Son muchas las empresas en el mundo que están inmersas en la búsqueda de mecanismos y soluciones que permitan mejorar los sistemas y las técnicas existentes, lo que propicia el nacimiento de

instrumentos de este tipo. La estructura arquitectónica de la Plataforma GeneSIG permite personalizar sus funcionalidades a cualquier negocio que lo requiera a través de la reutilización de sus componentes. El proceso de personalización de GeneSIG consiste en adecuar las funcionalidades de dicha herramienta a las condiciones necesarias para la creación de un SIG enmarcado a cualquier negocio. Entre los objetivos fundamentales en los que se centra esta plataforma se encuentran:

- "Permitir la representación geoespacial de la información asociada a cualquier negocio que lo requiera.

- Proporcionar servicios de acceso a la información geográfica, para su consulta, análisis y visualización, mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnología SIG.

- Integrar la información socioeconómica existente (recursos humanos, entidades de servicios, lugares de interés, etc.) con la información geográfica asociada". (Membrides, 2010)

Los SIG pueden ser considerados como Sistemas de Información Geográfica Web debido a que gestionan sus funcionalidades sobre el modelo cliente-servidor. El flujo de información en sistemas web como los SIG se realiza a través del Protocolo de Transferencia de Hipertexto (HTTP) el cual define la sintaxis y la semántica que utilizan los elementos de *software* de la arquitectura web (cliente y servidor) para comunicarse. HTTP está orientado a transacciones y sigue el esquema petición/respuesta entre un cliente y un servidor. El navegador web (cliente) establece una conexión con el servidor y le envía un mensaje HTTP con la petición; a continuación, el servidor envía al cliente otro mensaje HTTP con la respuesta y cierra la conexión (Ver Figura. 1). Al recibir la respuesta del servidor, el navegador actualiza la interfaz de usuario con los datos de todas las modificaciones realizadas por todos los usuarios hasta ese momento. En el modelo estándar de HTTP, un servidor no puede iniciar una conexión con un cliente ni puede enviar al cliente una respuesta HTTP no solicitada; de esta manera el servidor no puede empujar acontecimientos asincrónicos para los clientes (*push*) en caso de haber ocurrido una modificación realizada por otro usuario.


**Figura. 1: Esquema de funcionamiento de un sistema web.**[10]

Un servidor brinda servicios a múltiple clientes (Ver Figura. 2) pero no existe en mecanismo que posibilite que cuando un cliente realice una modificación en la información representada en un SIG esta se represente en los demás clientes a la misma vez. Para solucionar esta situación se adoptó como medida establecer un tiempo determinado de actualización al navegador web por parte del sistema. Esto soluciona de alguna manera la situación pero no es la mejor opción ya que se sobrecarga al navegador con actualizaciones innecesarias si en ese período de tiempo no fueron realizadas modificaciones.

[10] Adaptada de (Olaya, 2010).

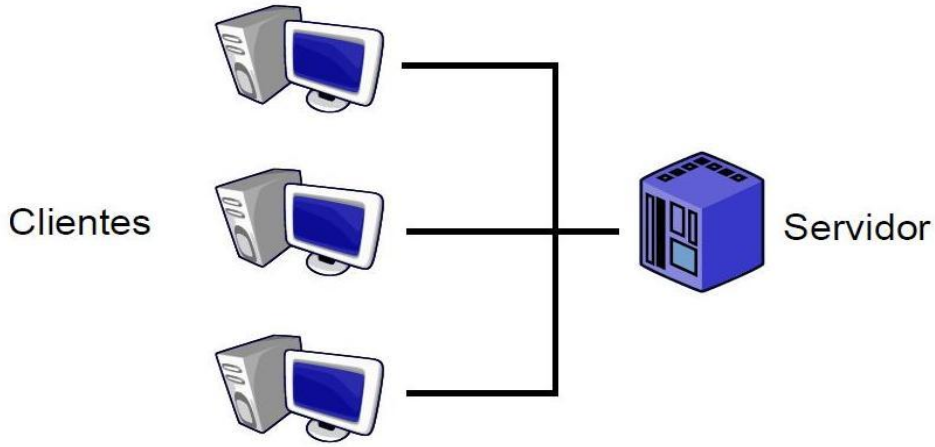



**Figura. 2: Relación entre clientes y servidores.**[11]

Debido a ello es necesario realizar una investigación sobre las principales tecnologías y protocolos de comunicación para la transmisión de datos existentes que puedan dar solución a la problemática representada.

## 1.4. Análisis de tecnologías y protocolos de comunicación.

Un protocolo de comunicación es “un conjunto de reglas que gobiernan el intercambio de datos entre dos entidades”. (UPV, 2006). A continuación se realizará un estudio de los principales protocolo de comunicación existente.

### AJAX[12].

Es una técnica de desarrollo web es decir de lado del cliente, es decir en el navegador web, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, logrando trabajar sin interferir con la visualización ni el comportamiento de la página; lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Esta técnica propone encuestar el servidor periódicamente para obtener

[11] Tomada de (Olaya, 2010).
[12] JavaScript asíncrono y XML.

el contenido nuevo (*polling*). AJAX no es una tecnología en sí misma, es una combinación de tecnologías usada de una manera concreta. Las ventajas de la utilización de AJAX son:

- Utiliza tecnologías ya existentes.
- Soportada por la mayoría de los navegadores modernos.
- Interactividad. El usuario no tiene que esperar hasta que lleguen los datos del servidor.
- Portabilidad.
- Mayor velocidad, esto debido a que no hay que retornar toda la página nuevamente.
- La página se asemeja a una aplicación de escritorio.

Las principales desventajas que presenta son:

- Se pierde el concepto de volver a la página anterior.
- La existencia de páginas con AJAX y otras sin esta tecnología hace que el usuario se desoriente.
- Problemas con navegadores antiguos que no implementan esta tecnología.
- No funciona si el usuario tiene desactivado el JavaScript en su navegador.
- Requiere programadores que conozcan todas las tecnologías que intervienen en AJAX.
- Dependiendo de la carga del servidor se puede experimentar tiempos tardíos de respuesta.

AJAX funciona bien cuando los datos se cambian constantemente pero obliga a realizar el encuestado continuo aún cuando no han existidos modificaciones en la información, realizándose

innecesariamente. Es por ello que se descarta su utilización para dar solución a la problemática presentada.

**WebSocket.**

WebSocket es protocolo que proporciona un canal de comunicación bidireccional y *full-duplex*[13] sobre un único *socket*[14] TCP[15]. Está diseñado para ser implementado en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente-servidor. WebSocket comienza su vida como una conexión HTTP, lo que garantiza total compatibilidad con el mundo pre-WebSocket. Este protocolo no trata de simular un canal de envío de información a partir del servidor sobre HTTP y sino que solamente define un protocolo de empaquetado sobre TCP. De esta manera WebSocket permite la comunicación en dos sentidos de forma nativa. El protocolo tiene dos partes: *handshake*[16] o saludo y la transferencia de datos. El interruptor de protocolo de HTTP a WebSocket se conoce como un saludo inicial y está orientado a ser compatible con *software* del lado del servidor e intermediarios basados en HTTP, de forma tal que un solo puerto puede ser usado por los clientes HTTP y por los clientes WebSocket comunicándose con el mismo servidor. Después de recibir el encabezado de respuesta HTTP, la información será transmitida de acuerdo con el protocolo WebSocket. Esto significa en este punto que solamente serán transmitidos paquetes WebSocket por la red. Un paquete pude ser enviado en cualquier momento y en cualquier dirección. Para establecer una conexión WebSocket, el cliente y el servidor realizan un *upgrade* del protocolo HTTP al protocolo WebSocket durante el saludo inicial. El uso de este protocolo brinda las siguientes ventajas:

- El canal de comunicación es full-duplex, es decir que se puede enviar información en ambos sentidos.

[13] Término utilizado para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.
[14] Mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos.
[15] Protocolo de Control de Transmisión: protocolo que fragmenta los datos en paquetes de información.
[16] Técnica de control de entradas/salidas utilizada para el intercambio de mensajes entre ordenadores y sus periféricos. Implica un diálogo entre los elementos a intercomunicar.

- El tamaño máximo de los encabezados de las tramas17 WebSocket son de 14 bytes por lo que se reduce notablemente la cantidad de información que se transmite por la red.

Se descarta a WebSocket como protocolo de comunicación para la solución de la problemática porque es un RFC- *Proposed Standard (IETF*[18] *Stream)* (IETF), es decir no es un estándar definido. Otra razón por la que no se utilizará WebSocket es que restringiría el uso de la plataforma GeneSIG ya que este protocolo solo es soportado los navegadores Mozilla Firefox 8, Google Chrome 4 y Safari 5, versiones superiores y para versiones inferiores no tiene soporte.

**Protocolo Extensible de Mensajería y Comunicación de Presencia.**

"El Protocolo Extensible de Mensajería y Comunicación de Presencia (XMPP) es una tecnología abierta para la comunicación en tiempo real, utilizando el Lenguaje de Marcado Extensible (XML) como el formato de base para el intercambio de información. XMPP proporciona una manera de enviar pequeñas piezas de XML desde una entidad a otra casi en tiempo real". (Saint-Andre, y otros, 2009)

Si bien varios navegadores web experimentan con características que usan XMPP, ninguno actualmente provee soporte para el protocolo XMPP, dado que el mecanismo de conexión estándar usado por XMPP es TCP. La tecnología que posibilita el encruzamiento eficiente entre HTTP y XMPP se denomina *Bidirectional-streams Over Synchronous* HTTP (BOSH). En esencia, BOSH proporciona una capa de emulación de conexiones TCP sobre el HTTP, bidireccional. Esto es posible porque utiliza la tecnología *Long Polling* que es una variación de la técnica tradicional de *polling* que permite emular información enviada desde un servidor a un cliente en forma similar al *polling* normal. Sin embargo, si el servidor no tiene información disponible para el cliente, en vez de enviar una respuesta HTTP vacía, el servidor guarda la petición HTTP y espera a que alguna información esté disponible. Una vez la información está disponible, se envía una respuesta completa al cliente. Entonces el cliente normalmente realizará un re-pedido de información al servidor, para que éste siempre tenga un pedido

[17] El encabezado de trama contiene la información de control que especifica el protocolo de capa de enlace de datos para la topología lógica específica y los medios en uso, básicamente el encabezado tiene datos que indica el inicio de las tramas, la calidad del servicio o los nodos de origen y destino.

[18] Organización internacional abierta de normalización.

en espera, que puede ser usado para responder a un evento. El uso de BOSH para XMPP brinda grandes facilidades como que:

- Los servidores de XMPP pueden manipular grandes cantidades de usuarios concurrentes.
- Puede ser usado en situaciones dónde el servidor XMPP no está disponible.
- No pierde datos. Los mensajes se guardarán en el servidor hasta que el cliente esté nuevamente.
- BOSH define qué tan arbitrario pueden ser los elementos XML transportados sobre el HTTP en ambas direcciones entre un cliente y servidor.

XMPP típicamente proporciona los siguientes servicios:

- **Codificación de canal:** Este servicio brinda encriptación de las conexiones entre un cliente y un servidor, o dos servidores, para mayor seguridad de las aplicaciones.
- **Autenticación:** Este servicio es definido para desarrollo de aplicaciones seguras. En este caso, el servicio de autenticación asegura que los entes intentando comunicarse deben ser primero autenticados por un servidor que actúa como cierto portero para acceso de red.
- **Presencia:** La función de este servicio es descubrir sobre la disponibilidad de red de otros entes. Típicamente, el compartir información de presencia está basado en un sistema de subscripción entre dos entes a fin de proteger la privacidad del usuario.
- **Mensajería uno a uno:** El propósito de este servicio es enviar mensajes a otra entidad. El uso clásico de uno a uno enviando mensajes que pueden ser arbitrarios XML. Cualesquiera dos entes en una red pueden intercambiar mensajes, ellos pueden ser servidores, componentes, dispositivos o servicios de red de XMPP habilitados.

- **Notificación:** Este servicio, permite generar una notificación y entregarlo a varios suscriptores. Este servicio está optimizado para de uno-a-muchos, este entrega las suscripciones a los canales explícitos o temas específicos (llamados "nodos").

XMPP ofrece ventajas claves como:

**- Abierto:** El protocolo XMPP es libre, gratis, público y fácil de entender, además existen múltiples implementaciones en forma de cliente, servidor, componentes de servidor y librerías de códigos.

**- Descentralizado:** La arquitectura de la red XMPP es similar a la del correo electrónico; como resultado, cualquiera puede correr su propio servidor XMPP, permitiendo a individuos y a organizaciones tomar el control de su experiencia en comunicaciones.

**- Estándar:** La IETF formalizó el núcleo de los protocolos de flujo de información XML como una tecnología aprobada para mensajería instantánea.

**- Seguro:** Cualquier servidor XMPP puede ser aislado de la red pública y presenta una seguridad robusta usando SASL[19] y TLS[20] que están incorporados en el funcionamiento base de las especificaciones XMPP.

**- Flexible:** Las aplicaciones XMPP más allá de la mensajería instantánea incluye manejo de red, sindicación de contenido, herramientas de colaboración, compartición de archivos, juegos, monitoreo de sistemas remotos, servicios Web y mucho más.

**- Diverso:** Un amplio rango de compañías y proyectos de código abierto usan XMPP para construir y desarrollar aplicaciones en tiempo real". (Saint-Andre, y otros, 2009)

Luego de realizar un estudio de las principales tecnologías y protocolos de comunicación se elige para realizar el desarrollo a XMPP como protocolo de comunicación. La principal razón por la que se elige a XMPP como protocolo, además de las ventajas que ofrece, es debido al Servicio de Notificación

[19] *Framework* para autenticación y autorización en protocolos.
[20] Protocolo criptográfico que proporciona comunicaciones seguras por una red.

que brinda, denominado Paradigma Publicación-Suscripción. A continuación se realiza un análisis del mismo.

**Paradigma Publicación-Suscripción.**

"El modelo de Publicación-Suscripción (Pubsub) es un paradigma de envío de mensajes asíncrono mediante el cual los usuarios que publican información (*productor*) no la envían directamente a los usuarios que solicitan esta información (*consumidor*), sino que lo hacen mediante un mediador (o *broker*).

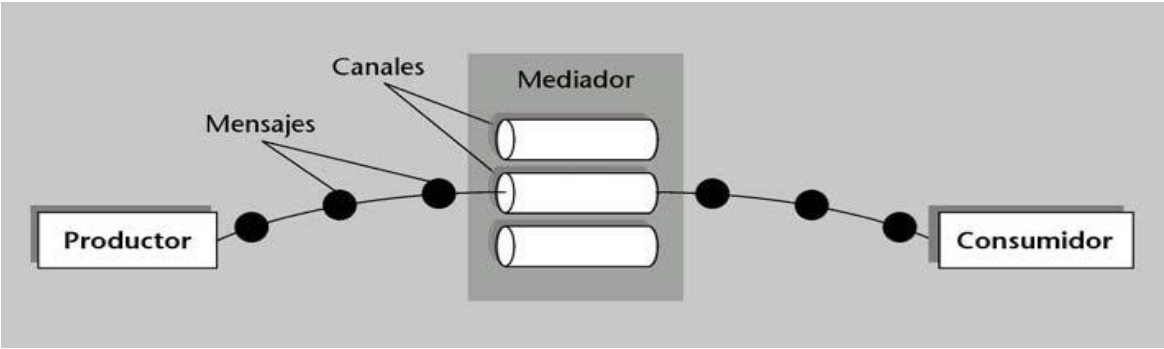



**Figura. 3. Paradigma de Publicación-Suscripción.** [21]

La manera como el paradigma Pubsub aborda esta situación es haciendo que el productor de la información anuncie la disponibilidad de un cierto tipo de información en un canal. El consumidor interesado se debe suscribir a este canal y será avisado cada vez que el productor publique nueva información.

Para que este sistema funcione, se precisan los siguientes componentes:

- **Productor de información**: Usuario que tiene la información a difundir. El productor publica la información en el mediador sin tener conocimiento alguno de los usuarios interesados en esa información.

[21] Imagen tomada de (Juan, 2009).

- **Consumidor de la información**: Usuario interesado en recibir información. El consumidor se suscribe a los temas en los que está interesado y cuando el productor publica información, el consumidor recibe una notificación indicando que hay nueva información disponible.

- **Mediador:** Está situado entre el productor y el consumidor, y se encarga de recibir la información de los productores y las peticiones de suscripción de los consumidores. Además, también se encarga de enviar a los consumidores las notificaciones precisas cuando un productor publica nuevos contenidos.

- **Canal (Nodos):** Se trata de los conectores lógicos entre el productor y el consumidor. El canal determina algunas de las propiedades de la información (tipo de información, formato, etc.). Además también gestiona la forma como se distribuyen los contenidos, si la información expira o es persistente, o si los datos se entregan en el momento de la publicación, o por el contrario el consumidor puede solicitarlos cuando quiera, independientemente del momento en que se generaron.

**Ventajas de Pubsub:**

- **Escalabilidad**: En entornos relativamente pequeños, este sistema ofrece una mejor escalabilidad que el cliente-servidor.

- **Débilmente acoplado**: Se puede entender este concepto como una de las características principales de la publicación-suscripción. Se basa en que el productor no tiene contacto directo con el consumidor (ni el consumidor con el productor), es muy probable que el productor desconozca quiénes son sus consumidores o si estos existen. A diferencia del paradigma cliente-servidor, los usuarios no precisan saber el estado de los demás usuarios. Es decir, los consumidores no dependen de que el productor esté en línea para recibir su información, ni el productor necesita que estén los consumidores conectados para añadir nuevos contenidos.

- **Reducción del tráfico**: Este sistema envía la información sólo a los que están interesados en recibirla, evitando así búsquedas innecesarias en la red". (Juan, 2009)

## 1.5. Conclusiones parciales.

En el capítulo actual se han explicado términos y conceptos asociados al tema de la investigación los cuales permiten un mayor entendimiento del mismo. Se realizó un análisis de la solución problemática y del objeto de estudio con el fin de comprender mejor el problema que está investigación presenta. Se realizó un estudio de los diferentes protocolos y tecnologías existentes que permiten la comunicación en tiempo real existentes lo que permitió seleccionar el que se empleará como solución al problema en cuestión.

# Capítulo 2. Tendencias y tecnologías

## 2.1. Introducción.

En este capítulo, a través de un estudio, se identifican y argumentan las diferentes tendencias y tecnologías actuales a utilizar en el desarrollo de la solución. Mediante este análisis se logra determinar la necesidad e importancia de su manejo y utilización. Dado que la solución propuesta es un módulo para la plataforma GeneSIG se utilizará la arquitectura base propuesta para la misma. Utilizando tecnologías y herramientas definidas con anterioridad por parte de los desarrolladores de este sistema para así asegurar su compatibilidad.

## 2.2. Metodología de desarrollo.

"Una metodología de desarrollo de software, es aquella que hace posible la planificación, organización y construcción de un sistema o proyecto, con independencia de su temática o complejidad. Actualmente estas metodologías son una guía en el proceso de desarrollo de las aplicaciones informáticas, permitiendo que se obtengan resultados con la mayor calidad, rapidez y eficiencia posible, para evitar cometer errores futuros". (Pressman, 2002) Todo el desarrollo de un *software* debe ser regido por metodologías que guíen los procesos a realizar ya que elaborarlo con la calidad requerida no es una tarea sencilla, y más cuando en ocasiones, el propio cliente no tiene bien definido lo que desea. Las metodologías se dividen en dos grupos, ágiles y tradicionales. A continuación se hará en breve resumen del estudio realizado:

Las metodologías tradicionales se enfatizan en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, y son recomendadas para los proyectos con grandes equipos de desarrollo. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, herramientas, notaciones para el modelado y documentación detallada. El Proceso Unificado de Desarrollo (RUP[22]) es la más utilizada de las metodologías tradicionales, esta se divide en cuatro fases las cuales son desarrolladas mediante un ciclo de iteraciones e incrementalmente. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

[22] *Rational Unified Process.*

Debido a que el equipo de desarrollo se encuentra integrado por una sola persona, la disponibilidad del tiempo es limitada y no se hace necesaria la documentación exhaustiva del módulo por lo que se descarta el uso de una metodología tradicional, optando por una ágil para guiar el desarrollo de la investigación.

Las metodologías ágiles están contenidas en el concepto de desarrollo ágil, el cual se basa en la entrega temprana del *software* con el uso de métodos no formales, enfocan su mayor esfuerzo en la elaboración y entrega del producto. Se clasifican de ágiles por su capacidad de responder rápida y efectivamente ante los cambios. Dentro de las metodologías ágiles se encuentra *Extreme Programing* (XP). Esta metodología de desarrollo está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promoviendo el trabajo en equipo, además se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP impone un alto nivel de disciplina entre los programadores. Una desventaja que deviene de la escasa documentación generada por esta metodología es la incapacidad de persistir la arquitectura y demás cuestiones de análisis, diseño e implementación, aún después de que el proyecto haya concluido. Debido a ello se descarta la utilización de esta metodología.

Otra de las metodologías ágiles existentes es el Proceso Unificado Ágil (AUP[23]), la cual es una versión simplificada de RUP descrita en una forma simple, fácil de entender y brinda un enfoque de desarrollo de *software* utilizando técnicas ágiles y conceptos del RUP. El ciclo de vida de esta metodología abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. AUP adopta muchas de las técnicas ágiles de XP y otros procesos ágiles manteniendo la formalidad de RUP. Fue seleccionada para guiar el ciclo de desarrollo de la solución debido principalmente a que está basada en la entrega temprana del *software*, lo que resulta conveniente puesto que el módulo a desarrollar se necesita en un corto plazo de tiempo dado las necesidades del LPS Aplicativos_SIG. Genera la documentación necesaria siendo esto imprescindible para asegurar la posterior comprensión del proceso de desarrollo.

[23] *Agile Unified Process.*

## 2.3. Lenguajes.

A continuación se realiza un resumen de las características de los lenguajes a utilizar para el desarrollo de la solución propuesta a la problemática.

### 2.3.1. Lenguaje de Modelado.

Como lenguaje de modelado para el desarrollo del módulo se decide utilizar al Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) en su versión 2.0. “UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas”. (Giraldo, y otros, 2005)

“Está pensado para usarse con todos los procesos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar”. (Fowler, y otros, 1999)

### 2.3.2. Lenguaje de programación del lado del cliente.

Para el desarrollo del módulo se elige a JavaScript (JS) como lenguaje de programación del lado del cliente. JS es un lenguaje interpretado orientado a objetos que puede funcionar además como lenguaje procedimental. Este lenguaje es dinámico y la soporta construcción de objetos basado en prototipos, por lo que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, en el que no es necesario compilar los programas para ejecutarlos así que los programas escritos con JS se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

### 2.3.3. Lenguaje de programación del lado del servidor.

Como lenguaje de programación del lado del servidor se selecciona a *Hypertext Preprocessor* (PHP) en su versión 5.0. PHP "es un lenguaje *Open Source*[24] interpretado de alto nivel, especialmente pensado para desarrollos web y puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. Este lenguaje brinda como principales ventajas que:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad". (Sæther, y otros, 2002)

## 2.4. Bibliotecas.

A continuación se realiza un resumen de las características de las bibliotecas a utilizar para el desarrollo del módulo.

### 2.4.1. Strophejs.

"Strophejs es una biblioteca JavaScript que facilita conexiones TCP persistentes, esta biblioteca se basa en BOSH para emular la persistencia, con estado de conexión, en ambos sentidos a un servidor

[24] Código abierto.

XMPP. Strophejs es una API[25] pequeña y fácil de entender, ya desde hace algunos años ha sido probado en los principales navegadores y su comunidad de desarrolladores ha ido aumentando. Esta biblioteca detecta muchos errores e incluye optimizaciones que ningún otro marco de AJAX permite hacer, ofreciendo así excelentes rendimientos." (Moffitt, 2010)

Se define utilizar a la biblioteca Strophejs en su versión 1.0 con el propósito principal de permitir establecer una conexión persistente desde el módulo a desarrollar con el navegador web.

### 2.4.2. ExtJS.

ExtJS, "es una biblioteca javascript ligera y de alto rendimiento, compatible con la mayoría de navegadores. Esta biblioteca permite crear páginas e interfaces web dinámicas, realizar aplicaciones web enriquecidas basándose en tecnología AJAX, JSON[26], DHTML[27] y DOM[28]. Esta librería incluye:

- Componentes interfaces de usuarios del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open Source y comerciales.

ExtJS tiene como principales ventajas:

- Crear aplicaciones complejas utilizando componentes predefinidos.
- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, Internet Explore, Safari, Opera, etc.).

[25] Interfaz de programación de aplicaciones: Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.
[26] *JavaScript Object Notation*: Es un formato ligero para el intercambio de datos.
[27] HTML Dinámico: Designa el conjunto de técnicas que permiten crear sitios web interactivos.
[28] Modelo de Objetos del Documento: API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

- Uso de ventanas flotantes.

- Relación entre cliente-servidor balanceada: Se distribuye la carga de procesamiento permitiendo que el servidor pueda atender más clientes al mismo tiempo.

- Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desea trasmitir al servidor y viceversa.

- Comunicación asíncrona. El motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a una acción del usuario, dándole la libertad de cargar información sin que el cliente se de cuenta." (Ramírez, 2012)

## 2.5. Herramienta CASE[29].

Se selecciona como herramienta para el modelado del módulo a *Visual Paradigm* para UML (VP-UML) en su versión 8.0 *Enterprise Edition*. VP-UML "es una herramienta de diseño UML y herramienta CASE-UML diseñada para ayudar al desarrollo de *software*. VP-UML soporta los estándares de modelado clave como UML, SysML[30], ERD[31], DFD[32], BPMN[33], etc. Es compatible con los equipos de desarrollo de *software* de captura de requerimientos, planificación de *software* (utilizar el análisis de casos), ingeniería de código, modelado de clases, modelado de datos, etc." (Visual Paradigm International, 2013)

## 2.6. Entorno de Desarrollo Integrado.

"Un Entorno de Desarrollo Integrado (IDE) es un *software* que sirve de base para la implementación de una aplicación de forma visual, sencilla y práctica. Es un programa compuesto por un conjunto de herramientas que facilita al programador el desarrollo de un *software*. Los IDE por lo general contienen un editor de código, un compilador, un depurador y un constructor de interfaces gráficas. Los más

[29] Ingeniería de *Software* Asistida por Computadora.
[30] Lenguaje de Modelado de Sistemas.
[31] Diagrama de Entidad Relación.
[32] Diagrama de Flujo de Datos.
[33] Notación para el Modelado de Procesos de Negocio.

modernos IDE incluyen un navegador de clases, un inspector de objetos y se integran con sistemas de control de versiones. Los mismos están hechos para soportar uno o más lenguajes de programación." (Cruz, y otros, 2008)

### 2.6.1. Netbeans

"NetBeans es un entorno de desarrollo visual de código abierto. Está disponible para diversos sistemas operativos como Solaris, Windows, MacOS y GNU Linux. Permite a los desarrolladores crear rápidamente aplicaciones web, de escritorio y móviles utilizando la plataforma Java. Soporta lenguajes como PHP, JavaScript y AJAX, Groovy y Grails, y C / C + +. Es un producto gratuito y no tiene restricciones de uso, tiene una comunidad que le ofrece soporte a nivel mundial. Permite que las aplicaciones sean desarrolladas a partir de módulos, lo cual posibilita que se puedan construir aplicaciones extensibles. Netbeans incluye herramientas de esquemas XML, orientación a servicios Web y modelado UML." (Oracle Corporation, 2013) Para el desarrollo de la solución se utilla Netbeans en su versión 7.0.1.

## 2.7. Servidor Web.

Se elige como servidor web a Apache en su versión 2.0.6, porque "es un servidor web de código libre para HTTP disponible para plataformas Unix, Windows, y Macintosh, entre otras. Su implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. Es un producto gratuito y no tiene restricciones de uso, cuenta con una comunidad que le ofrece soporte." (Mateu, 2007) "Sus principales características son:

- Servidor de web conforme al protocolo HTTP/1.1
- Soporta tanto *host* basados en IP como *host* virtuales.
- Apache soporta autentificación básica basada en la Web.

- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.

- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

- Personalización de las respuestas ante los posibles errores que se puedan dar en el servidor." (Ford, 2008)

## 2.8. Servidor de mensajería instantánea.

Se determinó como servidor XMPP a Ejabberd[34] en su versión 2.1.10. Ejabberd "es un servidor de mensajería instantánea de código abierto (GNU GPL) Esta versión actualmente es la más estable, es escrita principalmente en Erlang[35] y además está basada en los estándares abiertos para lograr la comunicación en tiempo real utilizando para ello el protocolo XMPP. A continuación se presentan algunas de las características más importantes del ejabberd.

- **Multiplataforma:** Funciona en las plataformas operativas más populares.

- **Distribuido:** Se puede ejecutar Ejabberd en más ordenadores y todos ellos se regirán por un único dominio de Jabber. Cuando se desea ampliar un servidor Jabber, simplemente se puede agregar un nuevo nodo al clúster[36].

- **Tolerante a fallos:** Los nodos del clúster Ejabberd comparten algunas o todas las tablas de las bases de datos, para que toda la información requerida por un servicio que tiene que funcionar adecuadamente se almacene permanentemente no en más de un nodo. Si uno de los nodos falla los otros seguirán funcionando sin ninguna interrupción.

[34] *Erlang Jabber Daemon*.
[35] Lenguaje de programación concurrente y funcional con evaluación estricta, asignación única y tipado dinámico.
[36] Conjunto de computadoras que utilizan componentes comunes y actúan como si se tratase de un solo sistema u ordenador.

- ***Hosts***[37] **virtuales:** Pueden alojarse diferentes *hosts* de Jabber en la misma instancia de Ejabberd. Solamente es necesario añadir un nuevo nombre de dominio a la lista de hosts del archivo de configuración." (Ejabberd Development Team, 2011)

## 2.9. Conclusiones parciales.

En el presente capítulo se han seleccionado las tecnologías y herramientas que se adecuan a las características del módulo que se desea desarrollar. Se asegura el uso de tecnologías libres, promoviendo la utilización de desarrollo de código abierto. Se realizó un estudio de las metodologías de desarrollo existentes con el propósito de seleccionar la más adecuada para la solución de la investigación y fue seleccionada la metodología AUP como guía en el proceso de desarrollo del *software* ya que permite que en un tiempo relativamente corto el proceso de desarrollo quede debidamente documentado. Ya concluido este capítulo se procederá a brindar una propuesta de solución del sistema basados en estas tecnologías y herramientas definidas.

[37] También conocido como anfitrión, es un ordenador que funciona como el punto de inicio y final de las transferencias de datos. Es todo equipo informático que posee una dirección IP y que se encuentra interconectado con uno o más equipos.

# Capítulo 3. Presentación de la solución propuesta

## 3.1. Introducción.

En este capítulo se formalizará la propuesta de solución al problema existente, con el apoyo de la metodología AUP para la planificación, investigación y diseño del módulo. A continuación se realizará un análisis del módulo a desarrollar y se definirán las características que tendrá. Además se presenta la descripción del modelo de dominio, así como el diagrama de clases correspondiente al modelo de objetos, incluyendo la gestión de requerimientos a través de las diferentes técnicas que existen para la captura, definición y validación de los mismos, pues estos constituyen la base para la realización del modelo de casos de uso.

## 3.2. Descripción del módulo a desarrollar.

El módulo a desarrollar tiene dos objetivos fundamentales: El primero es monitorear las modificaciones generadas por los usuarios en la gestión de la información geográfica. Su segundo objetivo es establecer la comunicación entre las estaciones de trabajo y el servidor XMPP, para así poder mostrar las modificaciones de los objetos geográficos gestionados, en tiempo real. Entre las principales funcionalidades que brindará se destacan el módulo:

- Establecer conexión al servidor XMPP: permite tener registro de las estaciones de trabajo que deberán ser notificadas de los cambios realizados por los usuarios.

- Registrar los plugin activos que gestionan objetos geográficos: permite reconocer a través de que plugin el usuario realiza las modificaciones y así obtener los datos que constituyen las modificaciones realizadas.

- Crear el nodo Pubsub de transmisión de datos: permite abrir el canal por el cual se enviarán los datos de las modificaciones realizadas.

- Publicar datos: permite publicar en tiempo real los datos que constituirán las modificaciones realizadas por los usuarios.

- Suscribir a nodo: permite suscribir las estaciones de trabajo al nodo en el servidor XMPP por donde se publicarán las actualizaciones.

Luego de descritas las principales características que tendrá el módulo se pasa a definir el modelo a utilizar para la realización del mismo.

### 3.3. Modelo de dominio.

El Modelo de Dominio es “la representación visual de los conceptos u objetos más importantes de un negocio, sus características y las relaciones entre dichos conceptos. Es el mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes. Es un diccionario visual del dominio del problema.” (Pressman, 2002) Se utiliza modelo de dominio en el desarrollo del módulo debido a la poca estructuración de los procesos de gestión de la información. Al no encontrarse definidos de forma clara los diferentes procesos del negocio y que los mismos, pueden estar sujetos a cambios, se propone como paso inicial, y para poder entender el contexto en que se desarrolla el sistema es necesario definir conceptos que sea posible agrupar en un Modelo de Dominio.

### 3.4. Descripción del diagrama.

La plataforma GeneSIG sirve de base para el desarrollo de SIG adaptables a cualquier entorno. Uno de los componentes fundamentales de estos SIG es el mapa, portador de información geográfica y socioeconómica que son necesarias para determinada organización o entidad. El gestor geográfico se encarga de aplicar los cambios realizados por los usuarios y de representarlos en los mapas. Tras la ocurrencia, las modificaciones son captadas por el observador e informadas al monitor quien se encarga de a través del servidor XMPP de actualizar el SIG en dependencia de los datos recibidos del observador.

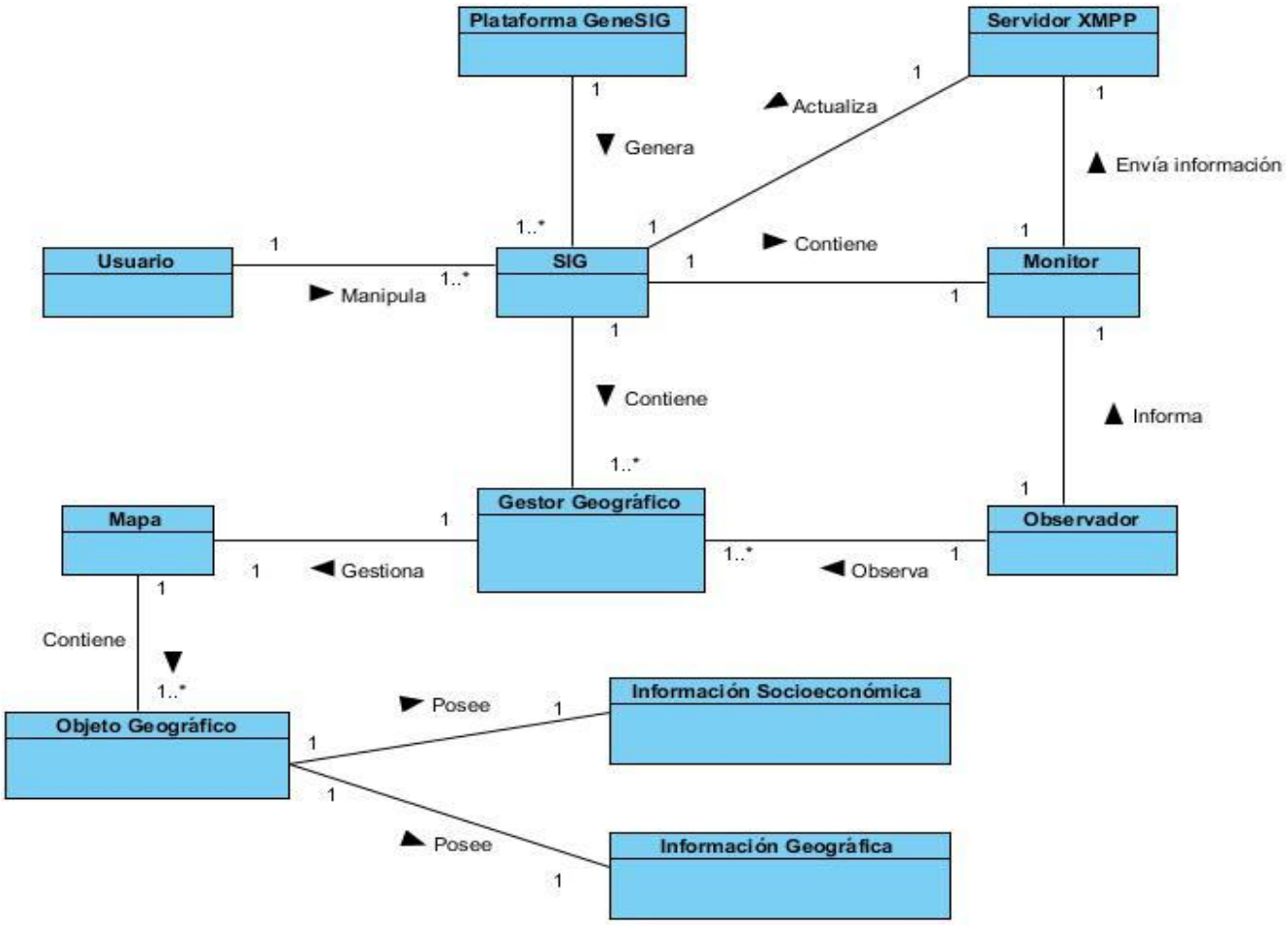



**Figura. 4: Diagrama de clases del Modelo de Dominio.**

### 3.5. Definición de clases del modelo del dominio.

- **Plataforma GeneSIG:** Sirve de base y modelo para el desarrollo de SIG. Los SIG generados tienen una estructura definida por ella.

- **SIG:** Personalización de la plataforma GeneSIG, contiene gestores de datos geográficos para la gestión de sus componentes.

- **Usuario:** Persona encargada de operar los SIG. Tiene la responsabilidad de consultar y gestionar los objetos geográficos de los mapas.

- **Gestor geográfico:** Es un componente interoperable de los SIG generados en la plataforma GeneSIG, quien se encarga de gestionar los mapas y sus objetos geográficos. Cada gestor tiene su propia función.

- **Mapa:** Es una representación digital de una porción de territorio. En él se ubican los objetos geográficos.

- **Objetos geográficos:** Son abstracciones de elementos del mundo real que están asociadas a una posición geográfica (información geográfica) y pueden tener además información socioeconómica.

- **Información geográfica:** Conjunto organizado de datos espaciales georreferenciados.

- **Información socioeconómica:** Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés.

- **Monitor:** Forma parte del SIG, se encarga de enviar la información obtenida por el observador referente a las modificaciones realizadas sobre el SIG, a través del servidor XMPP.

- **Observador:** Realiza un constante monitoreo del comportamiento de los gestores geográficos a la espera de modificaciones, cuando estas ocurren las envía como información al monitor.

- **Servidor XMPP:** Se encarga del envío de los datos a los SIG con las actualizaciones contempladas en ellos.

## 3.6. Especificación de requisitos.

Para determinar de manera correcta las funcionalidades que el módulo debe tener es necesario antes identificar los requisitos necesarios.

### 3.6.1. Técnicas de captura de requisitos.

“La captura de requisitos ayuda al equipo de desarrollo de un sistema a entender mejor el problema en cuya solución trabajarán. Este proceso incluye un conjunto de tareas que conducen a comprender, qué es lo que el cliente quiere y como interactuarán los usuarios finales del *software*”. (Pressman, 2002)

Para hacer el levantamiento de requisitos y determinar las funcionalidades necesarias para la elaboración del módulo a implementar se utilizaron las técnicas de:

- **Tormenta de ideas:** “Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la acumulación de ideas e información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar”. (Gallego, 2011)

- **Entrevistas:** “Resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural”. (Gallego, 2011)

### 3.6.2. Requerimientos funcionales.

“Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.” (Rumbaugh, y otros, 1999)

- **RF 1. Conectar al servidor XMPP las de estaciones de trabajo.**

Esta funcionalidad debe permitir que las estaciones de trabajo desde donde se trabajará con el SIG se establezcan como usuario en el servidor XMPP una vez iniciado el sistema.

- **RF 2. Conectar al servidor XMPP el módulo de monitoreo.**

Esta funcionalidad permite que el módulo se establezca como usuario en el servidor XMPP una vez iniciado el trabajo con el SIG.

- **RF 3. Autenticar usuario.**

Esta funcionalidad debe permitir identificar con que usuario se registrarán en el servidor XMPP los que se conecten a él.

- **RF 4. Observar gestores geográficos activos.**

Esta funcionalidad debe permitir identificar los gestores geográficos que están activos y obtener la información correspondiente a las modificaciones realizadas en el SIG.

- **RF 5. Crear nodo.**

Esta funcionalidad permite al módulo crear el nodo Pubsub para la publicación de información.

- **RF 6. Publicar datos.**

Esta funcionalidad debe permitir publicar la información, recogida por el observador, en el nodo.

- **RF 7. Subscribir a nodo.**

Esta funcionalidad debe permitir a las estaciones de trabajo subscribirse al nodo Pubsub del módulo de monitoreo.

- **RF 8. Actualizar el SIG.**

Esta funcionalidad debe permitir actualizar el SIG con los datos recibidos.

### 3.6.3. Requisitos no funcionales.

"Los requisitos no funcionales son aquellas propiedades que debe tener la solución y que no son una funcionalidad." (Rumbaugh, y otros, 1999) El módulo debe cumplir con los requisitos no funcionales pertenecientes a la plataforma GeneSIG, además de los que a continuación se definen:

- **Usabilidad.**

-El módulo podrá ser usado por personas con conocimientos de la plataforma GeneSIG.

- **Fiabilidad.**

-La información manejada por el módulo estará protegida de acceso no autorizado y divulgación.

-La información y las funcionalidades del módulo estarán disponibles y el usuario podrá acceder a ellas las 24 horas de los 7 días de la semana si tiene los permisos necesarios.

- **Eficiencia.**

-El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.

-Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar.

- **Soporte.**

-La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

- **Restricciones de diseño.**

-El producto de software final debe diseñarse sobre una arquitectura que permita una integrabilidad con la plataforma GeneSIG.

-Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará para la plataforma GeneSIG que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

- **Requisitos de Licencia.**

-De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen a utilizar para el desarrollo del producto se puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

- **Requisitos Legales, de Derecho de Autor y otros.**

-El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

-La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

-Como producto, se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

## 3.7. Descripción del sistema propuesto.

Para lograr una mejor comprensión del módulo propuesto es necesaria realizar una descripción de sus principales características por lo tanto se comenzará realizando la descripción de los actores que intervienen en él.

### 3.7.1. Descripción de los actores del sistema.

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

**Tabla 1. Descripción de los actores del sistema.**

| Actor | Descripción |
|---|---|
| **Usuario** | Representa a toda aquella persona que gestiona información el SIG. |
| **Monitor** | Representa un módulo de la plataforma GeneSIG. |

### 3.7.2. Modelo de caso de uso del sistema.

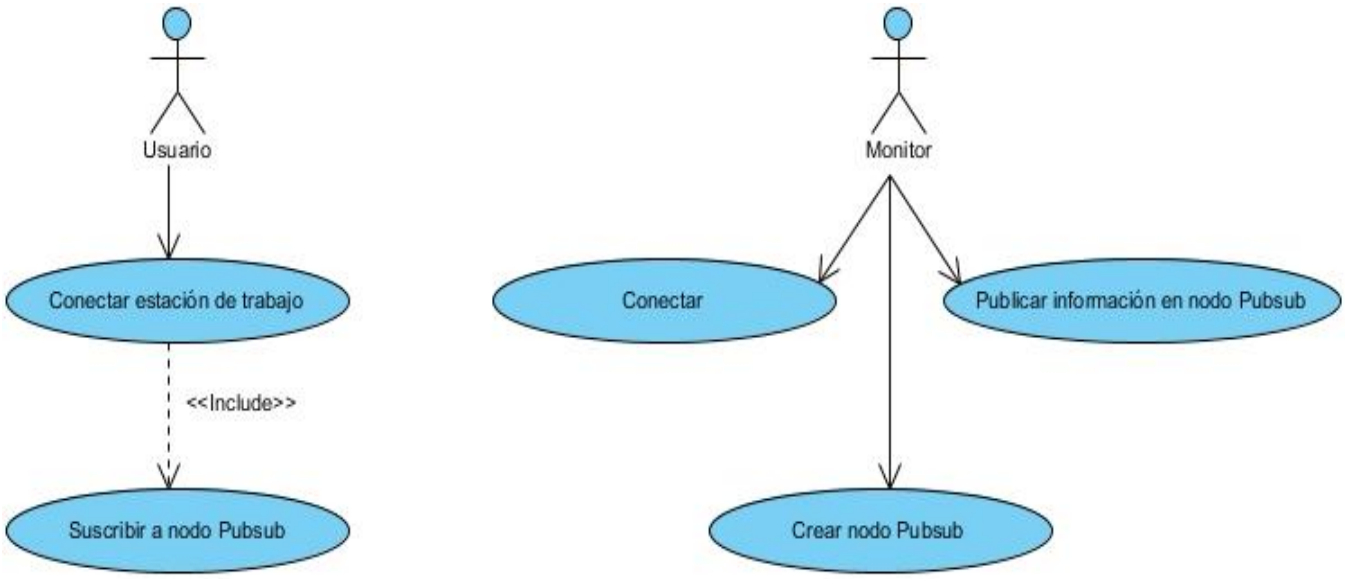



**Figura. 5: Diagrama de clases del Modelo del Sistema.**

### 3.7.3. Especificación de los Casos de Uso.

A continuación se muestra la descripción textual de los caso de uso *Crear nodo Pubsub* y *Publicar información en nodo Pubsub* para ver el resto ir a Anexo 1.

**Tabla 2. Descripción textual del caso de uso Publicar información en nodo Pubsub.**

| | |
|---|---|
| **Caso de Uso** | Publicar información en nodo Pubsub |
| **Actores** | Monitor |
| **Propósito** | Este caso de uso se realiza con el objetivo de recoger los datos de las modificaciones realizadas por los usuarios en el SIG y publicarlos el nodo Pubsub. |
| **Resumen** | Este caso de uso se inicia cuando al módulo monitor llega una |

| | |
|---|---|
| | notificación un cambio en la información geográfica y la publica en el nodo Pubsub. |
| **Precondiciones** | Debe haberse ejecutado el Caso de uso Conectar estación de trabajo y el de Crear nodo Pubsub. |
| **Referencias** | RF7 |
| **Prioridad** | Crítico |

| **Flujo Normal de Eventos** | |
|---|---|
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El monitor recibe un mensaje notificando un cambio en la información geográfica. | 1.1 El sistema obtiene la información proveniente en el mensaje notificado.<br>1.2 El sistema publica esta información en el nodo Pubsub a través del servidor XMPP. |
| | 2. El caso de uso concluye cuando el sistema actualiza los mapas correspondientes. |
| ***Prototipo de Interfaz*** | |
| | |
| **Poscondiciones** | El sistema actualiza los mapas con la nueva información. |

### 3.8. Conclusiones parciales.

En el presente capítulo se identificaron las características y cualidades que la solución propuesta debe cumplir, definiéndose los requisitos funcionales y no funcionales. Además se logró una mejor comprensión de las funcionalidades del sistema a través de la descripción de los casos de uso del módulo. Una vez cumplido este objetivo, es posible pasar a la implementación del Módulo de monitoreo para la gestión de la información geográfica en las personalizaciones de la plataforma GeneSIG ya que se cuenta con la documentación referente a como se implementará.

# Capítulo 4. Construcción y prueba de la solución propuesta

## 4.1. Introducción.

En el desarrollo de este capítulo se abordará todo lo referente al diseño, implementación y prueba del módulo propuesto. Para ello se realiza el modelo del diseño que describe las clases principales que se implementan con sus respectivas relaciones y atributos. Se definen los patrones de arquitectura a utilizar para el desarrollo de la solución, así como los patrones de diseño. Después se define el diagrama de despliegue el cual juega un papel fundamental en la etapa de implementación del sistema. Para finalizar con el capítulo y comprobar la calidad con que cuenta el módulo se presentan los casos de pruebas asociados al mismo.

## 4.2. Arquitectura.

La Arquitectura del *Software* es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución." Teniendo en cuenta que el módulo será desarrollado bajo la estructura definida por la plataforma GeneSIG es imprescindible realizar una descripción de la estructura de la misma, para así asegurar la comprensión del módulo.

La estructura de GeneSIG está basada en el framework CartoWeb que es una "aplicación de publicación WebGIS[38] construida en PHP sobre UMN MapServer[39] que explota AJAX[40]. Su característica más diferenciadora con respecto a otros proyectos de clientes web ligeros sobre MapServer es que CartoWeb ofrece un *framework*[41] que ha sido diseñado con una arquitectura bastante modular y escalable, lo que permite poder separar la lógica de un servidor (cartoserver) encargado del diálogo con MapServer y provisión de servicios, de un cliente (cartoclient)." (Montesinos, y otros, 2007)

[38] Un sistema de información geográfica basado en la web, es una herramienta en línea para representar la información espacial a través de Internet.

[39] MapServer es un entorno de desarrollo de código abierto para construir aplicaciones de Internet.

[40] AJAX s una técnica de desarrollo web para crear aplicaciones interactivas

[41] Marco de trabajo: define un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

"El diseño de CartoWeb está conformado por paquetes (ver Figura. 7), donde cada uno de ellos realiza una función determinante y la relación interactiva entre los mismos conlleva al funcionamiento óptimo del *framework*.

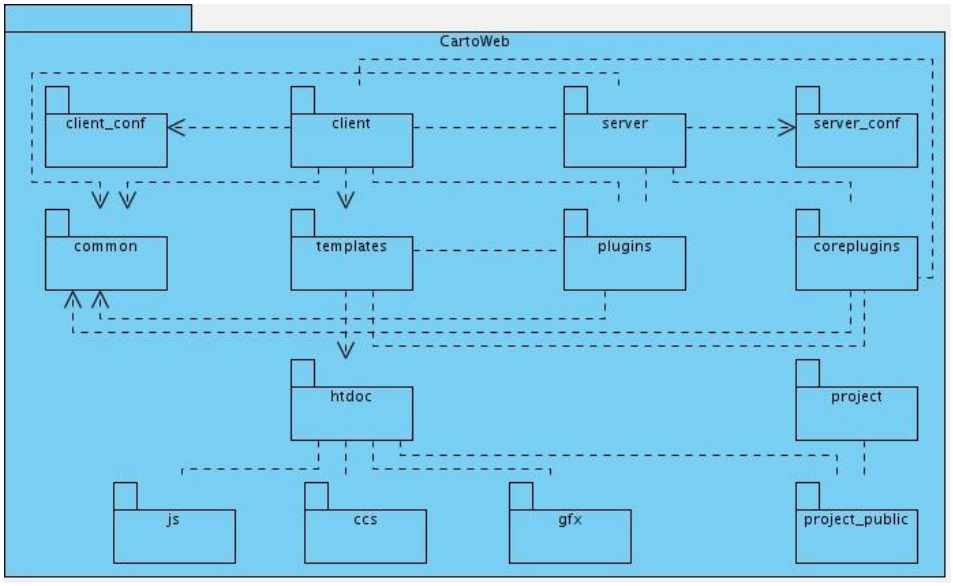



**Figura. 6. Estructura de CartoWeb.**[42]

GeneSIG cumple y respeta de manera estricta la estructuración del diseño propuesto por CartoWeb, pero solo utiliza los paquetes a los cuales le realizará cambios o aportes funcionales (como muestra la Figura.8). CartoWeb, presenta un abanico bastante completo de características propias de un geoportal[43], con la posibilidad de ir añadiendo o desarrollando nuevos plugins. Y es precisamente a través de estos plugins, que GeneSIG posee un amplio conjunto de funcionalidades, que actúan como herramientas de la misma plataforma y le brindan la posibilidad de ser altamente modular y escalable." (Martínez Ledea, y otros, 2011)

[42] Tomada de (Martínez Ledea, y otros, 2011)

[43] Geoportal: Es un punto de acceso vía Internet a información geográfica. Mediante un geoportal se utiliza la red para permitir el descubrimiento, acceso y visualización de los datos geoespaciales, utilizando un navegador estándar de Internet.

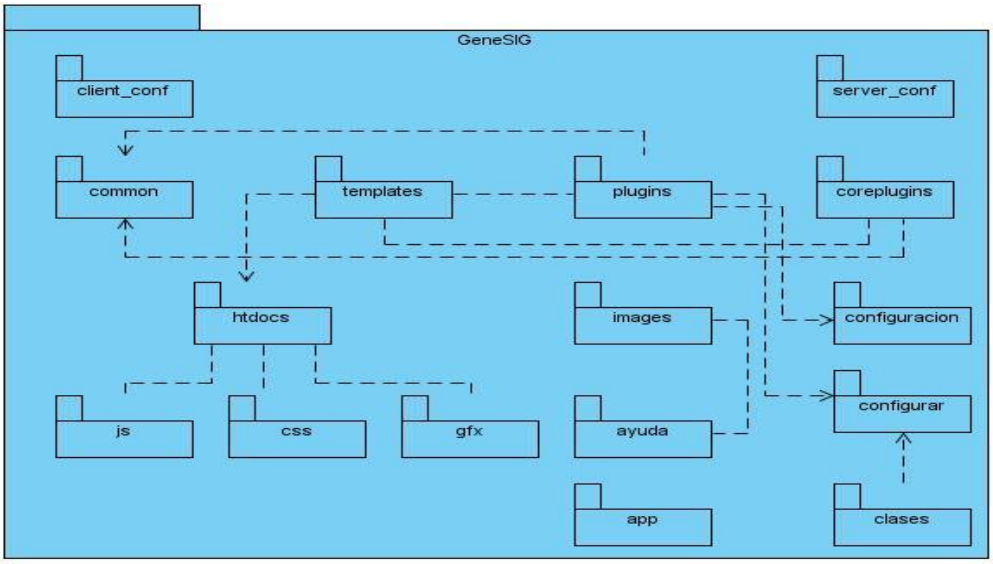



**Figura. 7. Estructura de GeneSIG.**[44]

Como se expresó anteriormente módulo a desarrollar está basado en la estructura de GeneSIG y se define como un plugin por lo tanto coincide con la estructura propuesta para estos por la plataforma (ver Figura. 9).

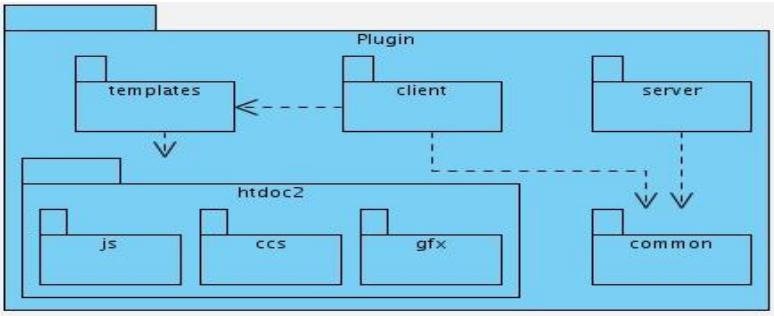



**Figura. 8. Estructura de un plugin.**[45]

[44] Tomada de (Martínez Ledea, y otros, 2011)
[45] Tomada de (Martínez Ledea, y otros, 2011)

**Tabla 3. Descripción de los paquetes del plugin.**

| Módulos | Descripción |
|---|---|
| client | Contiene todos los componentes de extensión .php de los plugins que implementan el módulo correspondiente al lado del cliente. Encargado de recibir las peticiones AJAX procedentes de las Interfaces de usuario y de entregarles las respuestas provenientes del paquete server, encargado de implementar toda la lógica de negocio. |
| server | Contiene todos los componentes de extensión .php del plugins que implementan la lógica del negocio. |
| common | Contiene todos los componentes que implementan aquellas clases que servirán de puente para la comunicación entre cliente-servidor, incluyendo el fichero *.wsdl.inc, donde se especifica cómo acceder a estas vía SOAP[46]. |
| templates | Contiene específicamente el fichero *.tpl basado en el generador de plantillas Smarty para el procesamiento de datos. |
| htdocs/js | Contiene todos los componentes de extensión .js del plugins que implementan el módulo cliente en javascript, además de definir las Interfaces de Usuario. |
| htdocs/css | Contiene las hojas de estilo que serán utilizadas en el plugins. |
| htdocs/gfx | Contiene las imágenes que serán utilizadas en el plugins. |

### 4.2.1. Patrones de arquitectura

"Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de *software*. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e

[46] Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones de arquitectura representan el nivel más alto en el sistema de patrones.” (Larman, 1999)

**- Arquitecturas Orientadas a Objetos:**

Esta arquitectura tiene como principales características que “sus componentes son los objetos, o de forma más acertada, las instancias de los tipos de dato abstractos. Estos se basan en principios Orientados a Objetos: encapsulamiento, herencia y polimorfismo. Son asimismo las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. Otra de sus características es que las interfaces están separadas de las implementaciones. En general la distribución de objetos es transparente, y en el estado de arte de la tecnología apenas importa si los objetos son locales o remotos. Entre las cualidades que presenta esta arquitectura se encuentran:

- La implementación de un objeto puede ser modificada sin que esto afecte a sus clientes.

- Un objeto es ante todo una entidad reutilizable en el entorno de desarrollo.” (Kicillof, y otros, 2004)

**- Arquitecturas Basadas en Componentes:**

Actualmente se trata mucho el tema de la reutilización, ya sea de módulos o partes de *software* existentes, tratando así de optimizar el proceso de desarrollo de nuevas aplicaciones. Dentro de este contexto, el término componente es muy empleado, pues en los procesos de ingeniería, dichas partes de *software* que pueden ser útiles para la creación de aplicaciones, son conocidas como: componentes de *software*. “Un componente de *software*, es una unidad de composición con interfaces especificadas contractualmente y dependencias explícitas del contexto.” (Kicillof, y otros, 2004)

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

**- Arquitectura basada en eventos.**

Este tipo de arquitectura se vincula históricamente con sistemas basados en publicación-suscripción. Los conectores de estos sistemas incluyen procedimientos de llamada tradicionales y vínculos entre anuncios de eventos e invocación de procedimientos. Se hace necesario aplicar esta arquitectura además porque procura mejorar la eficiencia, eliminando la necesidad de *polling* por ocurrencia de evento, así como otras ventajas que brinda dicho estilo. La idea dominante en la invocación implícita, como se le ha llamado también a las arquitecturas basadas en eventos, es que, en lugar de invocar un procedimiento en forma directa, un componente puede anunciar mediante difusión uno o más eventos. En términos de patrones de diseño, el patrón que corresponde más estrechamente a este estilo es el que se conoce como Observador.

**Justificación de la arquitectura elegida.**

La plataforma GeneSIG posee una arquitectura orientada a objetos y de componentes ya que está desarrollada sobre el *framework* CartoWeb y esta es la arquitectura que este tiene por definición. Por lo tanto el módulo será diseñado sobre esta arquitectura también, pero a su vez se define además sobre el patrón arquitectónico basado en eventos ya que esta es la arquitectura base para el desarrollo de sistemas basados en la publicación-suscripción.

## 4.3. Diseño de la solución.

Para desarrollar un sistema es necesario contar con descripciones detalladas y de alto nivel de la solución lógica y saber cómo satisfacer los requerimientos y las restricciones. El diseño tiene el objetivo de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución. Constituye el punto inicial para la fase de implementación y prueba del sistema, por lo que se puede definir como propósito del mismo:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.

- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

### 4.4. Diagramas de clases del diseño

Los diagramas de clases del diseño, representan las relaciones entre clases que mantienen la información manipulada por el sistema, constituyen además, la interacción de las clases del diseño y sus objetos en la realización de los casos de uso, de forma que de cada uno se desprende un diagrama de clases. El diagrama de clases del diseño presentado a continuación está basado en la implementación que se realiza con el framework (CartoWeb) utilizado, además de estar diseñados bajo una arquitectura de componentes. Debido a la complejidad y la extensión de la solución, incluyendo específicamente todas las clases contenidas en el *framework*, y siguiendo la premisa de ajustarse al flujo propio de la solución y no a los detalles transparentes al equipo de desarrollo, se decidió representar únicamente las clases y extensiones web que tuvieran que ver directamente con la construcción de la solución, sin dejar de incluir las partes fundamentales del *framework* que ayuden a su comprensión. A continuación, se muestra el diagrama de clases del diseño del caso de uso *Publicar información en nodo Pubsub* para ver el resto ir a Anexo 2:

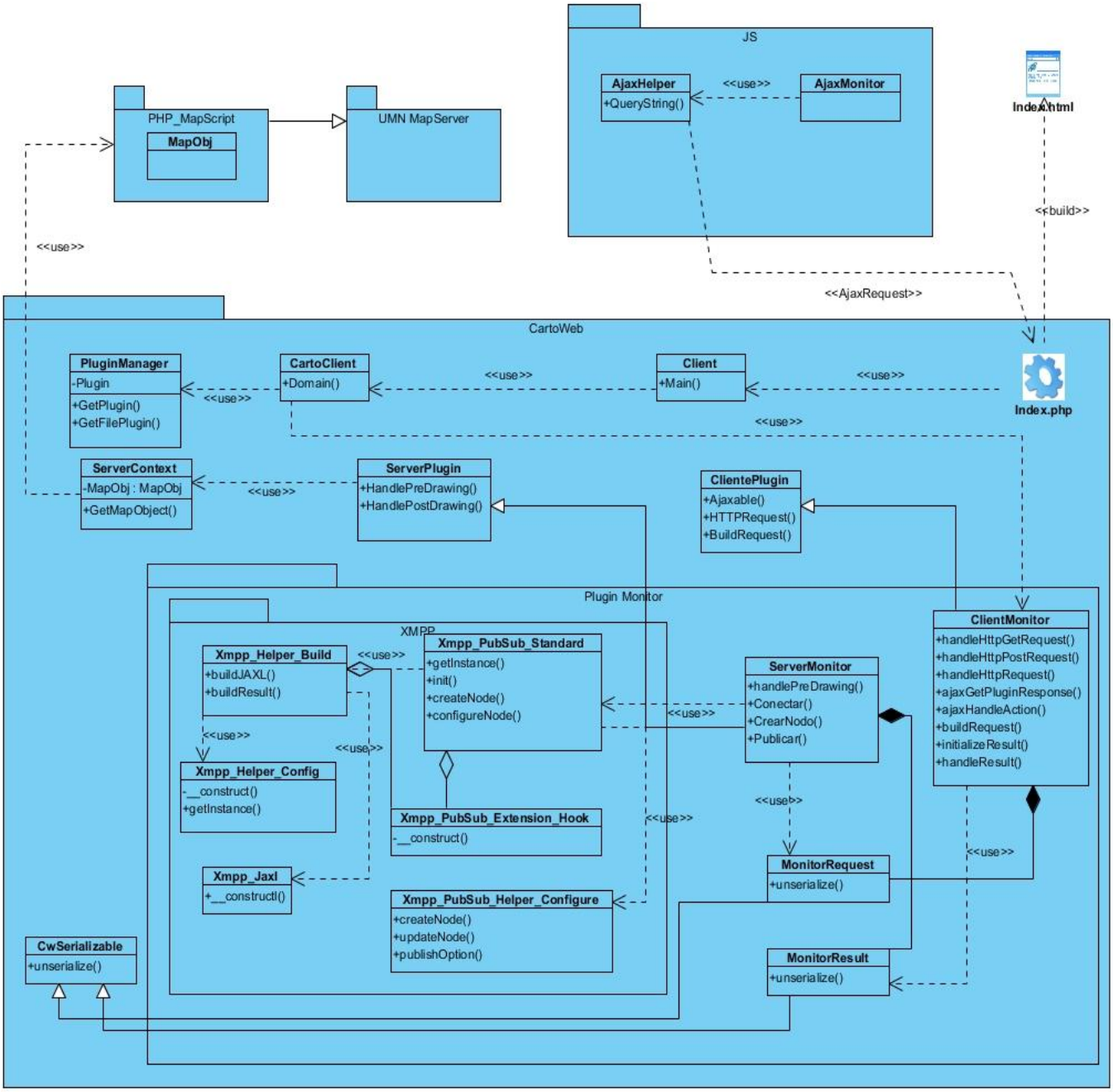



**Figura. 9: Diagrama de clases del Diseño.**

A continuación se realizará una descripción de las clases del diseño representadas anteriormente:

- **Index.php:** Tiene como propósito controlar la realización del CU en sí, recibe las peticiones realizadas por el cliente, gestiona las mismas y manda a construir la ClientPage.

- **Client:** Contiene todos los archivos específicos de PHP del lado de CartoClient y permite la interacción entre la Index.php y la CartoClient.

- **CartoClient.:** Integra y recoge todos los datos y funciones realizadas por cada una de las .js que intervienen en el Caso de Uso, y se definen una serie de variables globales que van a ser utilizadas por la aplicación.

- **PluginManager:** Clase que se utiliza para gestionar la base de plugins.

- **ClientPlugin:** Contiene las interfaces necesarias para los plugins del lado del cliente.

- **ServerPlugin:** Esta clase proporciona la base de herramientas para el desarrollo de plugins.

- **ServerContex:** Es la contenedora de la información común que ha de ser utilizada por la parte cliente y la servidora, empleando la información seleccionada como un objeto para un fácil manejo de los datos.

- **MapObj:** Es donde se definen los métodos, funciones, además del lenguaje para el intercambio de datos con el servidor de  mapa (mapa MapServer).

- **CwSerializable**: Se encarga de serializar todas aquellas clases que pueden ser serializadas, con el objetivo de transferir objetos a través de SOAP, permitiendo la comunicación entre el Client y el Server del plugin.

- **AJAXHelper:** Tiene como propósito enviar las respuestas de los plugins “AJAX”, para alimentar a los plugins que responden a las peticiones del usuario.

- **AJAXMonitor:** Se encarga de gestionar el pedido y respuesta a las peticiones del usuario por AJAX.

### 4.4.1. Patrones de diseño

"Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se han demostrado que funcionan. No son fáciles de entender, pero una vez entendido su funcionamiento, los diseños son más flexibles, modulares y reutilizables." (Gracia, 2005)

Para la realización de estos diagramas se aplicaron patrones de diseño los siguientes patrones:

**- Patrones Generales de *Software* para Asignación de Responsabilidades (GRASP)**

Para la realización del diseño del módulo de monitoreo se tienen en cuenta los patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se muestra una selección de estos patrones los cuales serán utilizados durante la realización del diseño de la aplicación.

**Experto:** es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa que siempre se debe asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para llevar a cabo la funcionalidad. Este patrón se pone de manifiesto entre otras en las clases MonitorRequest.php y MonitorResult.php.

**Creador:** este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del mismo es encontrar un creador que necesite conectarse al objeto creado en alguna situación, eligiéndolo como el creador. Se favorece el bajo acoplamiento. Se ve representado en la relación entre las clases Xmpp_PubSub_Standard.php, Xmpp_Helper_Build.php y la clase Xmpp_PubSub_Extension_Hook.

**Bajo Acoplamiento:** el patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados

negativos que puede producir un acoplamiento alto. No soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. El mismo no se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección, al asignar una responsabilidad.

**Alta Cohesión:** Sigue el principio de que cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. A través de este patrón se simplifica el mantenimiento y las mejoras del funcionamiento del sistema. Las clases que están sobrecargadas de métodos poseen una alta cohesión por lo que se recomienda para un buen diseño la creación de los paquetes de servicio o clases agrupadas por funcionalidades que son fácilmente reutilizables.

Estos dos patrones están muy relacionados y se ponen de manifiesto entre las clases ClientPlugin.php y ServerPlugin.php.

- **Patrones GOF.**

En búsqueda de un diseño claro, flexible y reutilizable también se utilizan patrones de diseño **GOF** (*Gang-Of-Four*). Estos se clasifican según su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos, de los mismos fue seleccionado uno para su uso:

**Observador:** este patrón proporciona un modo de acoplar débilmente los objetos en términos de la comunicación. Los emisores conocen a los suscriptores sólo a través de una interfaz, y los suscriptores pueden registrarse o darse de baja de los emisores dinámicamente. Se basa en el Polimorfismo, y proporciona variaciones protegidas en cuanto a que protege al emisor del conocimiento de la clase específica de objetos, y número de objetos, con los que se comunica cuando genera un evento. Este patrón se evidencia cuando se adiciona un observador a cada uno de los plugins que gestionan información geográfica para que este desencadene un conjunto de acciones ante la ejecución de alguno de ellos.

***Singlenton*** (Instancia única): En el diseño de clases es necesario aplicar la solución del patrón *Singlenton* que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. Este patrón se evidencia en el objeto getMapObject de la clase ServerContext.php y en el getInstance en la clase ServerMonitor.

***Command*** (Acción): Este patrón se pone de manifiesto en Ajaxhelper.js de comunicación con el servidor web.

### 4.5. Modelo de despliegue

El Modelo de despliegue se encarga de capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. Consiste en uno o más nodos, dispositivos y conectores, estos últimos estarán ubicados entre nodos y dispositivos. A continuación se propone la distribución física de los elementos que conforman el módulo:

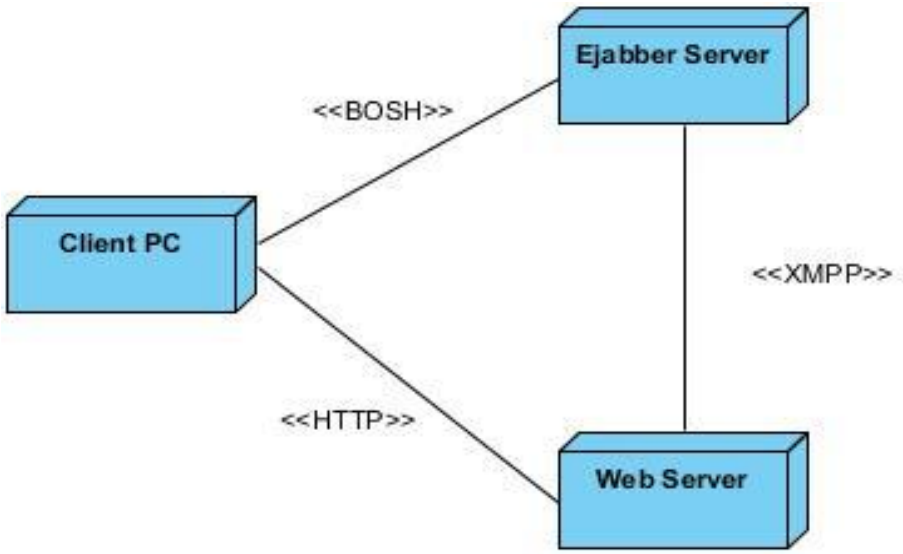



**Figura. 10: Diagrama de Despliegue.**

## 4.6. Validación y pruebas.

**Prueba de Caja Blanca:**

"La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba". (Pressman, 2002) La prueba de del camino básico es una técnica de prueba de caja blanca que permite obtener una medida de la complejidad lógica de un diseño procedimental. Esta técnica se basa en realizar un minucioso análisis de los detalles procedimentales de un determinado código, por lo que se hace necesario conocer la lógica del sistema. Con esta técnica se analiza la lógica interna del programa sin considerar los aspectos de rendimiento. Para llevar a cabo esta técnica se siguen un conjunto de pasos:

- Representar el programa en un grafo de flujo.
- Calcular la complejidad ciclomática.
- Determinar el conjunto básico de caminos independientes.
- Derivar los casos de prueba que fuerzan la ejecución de cada camino.

El siguiente fragmento de código corresponde al caso de uso Publicar información en nodo Pubsub el cual se encarga publicar en el servidor XMPP los datos recibidos:

```php
public function publicar($request){
    $object = Xmpp_PubSub_Standard::getInstance();

    $config = array(
                    'user'      =>  'celia',
                    'pass'      =>  'a',
                    'port'      =>  5222,
                    'host'      =>  '10.54.12.21',
                    'domain'    =>  'mydomain.test',
                    'logLevel'  =>  4,
                    'service' => 'pubsub.mydomain.test');
    $object->init($config);

    $data = json_encode(array('idplugin' => uniqid(), 'accion' => 'obtcoordenada',
    'estado' => 1, 'valor' => array('latitud' => '-12.4457', 'longitud' => '23.45678
    )));
    $hora = date("H:i:s");
    $fecha = date("Y-m-d");

    $item = "<ancla xmlns='http://genesig.cu/'><data>{$data}</data><fecha>{$fecha}
    </fecha><hora>{$hora}</hora></ancla>";
    $object->publishItem('priolus', $item);
    $json = array('success' => true);
    header('Content-Type: application/json');
    echo json_encode($json);
    die();
}
```

**Figura. 11: Fragmento de código perteneciente al caso de uso Publicar información en nodo Pubsub.**


**Figura. 12: Grafo de flujo.**

Calculando la complejidad ciclomática del grafo en cuestión:

La complejidad ciclomática, V (G), de un grafo de flujo G se define como V (G) =P+ 1 donde P es el número de nodos predicado contenidos en el grafo de flujo G.

V (G) = 0 +1

V (G) = 1

El resultado obtenido luego de calcular la complejidad ciclomática indica la cantidad de casos de prueba que son necesarios para realizar el caso de uso.

**Tabla 4: Caso de pruebas aplicando la técnica de caja blanca.**

| | |
|---|---|
| **Caso de Uso** | Publicar información en nodo Pubsub. |
| **Caso de prueba** | 1 |
| **Entrada** | La función ha sido invocada a partir de una petición realizada desde el cliente. |
| **Salida** | Se publica en el nodo la información. |

## 4.7. Conclusiones parciales.

En este capítulo se realizan los diagramas relacionados con el diseño del módulo para los casos de uso identificados en el sistema. Se exponen los patrones de arquitectura que guiarán el diseño de la aplicación, los cuales se basan fundamentalmente en la estructura del *framework* de desarrollo, posibilitando la flexibilidad y la fácil personalización de la arquitectura del sistema además de los propios del módulo. Además se presenta como queda el sistema expresado en componentes de implementación y las pruebas realizadas al mismo, las que demuestran que la aplicación cumple con las funcionalidades previstas.

## Conclusiones

Partiendo del incremento de la gestión de la información geográfica sobre las personalizaciones sobre GeneSIG fue necesario construir un módulo que automatizara su monitoreo en tiempo real. Con el desarrollo del módulo propuesto en el presente trabajo de diploma se concluye lo siguiente:

- Se logró visualizar en tiempo real la gestión de la información geográfica sobre personalizaciones de GeneSIG.
- Fue mejorado el rendimiento del proceso del monitoreo de la gestión de la información geográfica sobre GeneSIG.
- El módulo desarrollado permite que en cualquier personalización de GeneSIG sea monitoreada la información geográfica que se gestione.
- El diseño del módulo permite que sea utilizado otro protocolo para la visualización en tiempo real.

Por todo lo antes mencionado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

## Recomendaciones

Una vez vencido el objetivo de la investigación, y teniendo en cuenta las experiencias obtenidas, la autora recomienda:

- Integrar el módulo desarrollado en personalizaciones de GeneSIG que requieran de una visualización en tiempo real de su información.
- Utilizar la tecnología WebSocket en el proceso de visualización en tiempo real de la información geográfica gestionada en GeneSIG.
- Gestionar el proceso de sincronización de la gestión de la información geográfica que esta siento monitoreada.

## Referencia Bibliográfica

**AG. 1997.** *Analisis de sistemas de producción animal - Tomo 1: Las bases conceptuales.* 1997.

**Cruz, Liester y Dangel, Eddy. 2008.** *Modelación del subsistema de administración y gestión de información geológica del Balance Nacional de Recursos y Reservas de Minerales Sólidos en la Oficina Nacional de Recursos Minerales.* 2008.

**Ejabberd Development Team. 2011.** *Ejabber 2.1.10. Installation and Operation Guide.* 2011.

**Ford, Andrew. 2008.** *Apache 2. Pocket reference.* s.l. : O'Reilly Media, Inc, 2008.

**Fowler, Martin y Scott, Kendall. 1999.** *Uml Gota a Gota. Mexico.* México : Addison Wesley Longman, 1999.

**Gallego, G. J. 2011.** Ingeniería de Requerimientos. [En línea] 2011. //www.scribd.com.

**García, José Carlos Díaz. 2008.** *Estudio del protocolo XMPP de mensajería instantánea de sus antecdentes y de sus aplicaciones civiles y militares.* Madrid : s.n., 2008.

**Giraldo, Luis y Zapata, Yuliana. 2005.** *Herramientas de desarrollo de ingeniería de SW para LINUX. [Presentación].* 2005.

**Gracia, Joaquin. 2005.** Ingeniero de Software. [En línea] 2005. http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php..

**Graf, Esteban. 2004.** Estándar X3.12-1970 (ANSI), Estándar 2382/V, VI (ISO) Vocabulary for Information Processing. *ECOLOGÍA AGRARIA: El abordaje de la realidad a través del enfoque de sistemas.* [En línea] 2004. http://www.fagro.edu.uy/~ambiental/ecologia/Bibliografia/Unidad%201%20-%20Ambiente,%20agricultura%20y%20agronom%EDa/1.2%20Graf%202004%20El%20abordaje%20de%20la%20realidad%20a%20trav%E9s%20del%20Enfoque%20de%20Sistemas.pdf.

**Grupo 97 Catedra Unadista. 2011.** Herramientas Sincrónicas y Asincrónicas. [En línea] mayo de 2011. http://herramientasincronicayasincronica.blogspot.com/.


**Gutiérrez, Ángel. 2009.** IPN, UPIICSA. *Funamentos dela computación.* [En línea] 2009. http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_terminados/PolilibroFC/Unidad_II/Unidad%20II_5.htm.

**ICDE. 2013.** Infraestructura Colombiana de Datos Espaciales. [En línea] 2013. http://www.icde.org.co/web/guest/wiki/-/wiki/Wiki%20de%20la%20ICDE/Objetos+Geogr%C3%A1ficos.

**IETF.** The Internet Engineering Task Force (IETF). *RFC 6455: The WebSocket Protocol.* [En línea] http://datatracker.ietf.org/doc/rfc6455/.

**IGAC. 1998.** *Fundamentos de Sistemas de Información Geográfica.* Bogota : s.n., 1998.

**Isode Team. 2011 .** Isode. *XMPP PubSub.* [En línea] 2011 . http://www.isode.com/.

**IT Business Edge Network. 2013.** Webopedia. [En línea] 2013. http://www.webopedia.com/TERM/P/push.html.

**—. 2013.** Webopedia. [En línea] 2013. http://www.webopedia.com/TERM/P/polling.html.

**—. 2013.** Webopedia. [En línea] 2013. http://www.webopedia.com/TERM/B/browser.html.

**Juan, Alberto José. 2009.** *Uso de XMPP para el transporte de información cooperativa.* Cataluya : s.n., 2009.

**Kicillof, Nicolás; Reynoso, Carlos. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* BUENOS AIRES : s.n., 2004.

**Larman, C. 1999.** *UML y PATRONES.* México : s.n., 1999.

**Loreto, S. y Wilkins, G. 2011.** *Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP.* 2011.

**Martínez Ledea, Lilianne y Utria, Dianet. 2011.** *SIGRutas Modelo de Diseño v2.0.* 2011.

**Mateu, Carles. 2007.** *Desarrollo de aplicaciones web.* Barcelona : Eureca Media, SL, 2007. ISBN: 84-9788-118-4..

**Membrides, Antonio. 2010.** *Manual de GeneSIG.* Habana : s.n., 2010.

**Moffitt, Jack. 2010.** *XMPP Programming with JavaScript and jQuery.* Indianapolis : Wiley Publishing, Inc., 2010.

# Referencia bibliográfica


**Montesinos, Miguel y Gaspar, Jorge. 2007.** *Panorama actual del ecosistema del software libre para SIG.* 2007.

**Olaya, Victor. 2010.** *Sistemas de Información Geografica.* 2010.

**Oracle Corporation. 2013.** NetBeansIDE. [En línea] 2013. http://netbeans.org/community/releases/70/..

**Ordinas, José María Barceló, y otros. 2004.** *Redes de computadores.* Barcelona : s.n., 2004.

**Pantoja, Yoenis. 2010.** Manual de GeneSIG. [aut. libro] Antonio Membrides. Habana : s.n., 2010.

**Paterson, Ian, y otros. 2010.** *Bidirectional-streams Over Synchronous HTTP (BOSH).* s.l. : XMPP Standards Foundation, 2010.

**Pressman, Roger. 2002.** *Ingeniería del Software: Un enfoque práctico.* España : McGraw-Hill Companies, 2002. 5ta Edición.

**Ramírez, Carlos Enrique . 2012.** *Introducción a Ext JS [Presentación].* Habana : s.n., 2012.

**Rumbaugh, James , Jacobson, Ivar y Booch, Grady . 1999.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* I Addison Wesley Iberoamericana : s.n., 1999.

**Sæther, Stig, y otros. 2002.** *Manual de PHP.* s.l. : Rafael Martínez, 2002.

**Saint-Andre, Peter , Smith, Kevin y Tronçon, Remko . 2009.** *XMPP: The Definitive Guide.* s.l. : O'Reilly Media., 2009.

**Taylor, David Ruxton Fraser. 1991.** *Geographic information systems: The microcomputer and modern cartography.* Oxford : UK: Pergamon Press, 1991.

**UPV. 2006.** Unidad Docente de Redes de Computación. [En línea] 2006. http://www.redes.upv.es/oir/trasp/T1_4p_CAS.pdf.

**Visual Paradigm International. 2013.** Visual Paradigm. [En línea] 2013. http://www.visual-paradigm.com/product/vpuml/.

# Anexos

## Anexo 1. Descripciones de Casos de Uso.

**Tabla 5: Descripción textual del caso de uso Crear nodo Pubsub.**

| | |
|---|---|
| **Caso de Uso** | Crear nodo Pubsub |
| **Actores** | Monitor |
| **Propósito** | Este caso de uso se realiza con el objetivo de crear el nodo Pubsub servidor XMPP por donde se publicará la información. |
| **Resumen** | El caso de uso inicia cuando el módulo de monitoreo desea crear el nodo donde se publicará la información, para ello envían los datos necesarios. El caso de uso termina con la creación del nodo Pubsub perteneciente al módulo de monitoreo. |
| **Precondiciones** | Debe haberse ejecutado el caso de uso Conectar. |
| **Referencias** | RF1,RF3 |
| **Prioridad** | Crítico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El caso de uso inicia cuando el usuario accede al SIG. | 1.1 El sistema establece la conexión con el servidor XMPP.<br>1.2 El sistema envía los datos de la estación de trabajo para realizar la auto-conexión en el servidor XMPP. |
| | 2. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario y guarda los datos. |

| *Prototipo de Interfaz* | |
|---|---|
| | |
| **Poscondiciones** | El sistema conecta la estación de trabajo como usuario del servidor XMPP |

**Tabla 6: Descripción textual del caso de uso Conectar estación de trabajo.**

| **Caso de Uso** | Conectar estación de trabajo |
|---|---|
| **Actores** | Usuario |
| **Propósito** | Este caso de uso se realiza con el objetivo de conectar a la estación de trabajo como usuario del servidor XMPP. |
| **Resumen** | El caso de uso inicia cuando el usuario accede al SIG y se requiere conectarse al servido XMPP, por lo que se realiza un proceso de autenticación automático por parte del sistema. Una vez realizado este proceso, se conecta la estación de trabajo al servidor XMPP, finalizando así el caso de uso. |
| **Precondiciones** | El usuario debe de encontrarse registrado en el servidor XMPP. |
| **Referencias** | RF1,RF3 |
| **Prioridad** | Crítico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El caso de uso inicia cuando el usuario accede al SIG. | 1.1 El sistema establece la conexión con el servidor XMPP.<br>1.2 El sistema envía los datos de la estación de trabajo para realizar la auto-conexión en el servidor XMPP. |

| | |
|---|---|
| | 2. El caso de uso termina cuando el sistema procesa la información y conecta a la estación de trabajo como usuario del servidor XMPP. |
| ***Prototipo de Interfaz*** | |
| | |
| **Poscondiciones** | El sistema conecta la estación de trabajo como usuario del servidor XMPP |

**Tabla 7: Descripción textual del caso de uso Suscribir a nodo Pubsub.**

| | |
|---|---|
| **Caso de Uso** | Suscribirse a nodo Pubsub |
| **Actores** | Usuario |
| **Propósito** | Este caso de uso se realiza con el objetivo suscribirse al nodo Pubsub donde se publicará información |
| **Resumen** | El caso de uso inicia cuando es necesario que la estación de trabajo se suscriba al nodo Pubsub de publicación de información. |
| **Precondiciones** | Debe haberse ejecutado el Caso de uso Conectar estación de trabajo. |
| **Referencias** | RF7 |
| **Prioridad** | Crítico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| | 1.1 El sistema entra los datos del nodo al que desea conectarse. |

| | |
|---|---|
| | 1.2 El sistema envía los datos al servidor XMPP. |
| | 2. El caso de uso termina cuando el sistema procesa la información y suscribe a la estación de trabajo. |
| ***Prototipo de Interfaz*** | |
| | |
| **Poscondiciones** | El sistema suscribe a la estación de trabajo al nodo de publicación. |

**Tabla 8: Descripción textual del caso de uso Conectar.**

| | |
|---|---|
| **Caso de Uso** | Conectar |
| **Actores** | Monitor |
| **Propósito** | Este caso de uso se realiza con el objetivo de conectar al módulo monitor como usuario del servidor XMPP. |
| **Resumen** | El caso de uso inicia cuando módulo de monitoreo se ejecuta y se requiere conectarse al servido XMPP, por lo que se realiza un proceso de autenticación automático por parte del sistema. Una vez realizado este proceso, se conecta al servidor XMPP, finalizando así el caso de uso. |
| **Precondiciones** | El usuario debe de encontrarse registrado en el servidor XMPP. |
| **Referencias** | RF2 |
| **Prioridad** | Crítico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |

| | |
|---|---|
| 1. El caso de uso inicia cuando el usuario accede al SIG. | 1.1 El sistema establece la conexión con el servidor XMPP.<br>1.2 El sistema envía los datos de la estación de trabajo para realizar la auto-conexión en el servidor XMPP. |
| | 2. El caso de uso termina cuando el sistema procesa la información y conecta al módulo de monitoreo como usuario del servidor XMPP. |
| ***Prototipo de Interfaz*** | |
| | |
| **Poscondiciones** | El sistema conecta la estación de trabajo como usuario del servidor XMPP |

## Anexo 2. Diagramas de Clases del Diseño.

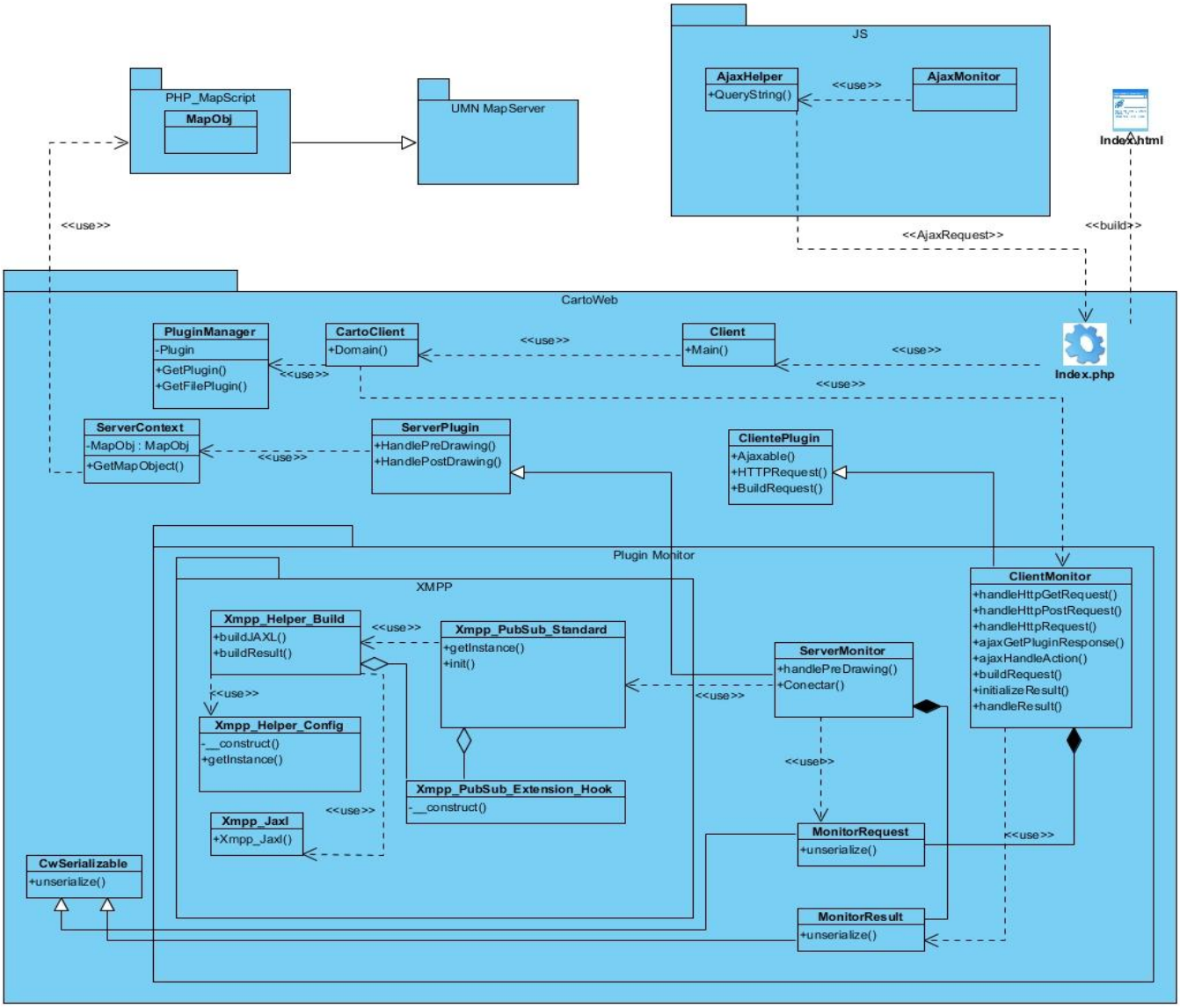



**Figura. 13: Diagrama de clases del Diseño del caso de uso conectar.**

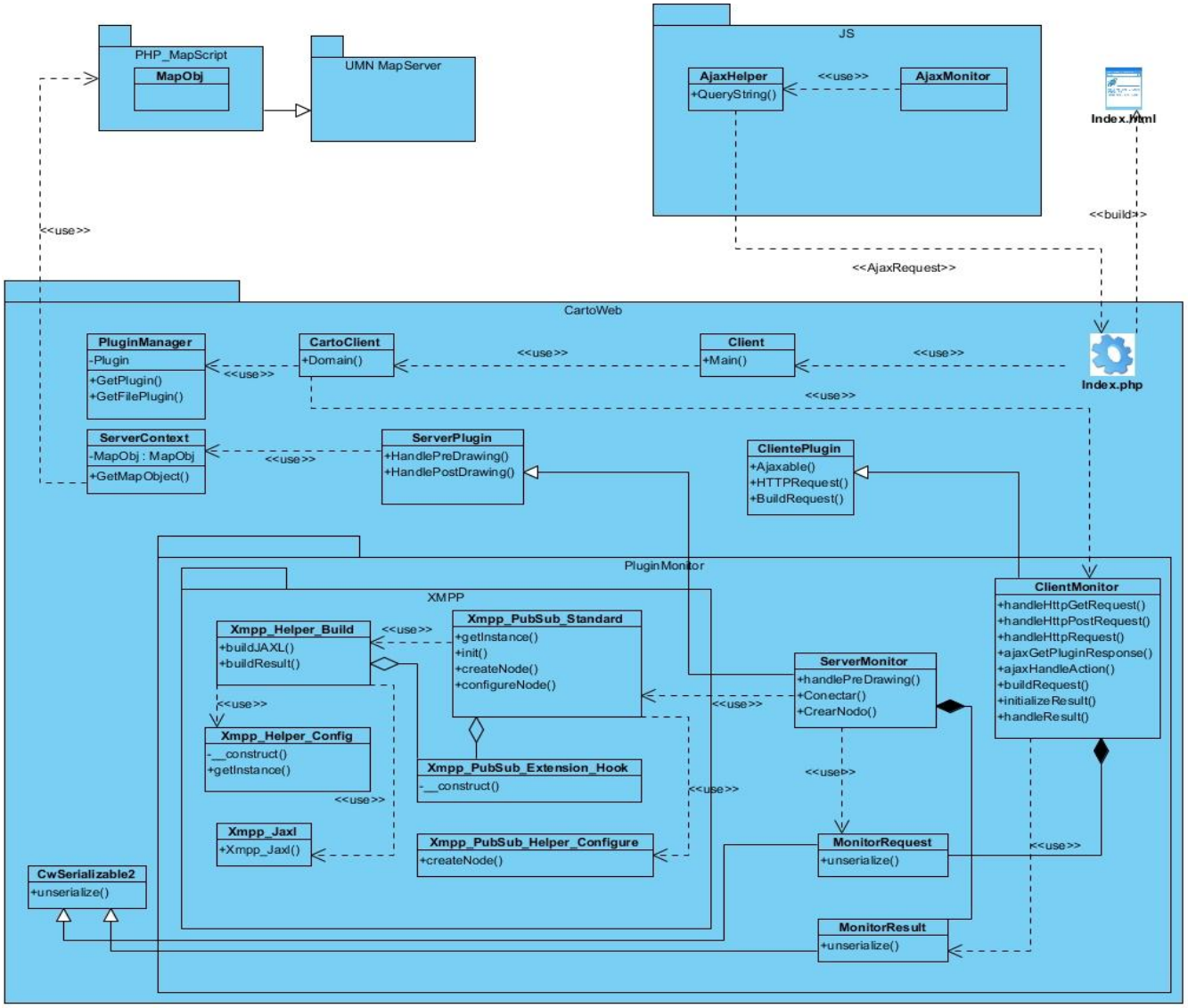



**Figura. 14: Diagrama de clases del Diseño del caso de uso crear nodo Pubsub.**

## Glosario de términos

**A**

**-AJAX:** JavaScript asíncrono y XML.

**-API:** Interfaz de programación de aplicaciones.

**-AUP:** Proceso Unificado Ágil.

**B**

**-BOSH:** *Bidirectional-streams Over Synchronous* HTTP.

**C**

**-CASE:** Ingeniería de *Software* Asistida por Computadora.

**E**

**-XP:** *Extreme Programing.*

**G**

**-GeoCuba:** Grupo empresarial cubano dedicado al desarrollo cartográfico.

**-GRASP:** Patrones Generales de *Software* para Asignación de Responsabilidades.

**-GOF:** *Gang-Of-Four.*

**H**

**-HTTP:** Protocolo de Transferencia de Hipertexto.

**I**

**-IDE:** Entorno de Desarrollo Integrado.

**-IETF:** Organización internacional abierta de normalización.

**J**

**-JS:** JavaScript.

**L**

**- LPS:** Línea de Producto de *Software.*

**P**

**-PHP:** *Hypertext Preprocessor.*

**-Pubsub:** Publicación-Suscripción.

**S**

**-SASL:** *Framework* para autenticación y autorización en protocolos.

**-SIG:** Sistema de Información Geográfico.

**T**

**-TIC:** Tecnologías de la Información y las Comunicaciones.

**-TCP:** Protocolo de Control de Transmisión.

**-TLS:** Protocolo criptográfico que proporciona comunicaciones seguras por una red.

**U**

**- UCI:** Universidad de la Ciencias Informáticas.

**-UCID:** Unidad de Compatibilización e Integración para la Defensa.

**-UML:** Lenguaje Unificado de Modelado.

**X**

**-XML:** Lenguaje de Marcado Extensible.

**-XMPP:** Protocolo Extensible de Mensajería y Comunicación de Presencia.