UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS
FACULTAD 6

Desarrollo de un plugin que permita el consumo de Servicios de Procesamiento Web al Quantum GIS

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

Autor: José Angel Arias Fonseca

Tutor: Ing. Yoandry Lazo Nodarse

Ciudad de la Habana, 2011

“Año 53 dela Revolución”

DEDICATORIA

A Mamin porque siempre me ha cuidado y me ha dado la sabiduría, la inteligencia y el amor necesario que me han permitido ser quien soy.

A mis padres por ser mis mejores amigos y apoyarme en los momentos más difíciles, siempre demostrándome su amor, apoyo y confianza.

A mi tía Severe y a mi Abuela Virginia que siempre estuvieron pendientes de la evolución del trabajo y dándome todo su apoyo.

A toda mi familia que en un momento u otro han sabido brindarme su amor y estar a mi lado.

AGRADECIMIENTOS

A Mamin y mi abuela Virginia por la educación que me dieron.

A mi Mamá, porque es lo que más quiero en la vida y por comprenderme siempre.

A mi Papá, por quererme tanto y siempre confiar en mí.

A mi novia Ivania, por haberme ayudado infinitamente, por estar a mi lado compartiendo buenos y malos.

A mis hermanos Luis Daniel, Luis Adolfo y Alejandro, que son mi razón de ser.

A mi Tía Severe, por guiarme en todo momento y ser mi segunda madre.

A mi Tío Alfredito, y a Isi por ser un ejemplo de superación para mí y por el apoyo que me brindaron cuando más lo necesité.

A mi familia de Macabí por siempre confiar en mí y estar a mi lado en especial a mis primos Oski y Anne a mis tías Dulce, Iyo y Ana y a mi tío Oscar.

A mi abuela Sule a mi tía Grisel a mis primos Jorgito y Ernesto a mi querida Vilma porque siempre han estado ahí para mí.

A mi padrastro Yoni por siempre cuidar lo que más quiero.

A mis suegros Pancha y Eduardo, por el apoyo que me han brindado y por quererme como a un hijo.

A Noel, Shampo y todos mis amigos de Moa por su amistad incondicional, por apoyarme en todo momento.

A todos mis amigos del 9108, siempre estarán presentes en mi vida, especialmente Aliuska, Lisette, Yasiel, Eduardo, Siomelis, José Carlos, Oslanier, Yudalis, Alain, Eduanis, Pedro, Carlos, Leiber.

A todos los profesores que han contribuido a mi formación profesional.

A todos los que confiaron en mí.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del trabajo titulado: “Desarrollo de un plugin que permita el consumo de Servicios de Procesamiento Web al Quantum GIS” y autorizo a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de Junio del año 2011.

José Angel Arias Fonseca

_____

Ing. Yoandry Lazo Nodarse.

_____

RESUMEN

Este documento es el resultado de la investigación realizada para obtener el título de Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI), realizado en el área de Sistemas de Información Geográfica durante el periodo de diciembre de 2010 a junio de 2011. El proyecto SIG-Desktop se encuentra inmerso en el desarrollo de nuevas funcionalidades para el Quantum GIS (QGIS) para lo cual se han desarrollado varios plugins entre los que se encuentra **qWPS**. Para el desarrollo del mismo se abordan diferentes temáticas con vista a lograr buenos resultados en la implementación del sistema, el cual permitirá ejecutar procesos a través de Web Processing Service (WPS), el protocolo promovido por el Open Geospatial Consortium (OGC). El software, se elaboró en Python en conjunto con Qt4 para el desarrollo de las interfaces de usuario, siguiendo un conjunto de estándares. Una vez incorporado al QGIS, permitirá que éste pueda conectarse a un servidor habilitado para brindar servicios WPS, ver los procesos disponibles en él, ejecutarlos de forma remota y visualizar el resultado.

PALABRAS CLAVES

Plugin, QGIS, SIG, WPS

ABSTRACT

This document is the result of research carried out to obtain the engineering degree in Computer Science from the University of Informatics Sciences (UCI) conducted in the area of Geographic Information Systems during the period December 2010 to June 2011. The project SIG-Desktop is immersed in the development of new capabilities for Quantum GIS (QGIS) for which have developed several plugins including qWPS. To develop the project will be addressing various topics in order to achieve good results in the implementation of the system which allow to run processes through Web Processing Service (WPS), protocol promoted by the Open Geospatial Consortium (OGC). The software will be developed in Python in conjunction with Qt4 for the development of user interfaces, using a set of standards. Once incorporated into QGIS, allow to connect to a server that is enabled to provide services WPS, see the processes available to remotely execute and display the result.

Contenido

Índice de figuras

Índice de tablas

INTRODUCCIÓN

Desde el surgimiento del hombre como especie le fue imprescindible desplazarse y recorrer grandes distancias de forma nómada con vista a conseguir refugios y comida, en los inicios de la civilización. Con el decursar de los siglos surgió el comercio, lo que trajo como consecuencia largos viajes, cuanto más era capaz de viajar el comerciante; mayores serían sus beneficios. Estos viajes dieron origen a los mapas. El primer mapa del que se tiene conocimiento data del 2300AC conocido como la Estela de Arcilla Sumeria. Estos en un primer momento eran artefactos rudimentarios grabados en la pared o grabados en las piedras, y cada vez se hacían más importantes, para saber dónde está ubicada una persona, para ubicar países, ciudades, ríos, montañas, para el servicio meteorológico, para el desarrollo de la economía, por mencionar algunos. Hoy día no han perdido esa importancia, pero gracias a los avances tecnológicos se cuenta con distintos tipos y con un alto grado de precisión. Estos se encuentran en los dispositivos más pequeños que acompañan a las personas a diario brindando servicios invaluables, para el desarrollo de actividades de forma más corta y rápida.

A medida que crece el flujo de información en el mundo, los países desarrollados y las grandes empresas aumentan la necesidad de contar con herramientas informáticas más precisas y funcionales, buscando mejorar la calidad de vida de las personas que intervienen en su utilización diaria. Las grandes trasnacionales del software se encuentran inmersas en suplirlas necesidades de las personas que necesitan y desean saber su localización en cada instante. Son pocos los sistemas de licencia GPL (Licencia Publica General) que han emprendido el camino de dar solución a esta problemática aunque sí existen algunas importantes. Sin embargo es imprescindible, con el auge que han alcanzado en este momento, que se le preste la debida atención al reciente fenómeno de los servicios basados en localización, pues este promete introducir todo tipo de nuevas funcionalidades y características, basadas en una combinación de mapas y análisis.

Es una necesidad para las personas en la actualidad saber qué rumbo llevan y cómo recorrerlo por la menor distancia, buscando optimizar el tiempo de la mejor forma posible. Como una solución a esta problemática surgieron los Sistemas de Información Geográfica (SIG) que son una colección de software que permite crear, visualizar, consultar y analizar datos geoespaciales; lo mismo teléfonos, carros, empresas y dispositivos con esos fines le están mostrando a los usuarios la localización geográfica

precisa. A pesar de parecer aplicaciones sencillas, por lo fácil que resulta su manipulación, no son solamente una simple aplicación que muestra un mapa, sino que básicamente todo el tiempo están haciendo análisis geoespacial y listando los resultados para la satisfacción de los usuarios finales.

Cuba no está exenta al desarrollo que se está alcanzando a nivel mundial en cuanto al tema de los datos geoespaciales y del posicionamiento global. En estos últimos años ha realizado grandes esfuerzos en pos de lograr un gran avance en el desarrollo de software basándose principalmente en el desarrollo con tecnologías libres. Paulatinamente se ha demostrado que las herramientas de software libre tienen las mismas o más potencialidades que las producidas por las más grandes empresas propietarias. Muchas empresas del área de la geología en Cuba, que se encuentran en la etapa de perfeccionamiento empresarial, se han visto en la necesidad de incrementar su potencialidad adquiriendo disímiles productos, como son los GIS, que le hicieran, de forma general, más sencillos y rápidos los procesos logrando así más eficiencia en la producción.

La Universidad de las Ciencias Informáticas (UCI), es una universidad surgida al calor de la batalla de ideas con el objetivo de vincular a sus estudiantes a proyectos productivos. Esta universidad cuenta con 10 facultades por las cuales se reparten exactamente 16 Centros de Desarrollo en los cuales está distribuida una gran fuerza de trabajo. El Centro de Desarrollo de Geoinformática y Señales Digitales (GEYSED), más específico, el proyecto SIG-Desktop, se encuentra inmerso en la personalización de la herramienta libre Quantum GIS (QGIS), para lo cual es preciso el desarrollo de nuevas funcionalidades.

De acuerdo con lo planteado anteriormente y las facilidades que brindan los SIG, es necesario para aumentar sus funcionalidades la implementación del consumo de Servicios de Procesamiento Web (WPS); la principal tarea está encaminada a resolver la siguiente **situación problemática:** el proyecto SIG-Desktop del centro de desarrollo GEYSED se encuentra inmerso en la personalización del QGIS, para lo cual se han detectado en dicho proyecto algunas de las principales deficiencias de este potente software; una de ellas es que no permite el consumo de servicios de procesamiento web lo cual disminuye su potencialidad, escalabilidad y usabilidad. La personalización de una herramienta como ésta favorecería el desarrollo de muchas empresas que existen en Cuba y que están sumergidas en el tema de la ubicación de los principales recursos minerales y naturales, a las cuales les podría resultar muy útil. Por este motivo en el proyecto de SIG-Desktop se está dando solución a dichos inconvenientes, de donde se

deriva el siguiente **problema a resolver:** el software QGIS no permite el consumo de *Web Processing Service*[1] *(WPS)*, lo que reduce su nivel de escalabilidad.

El **objeto de estudio** lo constituye el estándar WPS, enmarcándose en el **campo de acción** el estándar WPS para el Sistema de Información Geográfica QGIS.

Se persigue como **objetivo general** desarrollar un plugin a QGIS para el consumo del WPS. Para lograr el objetivo general se trazaron una serie de tareas base:

1. Caracterizar la tecnología WPS y otros conceptos relacionados con su funcionamiento.
2. Evaluar el estado actual de aplicación de la tecnología WPS.
3. Diseñar la propuesta de solución.
4. Implementar las nuevas funcionalidades en QGIS.
5. Evaluar la solución.

Como **idea a defender** se definió el desarrollo de un plugin, para que el QGIS consuma WPS, pues este permitirá una mayor escalabilidad al sistema.

Para darle cumplimiento a las tareas propuestas se utilizaron los siguientes métodos de investigación.

Teóricos:

- **Histórico-Lógico:** este método se usó con el fin de recopilar la información que se posee hasta el momento sobre los WPS y resumir los aspectos fundamentales de su evolución para la investigación en curso.
- **Analítico-Sintético:** este método se usó para analizar y estudiar las teorías recopiladas en la bibliografía y extraer los elementos más importantes que se relacionan con el objeto de estudio.

[1] *Servicio de procesamiento web*

- **Modelación:** este método se usó para descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. La modelación es justamente el proceso mediante el cual se crean modelos con vista a investigar la realidad, por lo que es el más importante en el proceso de construcción de software.

Empíricos:

- **Observación:** este método se usó para realizar valoraciones y obtener informaciones a partir de lo observado. Esto se manifiesta principalmente cuando se realizan observaciones sobre el funcionamiento de sistemas similares al desarrollado, lo que da una visión de cómo tiene que ser el sistema a realizar en su forma externa.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El desarrollo de este capítulo está dedicado a abordar algunos de los conceptos y aspectos relevantes sobre los SIG, haciendo énfasis principalmente en el QGIS, al que será aplicado el resultado final, así como la descripción del estándar WPS y sus principales características. Además se estarán abordando algunos conceptos fundamentales sobre los plugins en conjunto con sus características. Se caracterizan las metodologías de desarrollo y las principales herramientas que se utilizarán para el desarrollo del producto.

1.2. Sistemas de Información Geográfica

El uso de los SIG ha aumentado de forma exponencial en las últimas tres décadas y como consecuencia, estos sistemas han pasado del total desconocimiento a la práctica cotidiana en el mundo de los negocios, en las universidades y en los organismos gubernamentales, usándose para resolver diversos problemas. Por su importancia varias personalidades e instituciones se han preocupado por proponer algunas definiciones. Un SIG se define como un sistema computacional para la entrada; manejo (almacenamiento y recuperación de información); manipulación, análisis; y representación de datos geográficos (1). Una breve pero relevante definición es: Un SIG es un sistema de información de base informática que permite la captura, modelización, manipulación, recuperación, análisis y presentación de datos referenciados geográficamente. (2)

Desde un punto de vista informático un SIG es un sistema capaz de realizar una gestión completa de datos geográficos referenciados. Por referenciados se entiende que estos datos geográficos o cartográficos tienen coordenadas geográficas reales asociadas, las cuales permiten manejar y hacer análisis con datos reales, como longitudes, perímetros o áreas. Todos estos datos alfanuméricos asociados a las cartografías los gestiona el SIG.

Luego de más de 30 años de desarrollo de los SIG son muchas las funcionalidades que se han desarrollado lo cual no establece que ya está todo hecho y a medida que son más pequeñas y manuables es preciso desprenderse de algunas que ya están un poco atrasadas y seguir guiando a los SIG por el buen camino que hoy día no es más que el de la portabilidad.

1.2.1. Quantum GIS

QGIS es un SIG fácil de usar y Open Source que está disponible para las plataformas de Linux, Mac OSX y Windows. QGIS soporta capas vectoriales, rasters, y los formatos de base de datos como PostGIS. QGIS está licenciado bajo la GNU Public License. El proyecto nació en mayo de 2002 y se ha trabajado duro desde entonces para hacer del software SIG (que tradicionalmente es software comercial caro) una posibilidad viable para cualquiera con un acceso básico a un ordenador personal (3). QGIS está desarrollado utilizando el Qt y C++. Esto hace que QGIS sea rápido y tenga una interfaz de usuario agradable y fácil de usar. Presenta un gran número de funcionalidades, muchas de estas de los SIG tradicionales. La incorporación del lenguaje Python posibilita escribir complementos en Python para QGIS, lo cual deja atrás cualquiera de sus limitaciones, pues resulta relativamente fácil el desarrollo y la incorporación de nuevas funcionalidades.

1.3. Interoperabilidad y Open Geospatial Consortium

En los últimos años se está produciendo una evolución muy notable de los SIG y las tecnologías geoespaciales en general. Antes eran usadas solo por un reducido grupo de científicos y profesionales de la gestión del territorio, pero ahora su uso es mucho más generalizado y abierto. En gran medida se han visto impulsadas por Internet, la eclosión de la llamada Web 2.0 e iniciativas rompedoras como las de Google[2].

Con el cambio se está pasando de un entorno opaco, con software monolítico, formatos propietarios incompatibles y pocos datos digitales a un panorama mucho más integrado en la corriente general de las TICs, orientado a los sistemas distribuidos y servicios web, y con un creciente peso de los estándares y el intercambio.

Para permitir este nuevo sistema, en constante crecimiento, variado y complejo, es imprescindible trabajar en la interoperabilidad, y es ahí donde el Open Geospatial Consortium (OGC) cumple su papel. El OGC es un consorcio internacional constituido por empresas, universidades y organismos gubernamentales que lidera la generación de estándares en el área geoespacial. Dentro de estos estándares destacan los OGC

[2] *Nos referimos a la revolución impulsada por Google Maps y Google Earth en los últimos años y todo el fenómeno imparable de los Mashups, muchos de los cuales tienen como base servicios geográficos. Para una interesante reflexión al respecto puede consultarse Botella Plana, A. (2008).*

Web Services (OWS), un conjunto de servicios web definidos utilizando estándares de Internet no propietarios, particularmente World Wide Web (WWW), Hypertext Transfer Protocol (HTTP), Uniform Resource Locators (URLs), tipos Multipurpose Internet Mail Extensions (MIME) y el lenguaje eXtensible Markup Language (XML). En la Figura 1 se muestra un esquema con los OWS más extendidos y sus operaciones principales.

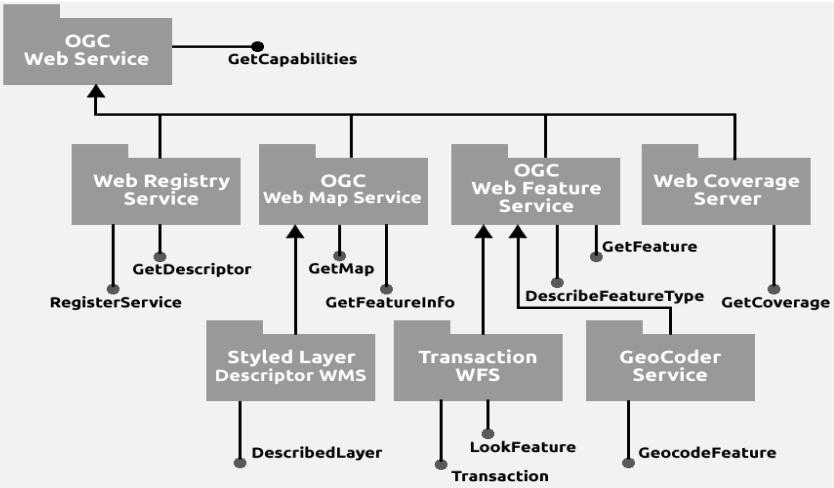


Figura 1. Esquema de los principales estándares del OGC (4).

En esta área se han logrado ya importantes avances como el WMS (Web Map Service) o WFS (Web Feature Service), que han favorecido enormemente la publicación de cartografía y datos a través de Internet por parte de organismos productores. Y recientemente ha sido propuesto como estándar WPS (Web Processing Service), apareciendo así un protocolo para publicar y consumir procesos remotos que puede suponer un paso fundamental en el avance de los SIG[3].

1.4. Estándar Servicios de Procesamiento Web (WPS)

Hay varios recursos para estudiar en qué consiste el estándar WPS, el primero de ellos y más destacado es el documento de especificación v1.0.0 “OpenGIS Web Processing Service OGC 05-007r7” del OGC.

[3]Existen *alternativas propietarias a WPS, pero no son interoperables. P.ej. existe un interesante entorno de ejecución de procesos con ESRI ArcGIS Server, pero que obliga a ejecutar software ESRI tanto en el extremo cliente (desktop, explorer, aplicación web) como en la parte* servidora.

En dicho documento se señala cómo WPS es un interfaz estandarizado de servicio para la publicación de procesos geoespaciales y su descubrimiento y consumo por parte de clientes. Dentro de la categoría de procesos geoespaciales se incluye cualquier algoritmo, cálculo o modelo que opere sobre datos con referencia espacial, desde operaciones simples como un buffer, hasta modelos complejos como cálculo de zonas inundables. (5)

En la Figura 2 se muestra la interfaz de un servicio conforme a WPS donde se muestran las tres operaciones que resultan obligatorias.

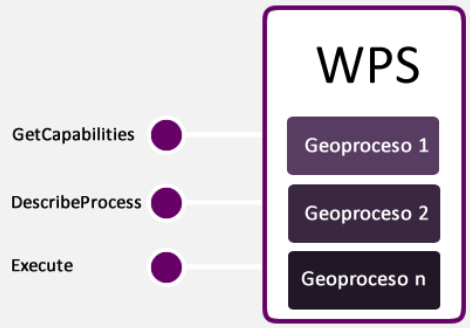


Figura 2. Operaciones de la interfaz WPS (Elaboración propia).

A continuación se describe brevemente cada una de estas operaciones:

- *GetCapabilities:* el cliente solicita al servidor metadatos sobre sus capacidades y éste le devuelve los nombres y descripciones generales de los procesos que aloja.
- *DescribeProcess:* el cliente está interesado en información detallada de un proceso y el servidor responde indicando los parámetros que requiere, sus formatos y los resultados que produce.
- *Execute:* el cliente realiza una petición de ejecución de un proceso, proporcionando los parámetros de entrada y quedando a la espera de los resultados.

Como datos de entrada para un servicio WPS pueden utilizarse datos vectoriales o raster y éstos pueden estar ubicados en local o en la red. El cliente es el responsable de suministrarlos al servicio, tal y como éste los requiera, pero una vez hecho esto, el cliente puede ejecutar la operación sin conocer ningún detalle interno de implementación.

1.4.1. Arquitecturas disponibles para WPS

En esta sección se describen algunas de las posibles arquitecturas que puede presentar un servicio WPS a partir de un conjunto de módulos disponibles en un SIG de escritorio. Se discute una arquitectura centrada en servidor como las usadas en la nueva web 2.0 y una arquitectura donde el servidor es un mero ejecutor de procesos basados en datos remotos. Cada arquitectura necesita de distintos elementos adicionales que no se describen en el propio estándar WPS, que son identificados y para los cuales se proponen soluciones.

- ✓ **ArquitecturaWPS-W2**

Las arquitecturas Web 2.0 se basan en el poder de la interacción con el usuario como fuente de nuevos datos. Estos datos se almacenan en grandes centros de datos de ubicación desconocida, e inaccesibles directamente por el usuario. El usuario debe utilizar los recursos que el servicio le aporta o transferir sus datos al sistema con anterioridad a cualquier acción. Esto simplifica la arquitectura WPS dado que el usuario dispone de un catálogo de capas con las que trabajar que será idéntico sea cual sea la ubicación de dicho usuario. De igual manera, el resultado de una ejecución será almacenado en el propio centro de datos. El sistema dispondrá de mecanismos para la visualización del resultado de la ejecución del proceso y, eventualmente, para la exportación del resultado.

Para la ejecución del proceso, el usuario deberá seleccionar las capas implicadas de entre las disponibles en el catálogo del servicio y posiblemente completar algunos metadatos de la nueva capa generada. Una vez solicitado el inicio del proceso, el servicio lo ejecutará e incluirá el resultado en el catálogo una vez acabado el proceso. Sólo entonces el usuario podrá ver el resultado si lo desea. (6)

En la Figura 3 se muestra qué aspectos hay que tener en cuenta para desarrollar un cliente WPS que utilice la arquitectura WPS-W2.

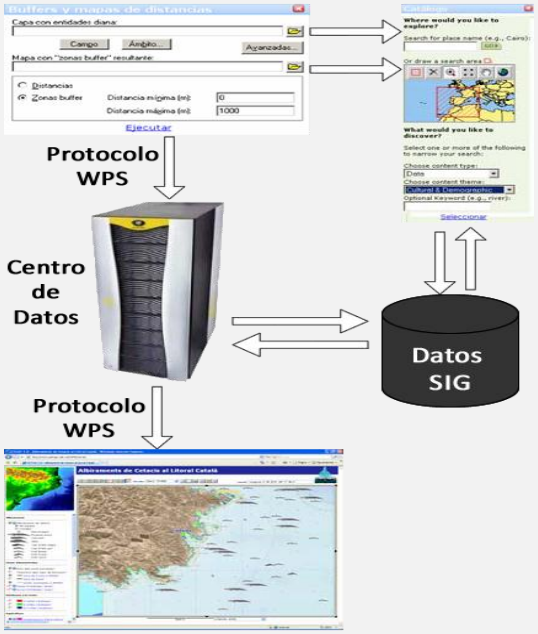


Figura 3. Arquitectura WPS-W2 (6)**.**

✓ **Arquitectura WPS-R**

Para que sea posible el uso de WPS-R es necesario que exista una interfaz de usuario en el lado cliente, un mecanismo para la transmisión de documentos cartográficos (en adelante capas, para simplificar) del cliente al servidor, una comunicación con el servidor para iniciar el proceso (junto con la información de los parámetros de entrada), un mecanismo para informar al usuario del estado del proceso y, finalmente, un sistema para transmitir al usuario el resultado del proceso, que generalmente será una capa pero puede ser una tabla o un solo valor.

A continuación se detalla cada uno de estos pasos. La interfaz de usuario puede ser la misma usada por el programa de escritorio con la salvedad de que debe indicarse qué servidor debe realizar la operación. El mecanismo de transmisión de capas del cliente al servidor puede no ser un tema de fácil solución para la mayoría de SIGs de escritorio dado que una capa puede estar formada por un gran número de archivos interrelacionados, o incluso una estructura de registros en varias tablas de una base de datos. Solamente algunos programas SIG disponen de mecanismos para la transmisión de una capa o un proyecto cartográfico en un solo archivo: ArcInfo dispone del formato de archivos *export* (e00), MiraMon dispone del

formato de archivo MMZ y el reciente Google Earth dispone del formato de archivo KMZ. Este tipo de archivos (todos ellos con capacidad de compresión) son ideales para transmitir capas del cliente al servidor manteniendo la mayoría o toda (según el formato) la calidad de los datos originales (geometría, geodesia, datos espaciales y alfanuméricos, tablas y documentos relacionados, metadatos, simbolización, y otros). En adelante, a estos formatos, se llamarán formatos de intercambio. Cada servidor WPS se basa en un perfil concreto del estándar WPS que establece claramente cómo iniciar la ejecución de un proceso y cómo relacionar las capas para cada proceso disponible. Una vez iniciado el proceso, la aplicación cliente puede consultar el estado de la ejecución al servidor y actualizar una ventana de información al usuario. Si el final de la ejecución da como resultado una capa, antes de dar por finalizado el proceso la aplicación servidora deberá empaquetar el resultado en un formato de intercambio para su entrega. Una vez que la aplicación cliente conozca que el proceso ha terminado deberá solicitar la descarga del archivo de intercambio resultante y, si lo estima oportuno, mostrarlo al usuario final. (6)

En la Figura 4 se muestra qué aspectos hay que tener en cuenta para desarrollar un cliente WPS que utilice la arquitectura WPS-R.

Figura 4. Arquitectura WPS-R (6)**.**

Un punto práctico importante es que en el caso de que el servidor y el cliente se encuentren en el mismo dominio y que el esquema de permisos así lo permita (por ejemplo cuando ambos se encuentren en la misma red de área local), los pasos de compresión de las capas a procesar, así como la descompresión de la capa de salida pueden ser eliminados dado que se supone que el servidor tiene acceso a los mismos datos que el usuario posee y que el cliente puede acceder directamente al producto resultante de la operación ejecutada.

1.4.2. Web Map Service (WMS)

Con vista al consumo de los servicios de procesamiento web para los SIG de escritorio existen dos servidores especializados en brindar este tipo de servicio. Estos, mediante una interfaz o estándar de comunicación, permiten que los SIG que tengan implementado este servicio puedan consumirlos.

El **servicio Web Map Service (WMS)** definido por el OGC, produce mapas de datos referenciados espacialmente de forma dinámica a partir de información geográfica. Este estándar internacional define un mapa como una representación de la información geográfica en forma de un archivo de imagen digital conveniente para la exhibición en una pantalla de ordenador. Los mapas producidos por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG, y opcionalmente como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o WebCGM (Web Computer Graphics Metafile).

El estándar define tres operaciones:

1. Devolver metadatos del nivel de servicio.
2. Devolver un mapa cuyos parámetros geográficos y dimensionales han sido bien definidos.
3. Devolver información de características particulares mostradas en el mapa (opcionales).

Las operaciones WMS pueden ser solicitadas usando un navegador estándar realizando peticiones en la forma de URLs. El contenido de tales URLs depende de la operación solicitada. Concretamente, al solicitar un mapa, la URL indica qué información debe ser mostrada, qué porción de la tierra debe dibujar, el sistema de coordenadas de referencia, y la anchura y la altura de la imagen de salida. Cuando dos o más mapas se producen con los mismos parámetros geográficos y tamaño de salida, los resultados se pueden solapar para producir un mapa compuesto. El uso de formatos de imagen que soportan fondos

transparentes (GIF o PNG) permite que los mapas subyacentes sean visibles. Además, se puede solicitar mapas individuales de diversos servidores.

El servicio WMS permite así la creación de una red de servidores distribuidos de mapas, a partir de los cuales los clientes pueden construir mapas a medida. Las operaciones WMS también pueden ser invocadas usando clientes avanzados SIG, realizando igualmente peticiones en la forma de URLs.

Web Feature Service (WFS)

WFS es un servicio estándar del OGC, que ofrece una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar la imagen que ofrece el servicio WMS o analizar la imagen siguiendo criterios geográficos. Para realizar estas operaciones se utiliza el lenguaje GML que deriva del XML, que es el estándar a través del que se transmiten las órdenes WFS.

WFS no transaccional permite hacer consultas y recuperación de elementos geográficos. Por el contrario WFS-T (Web Feature Service Transactional) permite además la creación, eliminación y actualización de estos elementos geográficos del mapa.

Web Converge Service (WCS)

El Open Geospatial Consortium Web Coverage Service Interface Estándar (SCM) ofrece una interfaz que permite las solicitudes de cobertura geográfica a través de la web, mediante llamadas independientes de la plataforma. Las coberturas son objetos (o imágenes) en un área geográfica, mientras que la interfaz WMS o portales de mapas en línea como Google Maps sólo un cambio de imagen, que los usuarios finales no se pueden editar o analizar espacialmente. Los miembros de OGC definen y mantienen la especificación WCS. GeoServer sirve como la implementación de referencia de la norma.

1.5. Plugin

Un Plugin es una aplicación que incrementa una característica o servicio a un sistema ya existente; y éste se ejecuta por la aplicación principal. A nivel mundial, existen aplicaciones basadas en arquitectura plugin que le brindan la capacidad de agregar funcionalidades nuevas en tiempo de ejecución. Entre sus principales ventajas está que permite que los desarrolladores externos colaboren con la aplicación, de modo que se extiendan sus funcionalidades, posibilita la personalización de la aplicación de manera no

pensada por el autor, o que al menos no tenía la intención de hacer. Los plugin en el QGIS tienen gran importancia, estos le brindan al mismo un gran potencial para la adaptación y el desarrollo de nuevas funcionalidades.

1.5.1. Estructura de los plugins para QGIS

Cada plugin está compuesto por una serie de ficheros y directorios que les dan la estructura, función y comunicación con el QGIS. La Figura 5 muestra la estructura con que deben estar formados los Plugin para poder ser interpretados por el QGIS cuando este se ejecute.

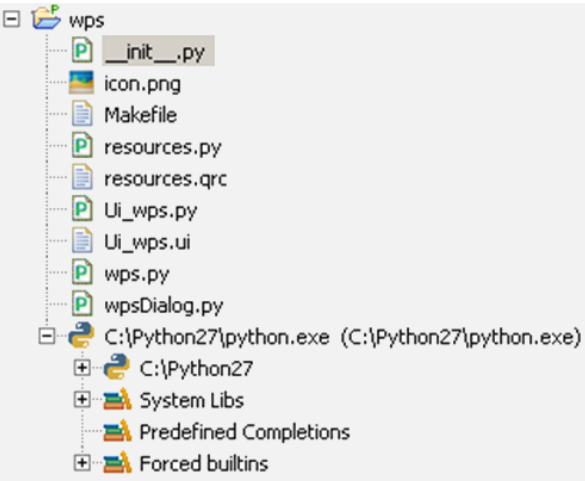


Figura 5. Estructura de los plugins de QGIS (Elaboración propia).

Los Plugin deben ser colocados con dicha estructura en la carpeta que tiene predefinida el QGIS, así éste mediante su intérprete de PyQt4 será capaz de reconocer el mismo e incorporarlo en su menú de Plugins.

1.6. Herramientas para el desarrollo

Para dar solución a la presente propuesta y con el fin de obtener un producto de software factible, eficaz y de alta calidad es preciso definir cuáles son las herramientas que serán de mayor utilidad para la implementación. Las herramientas y tecnologías a utilizar se escogieron luego de un estudio de las principales características, facilidades y restricciones del QGIS y dar solución al problema, con vista a lo que se iba a implementar para obtener el producto deseado.

1.6.1. Lenguaje de programación Python 2.5

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Debido a sus potencialidades y su facilidad de aprendizaje es un lenguaje que ha despuntado en los últimos años. Una de las ventajas de Python es que es un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos lo que lo dota de un alto grado de flexibilidad y portabilidad. El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS) por lo que si no utilizáramos librerías específicas de cada plataforma, el programa desarrollado en nuestra investigación podrá correr en todos estos sistemas operativos sin grandes cambios. (7)

QGIS viene preparado para interpretar plugin en Python y resulta relativamente sencillo escribirlos en este lenguaje, con solo establecer la estructura definida por él y colocarlo en la carpeta que se requiere el QGIS será capaz de, al iniciar, interpretar los nuevos plugins.

1.6.2. Framework Qt4

Qt4 es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica de usuario sin muchas complicaciones que pueden operar en varias plataformas. Qt4 no es un lenguaje, sino un grupo de clases que pueden ser utilizadas desde algún lenguaje de programación ya definido. Con Qt4 se pueden desarrollar ricas aplicaciones gráficas, incluye soporte de nuevas tecnologías como OpenGL, XML, Bases de Datos, programación para redes, internacionalización y mucho más, como se puede apreciar en la Figura 6. (8)

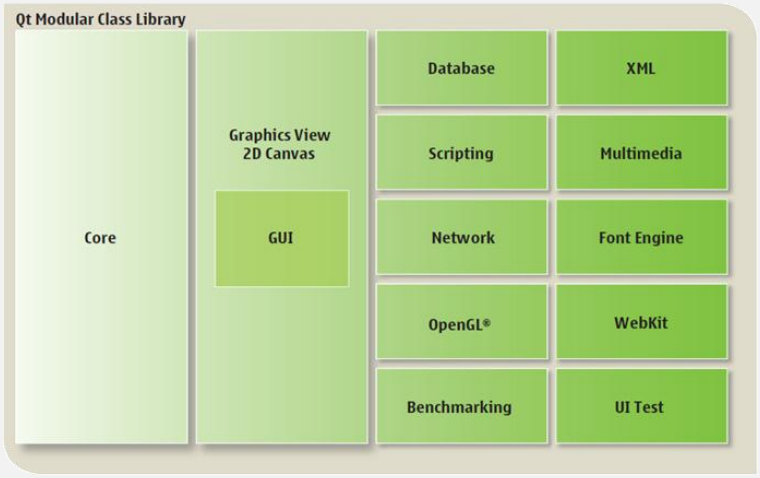


Figura 6. Módulos del Framework Qt4 (8)**.**

Qt4 dispone de una amplia gama de herramientas que facilitan la creación de formularios, botones y ventanas de diálogo. Las aplicaciones creadas con Qt4 son muy elegantes, se ven y se operan mejor que las aplicaciones nativas.

Qt4 dispone de tres grandes ventajas ante las librerías de ventanas rivales, es completamente gratuito para aplicaciones de código abierto. Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix, Linux, MacOS X, como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa. Tiene una extensa librería con clases y herramientas para la creación de ricas aplicaciones. Estas librerías y clases están bien documentadas, son muy fáciles de usar y están orientadas a objetos. (8)

Las librerías Qt4 no están sólo disponibles para C++, sino también ofrecen soluciones para utilizarse con otros lenguajes tales como: Java (QJambi), Python (PyQt), JavaScript (QtScript) y otros.

PyQt4

Una de las vías recomendadas para crear interfaces de usuario en Python es usar PyQt4, que no es la implementación del Framework Qt4 en Python, para el desarrollo de aplicaciones con interfaces gráficas en Python. PyQt4 es un conjunto de herramientas para crear aplicaciones GUI. Es una mezcla de lenguaje

de programación Python y el framework Qt4. Actualmente el soporte de PyQt4 es comercial, dado por Riverbank, sin embargo sigue siendo OpenSource. (9)

PyQt4 se implementa como un conjunto de módulos de Python. Cuenta con más de 300 clases y casi 6000 funciones y métodos. Se trata de un conjunto de herramientas multiplataforma. Se ejecuta en todos los sistemas operativos más importantes, incluyendo Unix, Windows y Mac. PyQt4 tiene licencia dual. Los desarrolladores pueden elegir entre la GPL y la licencia comercial. Anteriormente, la versión GPL sólo estaba disponible en Unix. A partir de la versión PyQt4, de licencia GPL, está disponible en todas las plataformas soportadas. (9)

Debido a que hay una gran cantidad de clases disponibles, se han dividido en varios módulos mostrados en la Figura 7.

Figura 7. Módulos de PyQt4 (9)**.**

El módulo QtCore contiene la funcionalidad básica del framework. Este módulo se utiliza para trabajar con el tiempo, los archivos y directorios, varios tipos de datos, URL, tipos MIME, hilos o procesos. El módulo QtGui contiene los componentes gráficos y las clases relacionadas. Estos incluyen, por ejemplo, botones, ventanas, barras de estado, barras de herramientas, barras de desplazamiento, mapas de bits, colores, fuentes. El módulo QtNetwork contiene las clases para la programación de la red, hace que la programación de la red sea más fácil y más portable. El QtXml contiene clases para trabajar con archivos XML. Este módulo permite la implementación de las APIs SAX y DOM. El módulo QtSvg proporciona clases para mostrar el contenido de archivos Scalable Vector Graphics (SVG), es un lenguaje para describir gráficos en dos dimensiones y aplicaciones gráficas en XML. El módulo QtOpenGL se utiliza para

el procesamiento 3D y gráficos en 2D utilizando la librería OpenGL. El módulo QtSql le proporciona clases para trabajar con bases de datos.

Para que el plugin quede con cualidades visuales similares a las del QGIS es necesario recurrir a las bondades de PyQt4. Además el framework permitirá que el desarrollo se realice de forma rápida, organizado con posibilidades de incorporar nuevas funcionalidades sin muchos esfuerzos, así como desarrollar unas interfaces amigables y fáciles de utilizar por el usuario.

1.6.3. Lenguaje de modelado UML

Lenguaje Unificado de Modelado (UML) por sus siglas en inglés, es el más conocido y utilizado en la actualidad de los utilizados para el modelado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML no es un lenguaje de programación, sino de propósito general para el modelado orientado a objetos. También puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes.

Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. En lugar de indicar los elementos y las reglas, se analizarán directamente los diagramas ya que serán utilizados para hacer el análisis del sistema.

- ✓ **Diagrama de clases** muestra determinadas propiedades y acciones de las clases en relación, así como sus relaciones.
- ✓ **Diagrama de objetos** representa los objetos y sus relaciones.
- ✓ **Diagrama de casos de uso** representa los casos de uso y su relación con los actores.
- ✓ **Diagrama de estados** muestra una secuencia de estados por los que puede transitar el actor.
- ✓ **Diagrama de secuencia** muestra la mecánica de la interacción con base en tiempos. (10)
- ✓ **Diagrama de actividades** muestra las interacciones que ocurren dentro de un caso de uso o dentro del comportamiento de un objeto, se dan en secuencia.
- ✓ **Diagrama de colaboraciones** muestra los elementos del sistema que trabajan en conjunto para cumplir con los objetivos del mismo.

- ✓ **Diagrama de componentes** muestra los componentes del sistema y sus relaciones.
- ✓ **Diagrama de distribución** muestra la arquitectura física de un sistema informático.

Estos diagramas logran representar de forma gráfica lo que se hará en el sistema.

1.6.4. Visual Paradigm 5.0

Las herramientas CASE (Computer-Aided Software Engineering o Ingeniería de Software Asistida por Computadora) se pueden definir como un conjunto de métodos, utilidades y técnicas que dan asistencia a los analistas, ingenieros de software y desarrolladores, facilitando la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. También se puede decir que es la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación. (11)

Visual Paradigm es una herramienta que soporta el UML como lenguaje de modelado, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. (11)

Una herramienta CASE muy parecida al Visual Paradigm, es el Rational Rose, sin embargo, ésta no es una herramienta multiplataforma y posee menos facilidades. Para el desarrollo de la aplicación se escogió el Visual Paradigm por las características y funcionalidades anteriormente explicadas que aportaron a la solución del problema. Con el uso de esta potente herramienta se realizó el modelado de todos los artefactos UML que se generan en el diseño de la aplicación.

1.6.5. Entorno de desarrollo integrado Eclipse Galileo

Un Entorno de Desarrollo Integrado (IDE) por sus siglas en inglés *Integrated Development Environment*, es un programa informático compuesto por un conjunto de herramientas de programación. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Para el desarrollo del

sistema se utiliza el Eclipse, que combinado con el plugin, Python IDE para Eclipse (PYDEV) (12) permite disponer de un IDE para Python cómodo y de calidad. Eclipse cuenta con un editor visual con sintaxis coloreada, ofrece compilación incremental de código, un potente depurador (que permite establecer puntos de interrupción, modificar e inspeccionar valores de variables, e incluso depurar código que resida en una máquina remota), un navegador de clases, un gestor de archivos y proyectos. La versión estándar de Eclipse proporciona también una biblioteca de refactorización de código y una lista de tareas. También incluye una herramienta para completar código. Otra característica de Eclipse es que es muy eficiente para reducir los tiempos de depuración y pruebas (13).

1.7. Metodología de desarrollo

El proceso de desarrollo de software no es simplemente tener un grupo de personas produciendo para terminar en tiempo el producto, pues, sin una guía basada en buenas prácticas, que organice el trabajo es casi imposible lograrlo. Las metodologías de desarrollo de software posibilitan lograr productos con mayor calidad y eficiencia, asegurando su desarrollo y mantenimiento.

La metodología de desarrollo se define como un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevos software (14).

Existen diversas metodologías, cada una con sus peculiaridades y puntos en común. Las metodologías tradicionales o robustas confieren gran peso a la planificación, conceptualización y descripción del sistema que se pretende realizar. De esta forma garantizan que al comenzar la implementación esté bien definido cada elemento a desarrollar. Este tipo de metodologías se ajusta para proyectos a largo plazo de gran envergadura, con equipos de desarrollo numerosos, donde la organización sea fundamental. También es recomendable cuando se requiera una documentación amplia que detalle cada elemento para lograr un entendimiento posterior del software u otras razones.

Las metodologías ágiles o livianas eliminan el burocratismo de las robustas, que en ocasiones resulta contraproducente emplearlas. Estas metodologías se centran en la capacidad de las personas involucradas en el proceso, evitando ir al detalle en cada paso, pero obteniendo el mayor fruto del trabajo de cada integrante. Perfecta para equipos de desarrollo con gran experiencia, pues es vital en este tipo de metodologías para obtener buenos resultados. Estos equipos, preferiblemente pequeños y en proyectos

en los que lo fundamental sea el producto final y la rapidez con que se concluya, sin que la documentación sea primordial.

Tabla 1. Diferencias entre metodologías ágiles y tradicionales (15)

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos.
Pocos Roles, más genéricos y flexibles.	Más Roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo (además in-situ).	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.

Entre las metodologías de desarrollo de software más importantes por su calidad y aplicación en el mundo se encuentran la Metodología Rational Unified Process (RUP) o Proceso Unificado que se enmarca dentro

de las metodologías clásicas, aunque existen variaciones de la misma para adaptarse a procesos ágiles. La Metodología Extreme Programing (XP), exponente de las metodologías ágiles y la Microsoft Solution Framework (MSF) que se adapta como metodología robusta o ágil.

Extreme Programing (XP)

XP proporciona una metodología para desarrollo de proyectos a corto plazo centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (16).

Como elemento de gran importancia en la metodología XP se encuentran las historias de usuario. Las historias de usuarios son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (17).

De acuerdo con la propuesta original de Beck XP define los siguientes roles de usuarios: Programador, Cliente, Encargado de pruebas (Tester), Encargado de seguimiento (Tracker), Entrenador (Coach), Consultor y Gestor (Big Boss).

El ciclo de vida ideal de XP consiste de seis fases (16)**:** Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Entre las prácticas que propone XP se encuentran (15)**:**

El juego de la planificación: El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad pretendida para la aplicación.

Metáfora: El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.

Diseño simple: Propone el diseño de la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

Pruebas: La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

Refactorización: Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.

Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores.

Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista.

40 horas por semana: Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas.

Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo.

Estándares de programación: XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación.

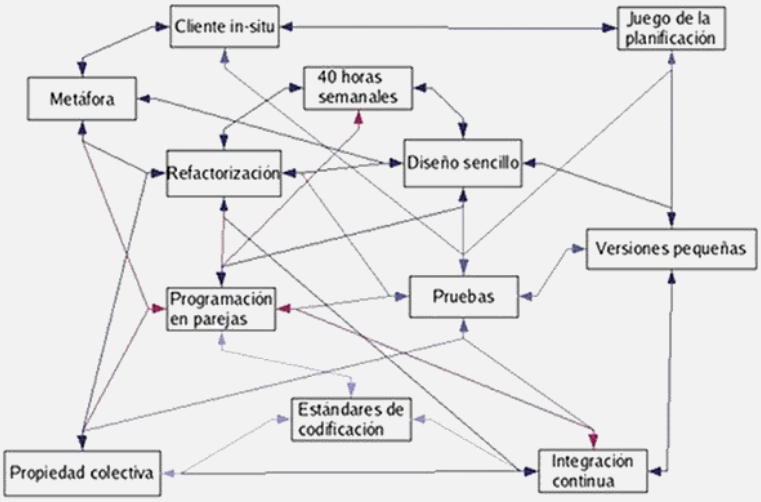


Figura 8. Interrelación entre las prácticas (15).

Microsoft Solution Framework (MSF)

MSF es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo, dejando en un segundo plano las elecciones tecnológicas.

MSF se basa en Principios, Modelos y Disciplinas.

Principios

Los principios de la metodología MSF son promover comunicaciones abiertas, trabajar para una visión compartida, fortalecer los miembros del equipo, establecer responsabilidades claras y compartidas, focalizarse en agregar valor al negocio, permanecer ágil y esperar los cambios, invertir en calidad, aprender de todas las experiencias.

Disciplinas

Administración de proyecto

Esta disciplina se basa en planificar sobre entregas cortas, incorporar nuevas características sucesivamente e identificar cambios ajustando el cronograma.

Administración de riesgos

Esta disciplina está centrada en ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan salir. Proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos.

Control de cambio

Diseñada para que el equipo sea proactivo y no reactivo. Los cambios deben considerarse riesgos inherentes por lo que deben registrarse y hacerse evidentes.

Modelos

Equipo de Trabajo

Alienta la agilidad para enfrentar los cambios involucrando a todo el equipo en las decisiones fundamentales asegurando una perspectiva crítica.

Proceso

Estrategia iterativa en la construcción de productos del proyecto, suministra una imagen más clara del estado de los mismos en cada etapa. Mejora el control del proyecto, minimiza los riesgos y aumenta la calidad acortando el tiempo de entrega.

Proceso Unificado (RUP)

El Proceso Unificado se caracteriza por ser dirigido por *casos de uso*[4], centrado en la arquitectura e iterativo incremental.

Dirigido por casos de uso debido a que estos representan los requisitos funcionales de la aplicación, guían el diseño, la implementación y prueba, es decir, guían el proceso de desarrollo. Ligado a los casos de usos está la arquitectura, que tiene influencia sobre ellos. La arquitectura debe posibilitar el desarrollo de todos los casos de usos y estos a su vez tienen que ajustarse a la arquitectura, por lo que deben desarrollarse de forma paralela. El Proceso Unificado es iterativo incremental pues propone dividir el

[4] *Fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.*

desarrollo del producto en varias iteraciones, en cada una de ellas se obtiene una serie de artefactos que servirá como base de la próxima. Así, en cada iteración se incrementa o modifican los artefactos, posibilitando entre otras cosas la identificación de riesgos, la corrección de errores, mejoras de las funcionalidades y refinar en sentido general el producto, hasta obtener el resultado esperado.

El ciclo de vida de RUP cuenta de cuatro fases: inicio, elaboración, construcción y transición, las que se subdividen en iteraciones y nueve flujos de trabajos, seis flujos de ingeniería y tres de apoyo.

Flujos de Trabajo

La metodología RUP cuenta con nueve flujos de trabajo divididos en dos grandes grupos, en el primero están los flujos de ingeniería que son el modelamiento del negocio, requerimientos, análisis y diseño, implementación, prueba, instalación. El otro grupo es el de apoyo que lo conforman los flujos de administración del proyecto, administración de configuración y cambios y por último ambiente.

En la Figura 9 se muestra la relación que existe entre cada una de las etapas y de los flujos de trabajo, dejando claro que RUP es una metodología completa con mucha flexibilidad y diseñada para obtener excelentes resultados.

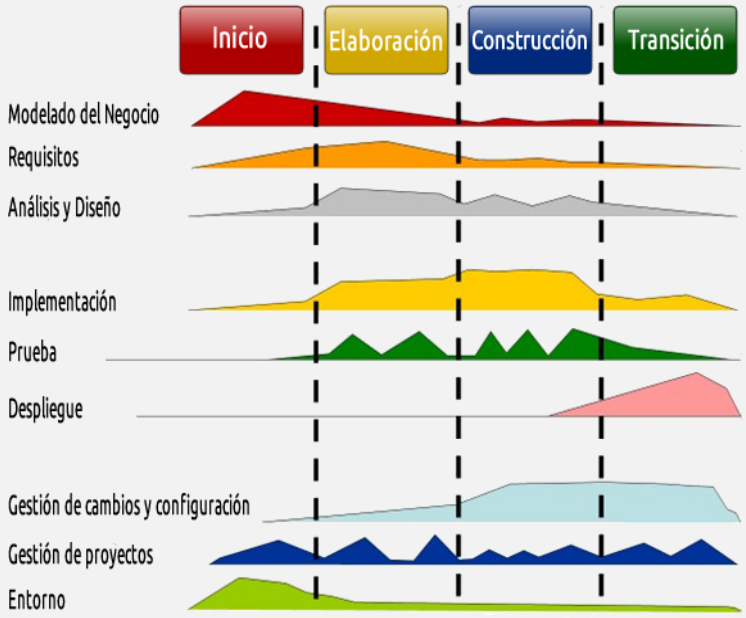


Figura 9. Diagrama de flujos de trabajos de RUP (18)**.**

Se selecciona RUP porque sus características son factibles para la realización del sistema, primero porque el Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza lenguaje modelado unificado (UML) como único lenguaje para describir el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Además es un sistema que tributa al desarrollo del sistema GEOQ en el cual se encuentra inmerso el proyecto SIG-Desktop, que se desarrolla siguiendo los procesos de dicha metodología. También el sistema se va desarrollando y documentando al mismo tiempo por si algún miembro del equipo no puede seguir con el trabajo, el que ocupe su lugar tenga por donde guiarse y conozca qué fue lo que se hizo.

1.8. Conclusiones Parciales

En este capítulo se han expuesto conceptos relacionados con los procesos asociados al Plugin WPS para QGIS, fundamentales para la comprensión de la presente investigación. Se abordan temas relacionados con el estándar WPS como posibles arquitecturas, sus características y algunos otros temas asociados al desarrollo del trabajo. Se analiza el estado actual de estos procesos y características de un conjunto de tecnologías, herramientas y tendencias asociadas a la posible propuesta para solucionar el problema científico que concierne a la investigación. Entre ellas se encuentran los lenguajes, frameworks y metodologías de desarrollo de software, entre otras. De esta manera se proporcionan los conocimientos teóricos necesarios para la mejor compresión y la futura implementación.

CAPITULO II: CARACTERÍSTICAS DE LA SOLUCIÓN

2.1. Introducción

El desarrollo de este capítulo está dedicado a explicar los elementos fundamentales del ciclo de desarrollo que propone la metodología RUP, entre las cuales se puede encontrar el Modelo de Dominio que fue el indicado para dar solución al problema. Se establecen los requisitos no funcionales de la aplicación, los cuales permiten obtener una solución de mayor calidad. Se realizó diagrama de casos de usos de sistema, la descripción de cada uno de los casos de uso y otros artefactos generados en esta etapa. Se abordan algunos términos de arquitectura fundamentales para la implementación de la solución así como las vistas, modelos y patrones.

2.2. Breve descripción de la aplicación

El plugin para QGIS qWPS es un sencillo cliente para QGIS que servirá de enlace entre el mismo y el servidor WPS o Web preparado para brindar procesos de una manera estandarizada para la versión WPS establecida por el OGC, logrando una perfecta interoperabilidad entre el QGIS y el servidor. De esta manera el QGIS será capaz de consumir cualquier tipo de proceso que se encuentre publicado, en cualquier servidor WPS asequible para el mismo, ampliando su potencial de forma considerable. El plugin será capaz de conectarse a cualquier servidor de los antes mencionados, luego de establecer la conexión deberá mostrar todos los procesos disponibles en el servidor para ser ejecutados, así como una descripción detallada de cada uno de estos, por último debe permitir al usuario ejecutar el que sea de su interés. Luego de ser ejecutado ese proceso y usando las librerías y funciones que el QGIS brinda se mostrará, en el panel principal el resultado de lo anteriormente ejecutado.

2.3. Modelo de Dominio

Con motivo de la poca estructuración de los procesos del negocio y para poder comprender el contexto en el cual se enmarca el sistema se determinó desarrollar un Modelo de Dominio, donde se expone un marco conceptual y las relaciones entre estas definiciones. Este permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Un Modelo del Dominio es una representación de las clases conceptuales del mundo real, no de componentes software. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de

software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML para representar conceptos, el Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión.

2.3.1. Glosario de términos para el dominio propuesto

Geoprocesos.- cálculos matemáticos complejos, gráficos, direcciones, códigos postales, diversas funciones que se podrán ejecutar desde el SIG QGIS.

QGIS.- es la plataforma que permitirá la ejecución del Plugin.

Especialista geográfico.- persona encargada de manipular el plugin que permitirá al QGIS el consumo de procesos publicados en un servidor WPS.

2.3.2. Descripción del Modelo de Dominio

El modelo de dominio del presente trabajo, representa las entidades y las relaciones entre ellas. En la Figura 10 se muestra el modelo de dominio para la solución propuesta.

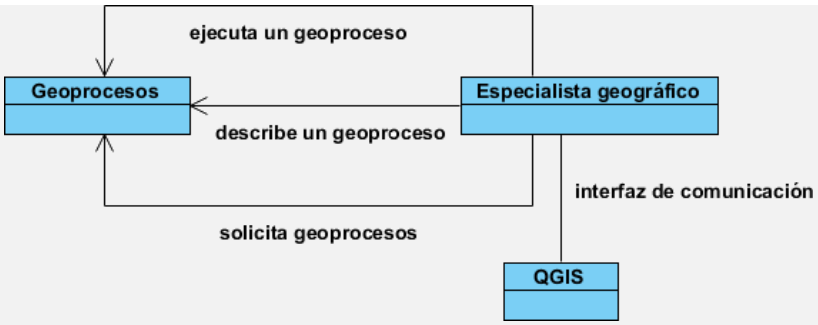


Figura 10. Modelo de dominio.

2.4. Especificación de requisitos del sistema propuesto

2.4.1. Requisitos funcionales

Los requisitos funcionales expresan una especificación detallada de las responsabilidades de la herramienta que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el software.

Los requisitos funcionales del sistema propuesto son los siguientes:

R1 Listar procesos disponibles en el servidor WPS.

Descripción:

El sistema debe mostrar al usuario todos los procesos que están disponibles en el servidor WPS.

Entradas:

No tiene.

Salidas:

Listado de procesos: Muestra un listado con los procesos disponibles.

R2 Obtener información de un proceso.

Descripción:

El sistema debe mostrar al usuario la descripción de un proceso luego de este seleccionarlo para su posterior ejecución.

Entradas:

No tiene.

Salidas:

Mensaje de notificación: Muestra un formulario con toda la descripción del proceso.

Facilita los campos o acciones pertinentes para la ejecución del proceso.

R3 Ejecutar un proceso.

Descripción:

El sistema debe ejecutar un proceso del servidor.

Entradas:

No tiene.

Salidas:

No tiene.

R4 Añadir nueva conexión.

Descripción:

El sistema debe permitir al usuario añadir todas las conexiones que desee.

Entradas:

Campo donde especificar el servidor WPS: Se entrará una nueva dirección URL válida de la dirección del servidor con el cual se desea establecer la conexión.

Campo donde especificar el nombre de la conexión: Se entrará un nombre para la conexión.

Salidas:

Se actualiza el listado de las conexiones con los nuevos parámetros.

R5 Editar conexión disponible.

Descripción:

El sistema debe permitir al usuario editar una conexión WPS determinada.

Entradas:

Campo donde especificar el servidor WPS: Se entrará una nueva dirección URL válida de la dirección del servidor con el cual se desea establecer la conexión.

Campo donde especificar el nombre de la conexión: Se entrará un nombre para la conexión.

Salidas:

Se actualiza el listado de las conexiones con los nuevos parámetros.

R6 Terminar conexión con un servidor WPS.

Descripción:

El sistema debe permitir al usuario desconectarse de un servidor WPS determinado.

Entradas:

No tiene.

Salidas:

Se elimina la conexión de la lista de conexiones disponibles.

2.4.2. Requisitos no funcionales

Los requisitos no funcionales, son aquellas propiedades o cualidades que el producto debe tener, pues representan sus características. A continuación se describen cuáles son algunas de las cualidades más significativas que deberá cumplir la aplicación:

- ✓ **Apariencia o interfaz externa:**

La aplicación debe diseñarse con una interfaz amigable, de forma tal que siga las pautas de diseño del QGIS, ajustándose a los estándares establecidos para el desarrollo de un buen diseño.

- ✓ **Usabilidad:**

Debe tener una interfaz gráfica, visualmente atractiva para el usuario. La aplicación podrá ser usada por cualquier usuario con conocimientos básicos sobre geografía e informática. Debe mostrar mensajes al usuario que le ayuden a llevar a cabo la tarea que realiza.

- ✓ **Seguridad:**

El usuario tendrá control total de la aplicación, sin restricción alguna.

- ✓ **Software:**

Se debe disponer del sistema de información geográfica QGIS versión 1.5 o superior instalado para la posterior integración del plugin de la aplicación.

- ✓ **Hardware:**

Para el desarrollo y puesta en práctica del plugin se requieren máquinas con tarjeta de red, procesador Pentium 4 o superior, 1 GB de memoria RAM y una tarjeta gráfica.

2.5. Definición del sistema propuesto

2.5.1. Actores del sistema

Los actores, son terceros fuera del sistema, que interactúan con él de alguna manera. Generalmente estimulan al sistema con eventos de entrada o reciben algo de él. Pueden intercambiar información o ser un recipiente pasivo de información. El actor del sistema, así como una breve descripción de la función que tendrá dentro de éste, se representa en la siguiente tabla.

Tabla 2. Actores del sistema

Actores	Descripción
Especialista geográfico	Representa el usuario que va a hacer uso de la aplicación, teniendo la posibilidad de interactuar con todas la funcionalidades de ésta.

2.5.2. Diagrama de Casos de Uso del Sistema

Los casos de uso del sistema se relacionan estrechamente con los requisitos funcionales planteados con anterioridad, pues representan las funcionalidades que tendrá el sistema. La Figura 11 muestra el diagrama de casos de uso de la aplicación propuesta. Se muestra cómo el especialista geográfico interactúa con las funcionalidades de la aplicación.

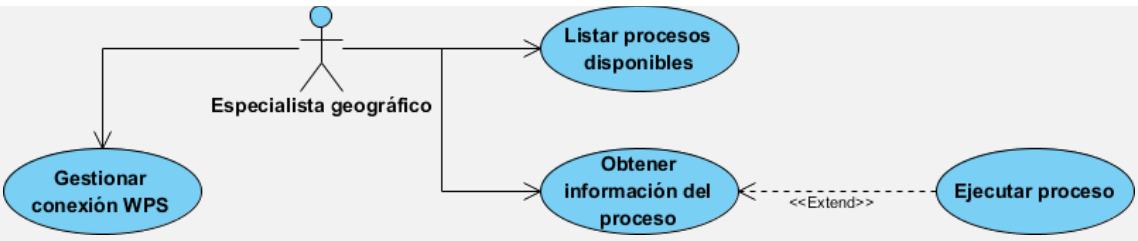


Figura 11. Diagrama de casos de uso del sistema propuesto.

2.5.3. Descripción textual de los Casos de Uso del Sistema

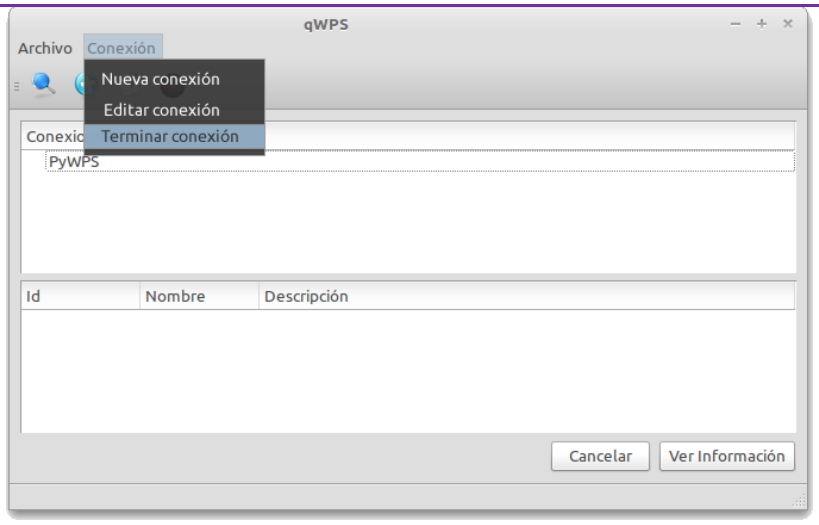
Con la representación gráfica del diagrama de casos de uso no es suficiente para lograr entender las funcionalidades asociadas a un caso de uso, por lo que es necesario describir textualmente cada uno de éstos, de manera que queden especificadas todas las acciones necesarias que realizan el actor y el sistema. Con las descripciones que se presentan a continuación de los CU se obtendrá una idea más clara de cómo se realiza el proceso de automatización y quiénes intervienen directamente en éste.

Tabla 3. Descripción del CU: Gestionar conexión

Caso de Uso:	Gestionar conexión WPS
Actores:	Especialista geográfico
Resumen:	El caso de uso se inicia cuando el Especialista geográfico desde la opción Conexión podrá crear, modificar o eliminar una conexión. El caso de uso termina con la creación, actualización o la eliminación de una conexión.
Precondiciones:	
Referencias	R4, R5, R6
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El Especialista geográfico presiona la opción Conexión de la barra de herramientas.	**2.** El sistema muestra tres opciones “Nueva conexión”, “Editar conexión” y “Terminar conexión”.
3. El Especialista geográfico escoge una de las opciones: a. Crear una nueva conexión, pulsando el botón “Nueva conexión”. Sección Añadir nueva conexión WPS. b. Editar datos de la conexión, seleccionando el botón “Editar conexión”. Sección Editar conexión	

disponible. c. Eliminar una conexión, seleccionando el botón “Terminar conexión”. Sección Eliminar usuario.	

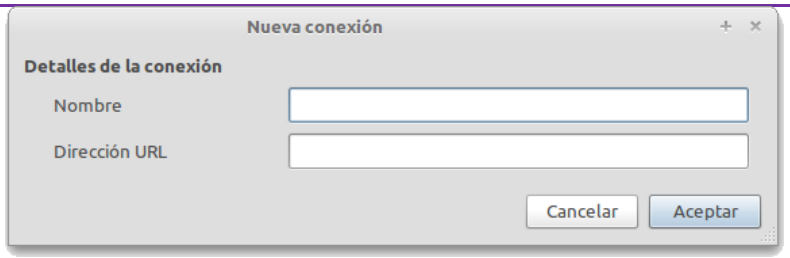
Prototipo de Interfaz



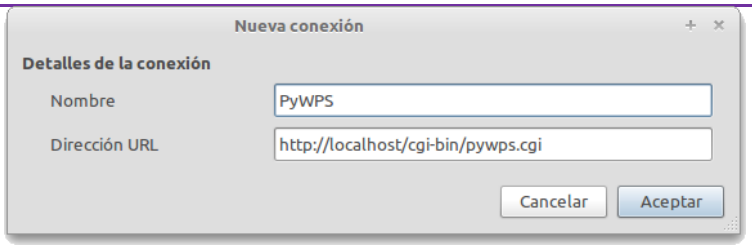
Añadir nueva conexión WPS

Acción del Actor	Respuesta del Sistema
	4. El plugin muestra un formulario con los campos para entrar el nombre y la dirección del servidor.
5. El usuario entra los datos correctos y presiona Aceptar.	**6.** Este caso de uso finaliza cuando se crea la conexión.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
7. El usuario ya no desea añadir ninguna conexión y presiona Cancelar.	**8.** El sistema cancela la operación y vuelve a la interfaz anterior.
Editar conexión disponible	
Acción del Actor	**Respuesta del Sistema**
4. El usuario selecciona una conexión del listado de conexiones disponibles y presiona la opción "Editar conexión"	**5.** El plugin muestra un formulario con los datos a editar y las opciones Aceptar y Cancelar.
6. El usuario modifica los datos deseados y selecciona la opción Aceptar.	**7.** Se guardan los datos modificados y se muestran en el listado de conexiones.
Prototipo de Interfaz	
	
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
6. El usuario selecciona la opción Cancelar.	**7.** El sistema cancela la operación y vuelve a la interfaz anterior.
Terminar conexión	
Acción del Actor	**Respuesta del Sistema**
4. El usuario selecciona una conexión del listado de conexiones disponibles y presiona la opción "Terminar conexión"	**5.** Este caso de uso finaliza cuando termina la conexión con el servidor y se eliminan los procesos correspondientes.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
4. El usuario no selecciona ninguna conexión del listado de conexiones disponibles.	**5.** El sistema muestra un mensaje indicando que debe seleccionar una conexión.

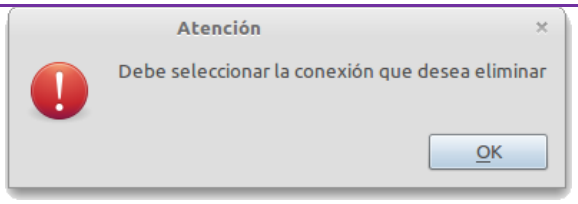
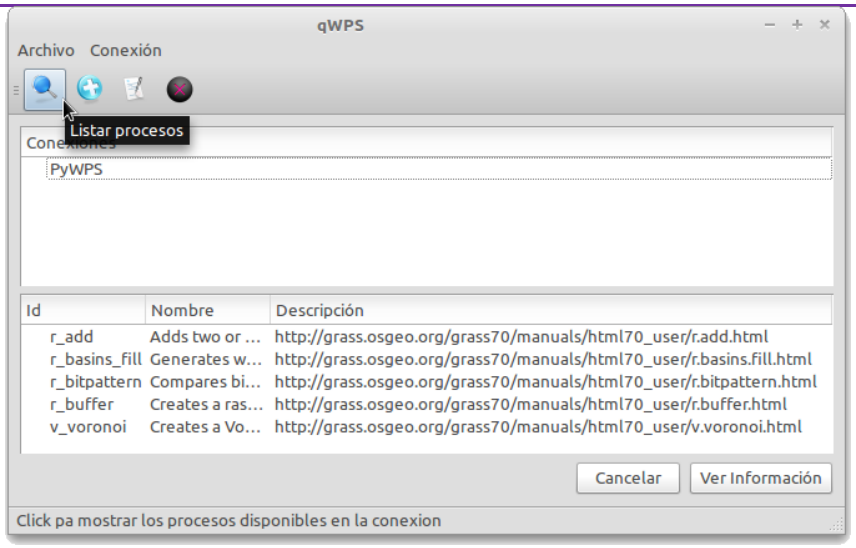
Prototipo de Interfaz


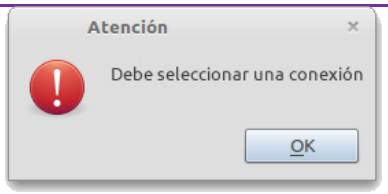
Tabla 4. Descripción del CU: Listar procesos.

Caso de Uso:	Listar procesos
Actores:	Especialista geográfico
Resumen:	El caso de uso se inicia cuando el usuario se conecta al servidor WPS y la aplicación muestra un listado de los procesos habilitados en el servidor.
Precondiciones:	Que se haya creado una conexión WPS.
Referencias	R1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando el usuario selecciona la conexión de la cual desea visualizar los procesos y selecciona la opción “Listar proceso” de la barra de herramientas.	2. Este caso de uso finaliza cuando el sistema muestra los procesos asociados al servidor.
Prototipo de Interfaz	



Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario no selecciona ninguna conexión y selecciona la opción "Listar proceso" de la barra de herramientas.	**2.** El sistema muestra un mensaje indicando que debe seleccionar una conexión.

Prototipo de Interfaz



Poscondiciones	Listado de los procesos.

Tabla 5. Descripción del CU: Obtener información.

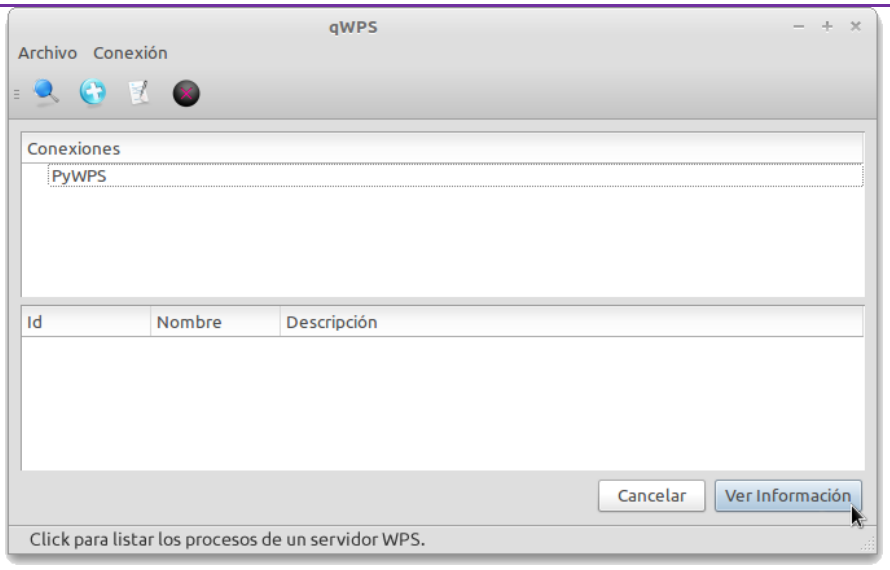
Caso de Uso:	Obtener información
Actores:	Especialista geográfico
Resumen:	El caso de uso se inicia cuando el usuario selecciona uno de los procesos disponibles, después el sistema muestra la información del proceso y culmina

	mostrando parámetros de entrada del servicio en cuestión.
Precondiciones:	Que se tenga un listado de un grupo de servicios WPS y se haya establecido de forma correcta la conexión.
Referencias	R2
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario selecciona uno de los procesos disponibles en el servidor y selecciona la opción “Ver información”.	**2.** El sistema realiza una petición al servidor WPS para obtener los detalles del proceso.
	3. El servidor responde adecuadamente a la petición y se muestran por pantalla los datos concretos del proceso, incluyendo su nombre, descripción y parámetros de entrada.
	4. Este caso de uso finaliza mostrando los parámetros necesarios para la ejecución del proceso asociado.

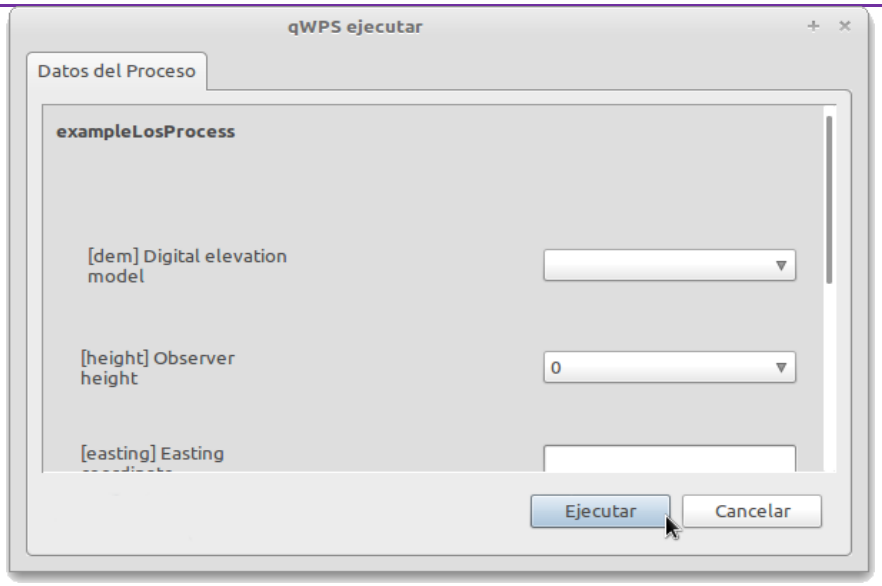
Prototipo de Interfaz



Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	3. No es posible establecer la conexión con el servidor.
Prototipo de Interfaz	
Poscondiciones	Se muestran los datos necesarios para la ejecución del proceso.

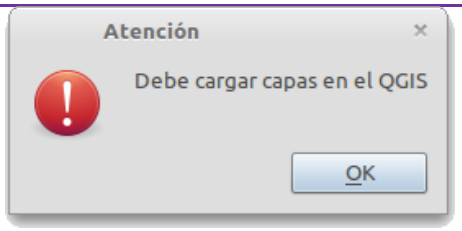
Tabla 6. Descripción del CU: Ejecutar un proceso.

Caso de Uso:	Ejecutar proceso.
Actores:	Especialista geográfico
Resumen:	El caso de uso se inicia cuando el usuario pulsa Ejecutar, el sistema debe ser capaz de ejecutar un proceso y culmina mostrando un mensaje con el estado de la ejecución.
Precondiciones:	Que se tenga la información completa del proceso.
Referencias	R3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando el usuario completa los parámetros de entrada necesarios para la ejecución del proceso.	**2.** El sistema valida los datos especificados por el Especialista geográfico.
	3. El sistema realiza la petición de ejecución al servidor WPS.
	4. Este caso de uso finaliza cuando el sistema muestra el resultado de la ejecución del proceso en el QGIS.
Prototipo de Interfaz	



Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	2. El sistema muestra un error indicando que los parámetros no son válidos y pasa a la acción 1 del flujo normal de los eventos.
	3. El sistema muestra un mensaje indicando que no fue posible establecer la conexión.

Prototipo de Interfaz



Poscondiciones	Se ejecuta un servicio WPS.

1. Conclusiones parciales

En este capítulo se determinaron las características básicas con las que debe contar un sistema para poder utilizar el producto. Se seleccionó el Especialista geográfico como actor del sistema y se definieron

siete requisitos funcionales, con los cuales se confeccionó del diagrama de CUS. Además, se consumó la descripción textual de los CUS, para lograr un mejor entendimiento de las funcionalidades de la herramienta a desarrollar.

CAPITULO III: DISEÑO DEL SISTEMA

3.1. Introducción

Para la confección de cualquier sistema informático es fundamental su arquitectura pues de ella dependen requisitos tanto funcionales como no funcionales de dicha solución. Para lograr el cumplimiento de los requisitos es necesario el diseño de una arquitectura de software robusta y confiable.

3.2. Descripción de la arquitectura

El diseño de la arquitectura de un sistema es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el sistema, cómo se relacionan entre ellos y cómo mediante su interacción llevan a cabo la funcionalidad especificada, cumpliendo con los criterios de calidad indicados como seguridad, disponibilidad, eficiencia o usabilidad. También se puede ver como una vista estructural de alto nivel que ocurre tempranamente en el ciclo y define los estilos o grupo de estilos adecuados para cumplir los requerimientos no funcionales.

3.2.1. Patrón arquitectónico

Las Arquitectura en Capas constituye uno de los estilos más completos que existen en el mundo de la arquitectura, define el estilo en capas, como una organización jerárquica tal que cada capa proporciona servicios a cada capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Con esta se logra abstraer las funcionalidades de una capa de manera tal que pueda ser totalmente remplazada. De esta arquitectura la más común es la compuesta por tres capas, presentación, modelo o reglas del negocio y acceso a datos. De esta forma se puede remplazar cualquier capa sin afectar a las otras, solamente cambiar las referencias de las implicadas en el cambio.

En la Figura 12 se muestra la propuesta de la arquitectura del sistema, donde se puede apreciar que está compuesta por tres capas fundamentales.

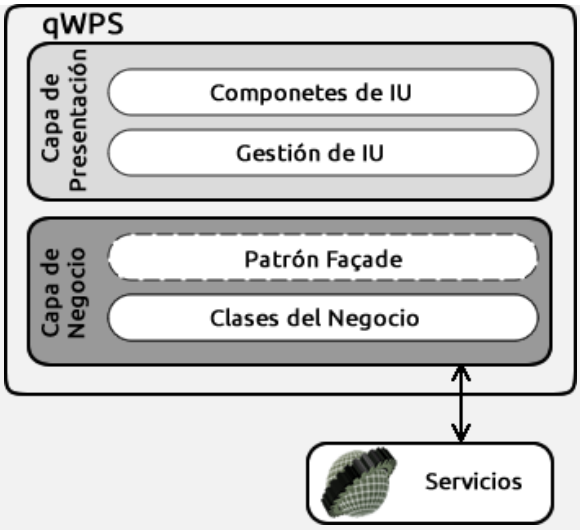


Figura 12. Arquitectura del sistema (Elaboración propia).

Algunas de las ventajas que presenta la arquitectura en capas para la aplicación son:

El mantenimiento y las mejoras de la solución son fáciles debido al bajo acoplamiento entre las capas, la alta cohesión de las capas y la posibilidad de cambiar su implementación sin cambiar las interfaces.

El desarrollo distribuido es fácil si este se puede dividir con las capas como fronteras.

Distribuir las capas a lo largo de múltiples capas físicas puede mejorar la escalabilidad, tolerancia a errores y rendimiento.

Beneficios a la hora de realizar las pruebas teniendo bien definidas las interfaces de las capas por la posibilidad de cambiar las implementaciones de estas capas manteniendo la interfaz.

3.2.1.1. Capa de presentación

La capa de presentación es la parte de la aplicación con que el usuario interactúa, por lo que deberá cumplir muchos requisitos. Estos requisitos abarcan factores generales como la facilidad de uso, rendimiento, diseño e interactividad. Es importante diseñar la aplicación para apoyar una experiencia de usuario intuitiva, desde el principio, ya que la experiencia del usuario es influenciada por muchos aspectos diferentes de la arquitectura de la aplicación. Esta capa solamente se comunica con la capa de negocio.

3.2.1.2. Capa de negocio

La capa de negocio es parte comercial del sistema y generalmente se muestra como un servicio. Aquí es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al servidor los procesos.

3.3. Patrones de diseño de software

Los patrones de diseño son el inicio en la búsqueda de soluciones en el desarrollo de software y otros ámbitos referente al diseño de interacción o interfaces. Para que la solución sea factible el patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores y ser reusables, lo que significa que se puede aplicar a diferentes problemas de diseño en diferentes circunstancias.

3.3.1. Patrones GRASP

GRASP es el acrónimo de General Responsibility Assignment Software Patterns, son patrones de diseño que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Una de las cosas más complicadas en Orientación a Objeto consiste en elegir las clases adecuadas y decidir cómo estas clases deben interactuar. Es inevitable elegir cuidadosamente las responsabilidades de cada clase en la primera codificación de nuestro programa.

Patrón Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Una clase contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras estemos considerando los mismos aspectos del sistema: Lógica del negocio.

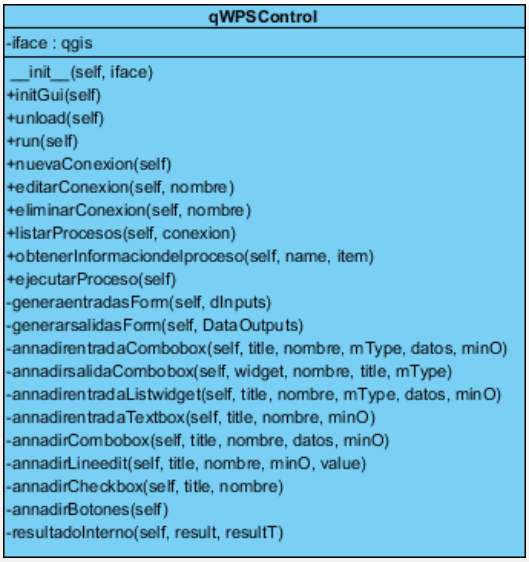


Figura 13. Evidencia del patrón Experto en la clase qWPS.

Patrón controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

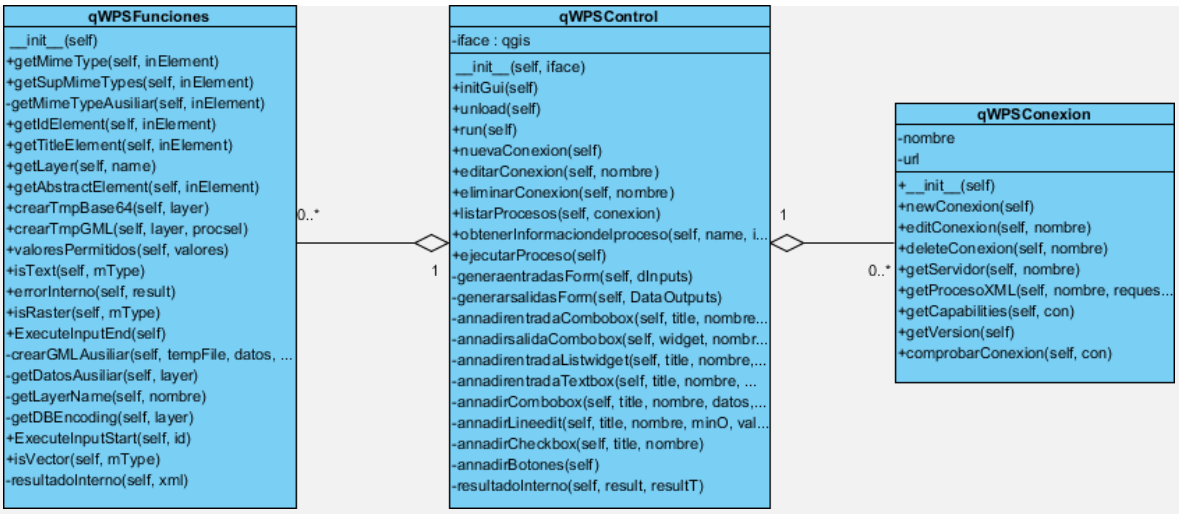


Figura 14. Evidencia del patrón Controlador.

Patrones GOF

En busca de mejores resultados y una solución robusta también se tuvieron en cuenta los patrones GOF descritos en el libro *"Design Patterns: Elements of Reusable Object-Oriented Software",* según el cual existen tres tipos:

- ✓ **De Creación:** son los que abstraen el proceso de creación de instancias.
- ✓ **Estructurales:** se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
- ✓ **De Comportamiento:** atañen a los algoritmos y a la asignación de responsabilidades entre objetos (19).

Template Method (Método plantilla): es un patrón de comportamiento que define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

El patrón Template Method se evidencia en las clases que representan interfaces de usuario, pues estas heredan todas las funcionalidades de la clase object como se muestra en la Figura 15.

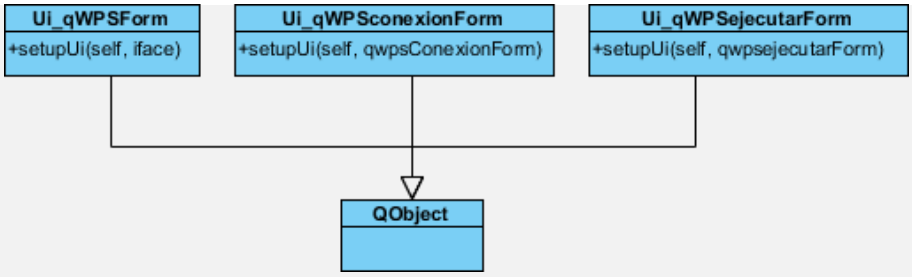


Figura 15. Evidencia del patrón Plantilla.

Patrón Facade: Buscando proporcionar una interfaz fácil de manipular y unificada para el sistema, reduciendo su complejidad y minimizando las dependencias se utilizó el patron Facade representado en la Figura 16.

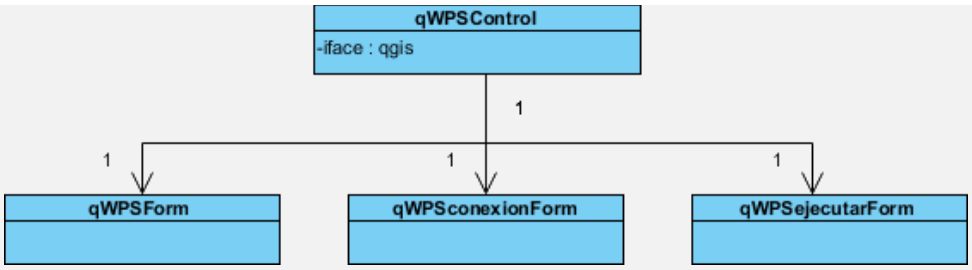


Figura 16. Evidencia del patrón Facade.

3.4. Modelo de Diseño

El modelo de diseño es no genérico, específico para una implementación. Dinámico (centrado en las secuencias). Es un manifiesto del diseño del sistema, incluyendo su arquitectura. Debe mantenerse durante todo el ciclo de vida del software. Además, da forma al sistema, mientras intenta preservar la estructura definida por el modelo de análisis.

3.4.1. Diagrama de clases del diseño

Los diagramas de clases de diseño tienen como premisa el análisis del sistema y su objetivo es dejar la estructura de la solución planteada en un lenguaje de programación específico.

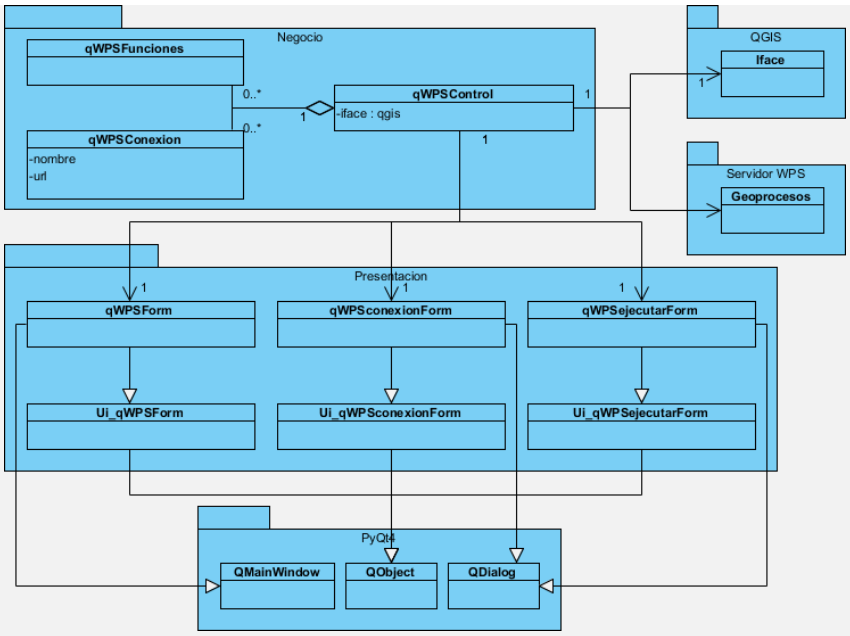


Figura 17. Diagrama de clases del diseño.

3.4.2. Descripción de las clases principales

La clase ***qWPS*** es la clase controladora de la aplicación, la que se encarga de la gestión referente al negocio, es decir, que es la encargada de controlar el flujo de información entre el usuario y la aplicación en general. En la Tabla 7 se aprecia la descripción de la clase ***qWPS***.

Tabla 7. Descripción de la clase qWPS

Descripción de la clase controladora qWPSControl del plugin qWPS	
Nombre: qWPSControl	
Tipo de Clase: Controladora	
Atributos:	**Tipo:**
iface	qgis
Principales Responsabilidades:	
Nombre:	class **qWPSControl:**
Descripción:	Controlar las interfaces.

3.5. Diagramas de secuencias

Un diagrama de secuencia muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal. Estos diagramas no muestran los enlaces existentes entre objetos. Los diagramas de secuencia tienen distintos formatos, adecuados para propósitos diferentes.

A continuación se muestran los diagramas de secuencias para los CU Obtener información del proceso Figura 18 y Listar procesos disponibles Figura 19. Para ver los restantes diagramas, correspondientes a los demás casos de usos dirigirse a los Anexos 1 y 2.

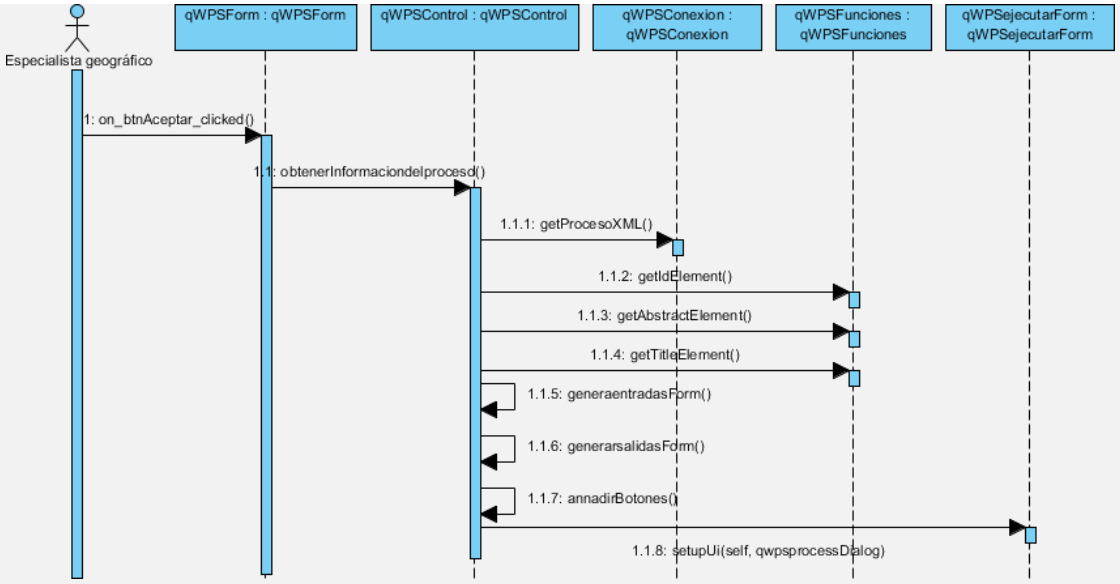


Figura 18. Diagrama de secuencia CU Obtener información del proceso.

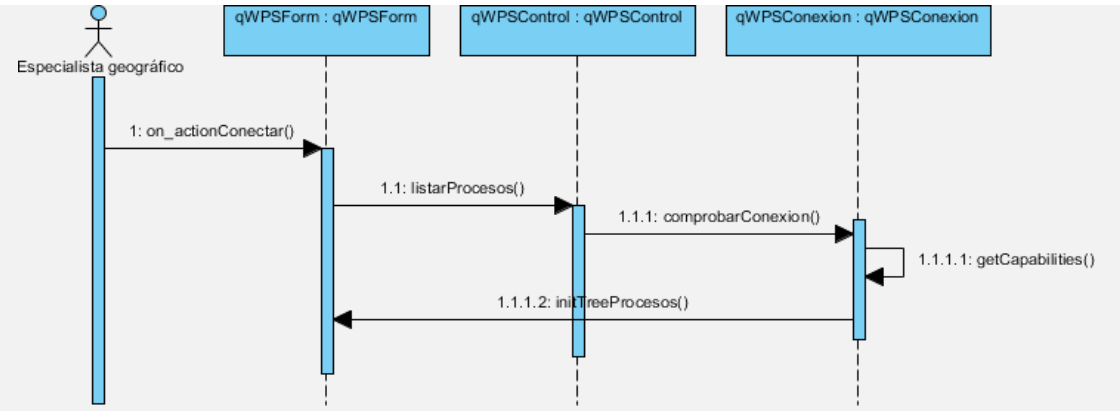


Figura 19. Diagrama de secuencia CU Listar procesos disponibles.

3.6. Modelo de despliegue

El modelo de despliegue, describe la arquitectura física del sistema durante su ejecución, en términos de procesadores, dispositivos y componentes de software. Describe, además, la topología del sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

En la Figura 20 se muestra el modelo de despliegue de la aplicación desarrollada para la cual es necesario una ***PC cliente*** con el QGIS instalado y, además, el mismo debe tener instalado el plugin qWPS

que es donde el especialista geográfico podrá ejecutar los procesos disponibles en un servidor WPS y utilizar el resultado en una nueva capa del QGIS. La comunicación con el ***Servidor WPS*** se realiza a través del protocolo HTTP, en éste estarán disponibles una serie de procesos.

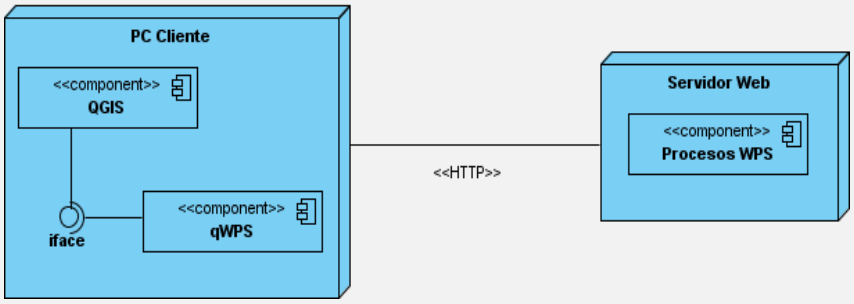


Figura 20. Diagrama de despliegue.

3.7. Conclusiones Parciales

Para el desarrollo de la aplicación en el presente capítulo se describieron el estilo arquitectónico en n capas, los principales patrones que se usaron como son los patrones GOF y GRASP. Se realizaron los diagramas de clases de diseño y de secuencia para cada uno de los casos de usos del sistema. Además se describió cada uno de los nodos necesarios para que se ejecute la aplicación mediante el diagrama de despliegue de la aplicación.

CAPITULO IV: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN

4.1. Introducción

En el presente capítulo se describe la implementación del sistema el cual es uno de los flujos más importantes en el proceso de desarrollo que establece la metodología RUP, entre otras cosas se estará diseñando el diagrama de componentes de la aplicación. Para la realización de este capítulo se tendrán en cuenta las bases creadas por el flujo de trabajo de diseño, descrito en el capítulo anterior. Además, se describen detalladamente las pruebas de caja negra que se le realizan al sistema en busca de cualquier detalle en el funcionamiento que no se haya tomado en cuenta durante su implementación.

4.2. Modelo de implementación

El modelo de implementación es una visión general de lo que se debe implementar, que coincide con la plantilla del plan de integración, con los componentes y subsistemas a implementar, así como con los resultados que se deben obtener y las pruebas que se deben realizar sobre ellos.

4.2.1. Diagrama de componentes

Un diagrama de componentes es la representación de la forma en que los componentes físicos de un sistema serán separados, pueden ser: archivos, cabeceras, módulos, paquetes. Muestra la organización y las dependencias que existen entre un conjunto de componentes. Además, se utilizan para modelar la vista estática de un sistema.

En el diagrama de componentes del plugin Figura 21, se muestran los componentes separados por paquetes, según los requerimientos del patrón en capas y según la función que cumple cada uno de éstos.

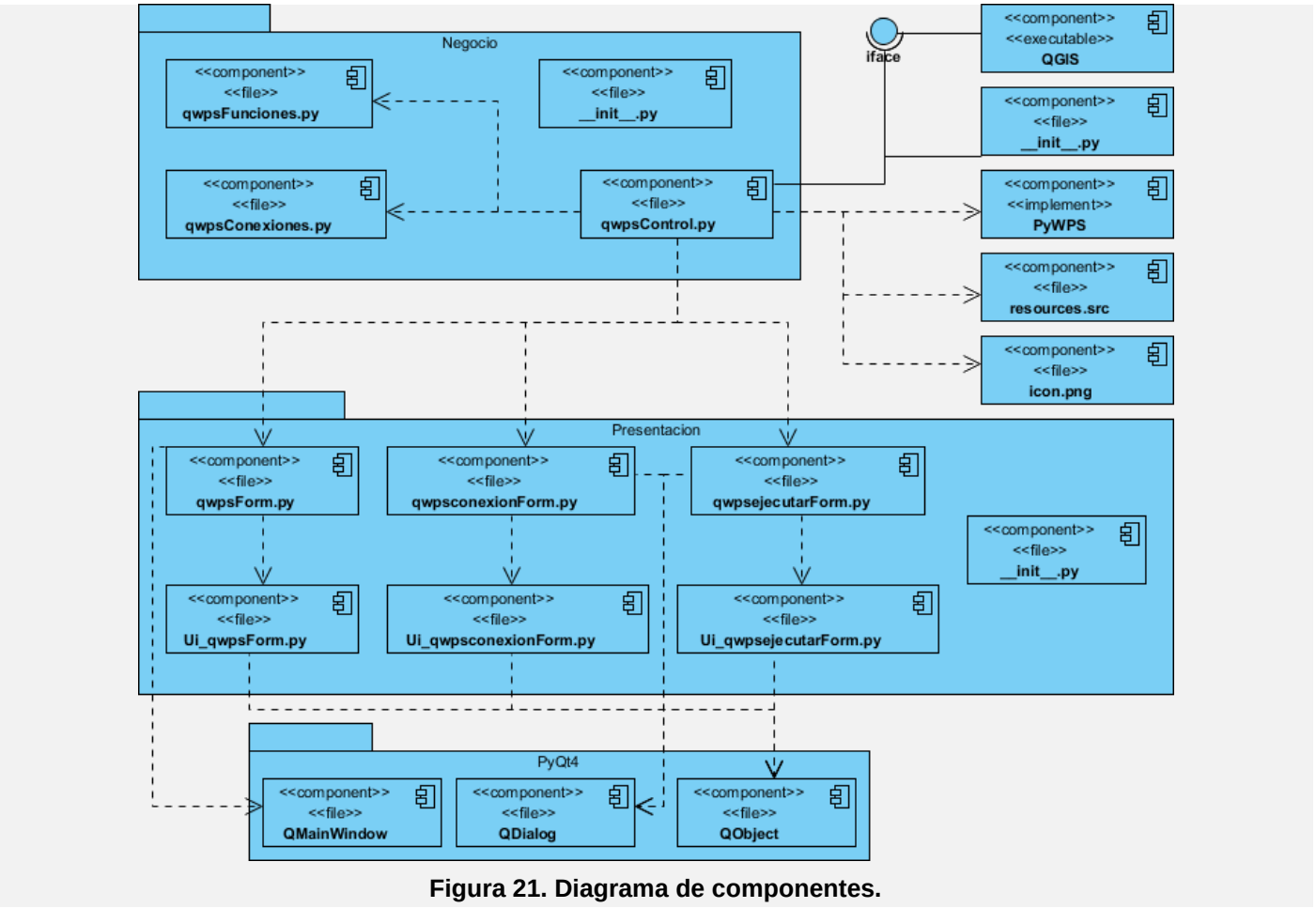


Figura 21. Diagrama de componentes.

4.3. Integración y funcionamiento

Integración del plugin qWPS al QGIS

Para la correcta integración de qWPS con el QGIS es necesario como acción primera copiar la carpeta que contiene la estructura adecuada y todas las funcionalidades implementadas en la carpeta que el QGIS tiene habilitada para los plugin, que este cada vez que se ejecuta va a esa carpeta y lee todo su contenido en busca de nuevos plugin, luego el especialista geográfico debe dar clic en la barra de herramientas el menú Complementos Figura 22, luego Administrar Complementos, y se mostrará un formulario con todos los plugin habilitados y los disponibles, se marca el plugin con nombre qWPS Figura 23 y aparecerá un ícono en el toolbar del QGIS Figura 24 correspondiente al plugin listo para su uso, además de que en la barra de Herramienta en el mismo menú saldrá una nueva opción con el nombre del plugin.

Una vez que se ejecuta el plugin, éste muestra un formulario con las opciones que le brinda para realizar las funciones implementadas. El especialista geográfico luego de establecer la conexión con un servidor WPS elige el proceso que desee ejecutar y mediante el botón Ver Información se le muestra al especialista un formulario con todos los datos de entrada y salida necesarios para que se ejecute el proceso con un botón Ejecutar, luego que dar clic, se muestra el resultado en una nueva capa en el QGIS.

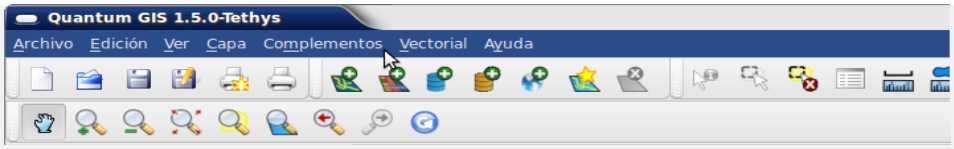


Figura 22. Menú Complementos del QGIS.

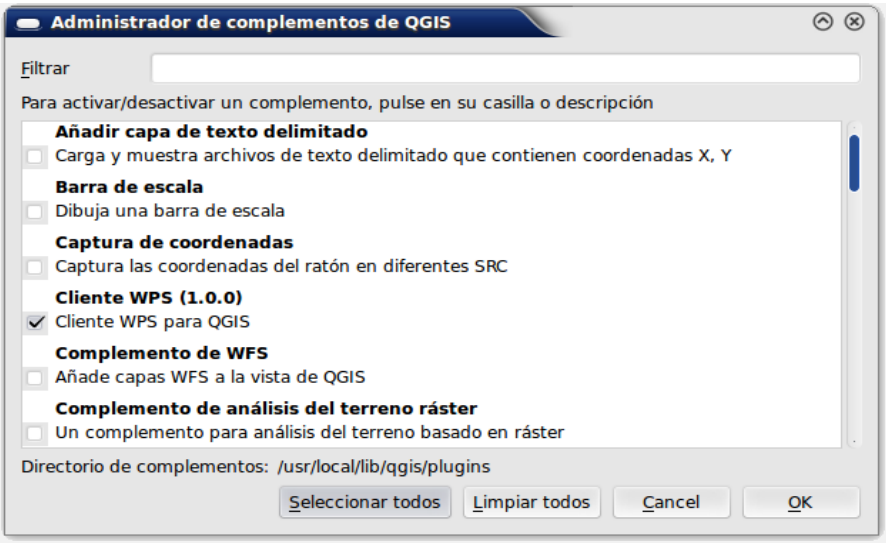


Figura 23. Administrar complementos de QGIS.

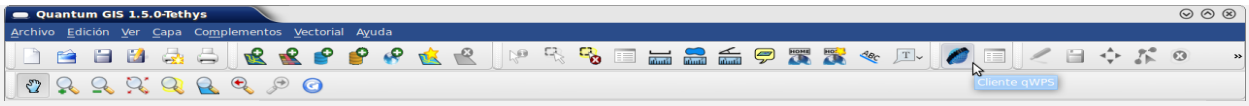


Figura 24. Ícono del plugin en el toolbar de QGIS.

4.4. Pruebas

El objetivo principal de las pruebas de software es descubrir errores. Según Pressman, las pruebas de caja negra se centran en los requisitos funcionales del software. La prueba de caja negra permite obtener

conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca. (20)

Las pruebas de caja negra son aquellas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba tienen como objetivo demostrar que las funcionalidades de la aplicación son operativas, que los datos de entrada se acepten correctamente y que se produce una salida adecuada que garantiza la integridad de la información que se almacena y procesa.

4.4.1. Diseño de prueba de caja negra

Las pruebas de caja negra permiten identificar las posibles fallas en el funcionamiento del sistema. Estas pruebas se centran principalmente en las características del producto independiente del código. Con la ejecución de estas pruebas es posible encontrar errores de interfaz, errores de inicialización y terminación así como funciones incorrectas o ausentes.

Para llevar a cabo estas pruebas es necesario tener conocimiento sobre los escenarios de prueba y secciones que serán probadas, con estos escenarios es muy fácil comprender el funcionamiento del requisito en cuestión. En los casos de pruebas se combinan los posibles juegos de datos válidos e inválidos que son necesarios para verificar el correcto funcionamiento del caso de uso. Estas pruebas se realizan a nivel de sistema, el tipo de prueba es funcional, empleando el enfoque estructural o de caja negra, específicamente usando la técnica de particiones equivalentes. Esta técnica es muy efectiva al realizar pruebas sobre la interfaz de la plataforma, estás prueban la validez de cada entrada de datos o información al sistema. Pretenden demostrar que las funciones de la plataforma son operativas.

4.4.1.1. Diseño para CU Gestionar conexión

Tabla 8. Secciones a probar del caso de uso Gestionar conexión.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo central
SC 1 Añadir	EC 1.1 Añadir conexión	El especialista geográfico selecciona la opción Nueva conexión y se	Clic en Conexión/Editar

conexión	exitosamente.	muestra un formulario con los datos que tiene que pasar a la nueva conexión.	conexión
	EC 1.2 Añadir conexión (datos incorrectos).	En caso de entrar datos incorrectos, el sistema muestra un mensaje indicando que los datos no son válidos.	El especialista geográfico no introduce correctamente los datos.
SC 2 Editar conexión	EC 2.1 Editar conexión exitosamente.	El especialista geográfico selecciona la conexión que desea modificar de las conexiones disponibles y selecciona la opción en el menú Conexión/Editar conexión el sistema muestra un formulario con los datos que desea modificar.	Clic en Conexión/Editar conexión
	EC 2.2 Editar conexión (datos incorrectos).	En caso de entrar datos incorrectos, el sistema muestra un mensaje indicando que los datos no son válidos.	El especialista geográfico no introduce correctamente los nuevos datos.
SC 3 Terminar conexión	EC 3.1 Terminar conexión exitosamente.	El especialista geográfico selecciona la conexión que desea finalizar y acciona la opción Terminar conexión en el menú Conexión. El sistema termina la conexión y se elimina la conexión de la lista de conexiones disponibles.	Clic en Conexión/Terminar conexión
	EC 3.2 Terminar conexión fallida.	El especialista geográfico no selecciona ninguna conexión.	Conexión/Terminar conexión

Diseño de las pruebas para cada sección del caso de uso Gestionar conexión.

SC 1 Añadir nueva conexión

Variables para todos los escenarios

Var 1: Nombre

Var 2: Dirección URL

Var 3: Seleccionar conexión

Tabla 9. SC 1 Añadir nueva conexión

ID del escenario	Escenario	Var 1	Var 2	Respuesta del sistema	Resultado de la prueba
EC 1.1	Añadir nueva conexión exitosamente	V PyWPS	V http://localhost/wps	El sistema guarda la nueva conexión y la muestra en el listado de conexiones disponibles.	Se cambiaron satisfactoriamente los datos.
EC 1.2	Añadir conexión (datos incorrectos)	I “Dejar campo vacío”	I “Dejar campo vacío”	El sistema muestra un mensaje de error “Debe llenar los campos”.	Se muestra un mensaje de error.

Tabla 10. SC 2 Editar conexión

ID del escenario	Escenario	Var 1	Var 2	Respuesta del sistema	Resultado de la prueba
EC 1.1	Editar conexión exitosa	V PyWPS	V http://localhost/wps1	El sistema se cambia los datos y se muestra la nueva conexión.	Se cambiaron satisfactoriamente los datos.
	Editar	I	I	El sistema muestra	Se muestra un

	conexión (datos incorrectos)	"Dejar campo vacío"	"Dejar campo vacío"	un mensaje de error "Debe llenar los campos".	mensaje de error.

Tabla 11. SC 3 Terminar conexión

ID del escenario	Escenario	Var 1	Respuesta del sistema	Resultado de la prueba
EC 1.1	Terminar conexión exitosa	V Seleccionar conexión	El sistema termina la conexión seleccionada y la misma se elimina de la lista de conexiones disponibles.	Se terminó la conexión de forma satisfactoria.
EC 1.2	Terminar conexión fallida	I No seleccionar conexión	Muestra el mensaje de error "Seleccione alguna de las conexiones disponibles".	Se muestra un mensaje de error.

4.4.1.2. Diseño para CU Ejecutar proceso

Tabla 12. Secciones a probar del caso de uso Ejecutar proceso.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo central
SC 1 Ejecutar proceso	EC 1.1 Ejecutar proceso exitosamente	El especialista geográfico selecciona el botón Ejecutar y el sistema ejecuta el proceso.	Formulario/Ejecutar
	EC 1.2 Ejecutar proceso fallida	El especialista geográfico introduce incorrectamente los parámetros asociados a la ejecución del proceso.	Formulario/Ejecutar

4.4.1.3. Diseño para CU Listar procesos

Tabla 13. Secciones a probar del caso de uso Listar procesos.

Nombre de la	Escenario de la	Descripción de la funcionalidad	Flujo central

sección	sección		
SC 1 Listar procesos	EC 1.1 Listar procesos satisfactoriamente	El especialista geográfico selecciona la opción en el menú principal Nueva conexión el sistema muestra el listado de todos los procesos disponibles.	Clic en Archivo/Nueva conexión

4.4.1.4. Diseño para CU Obtener información

Tabla 14. Secciones a probar del caso de uso Obtener información.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo central
SC 1 Obtener información	EC 1.1 Obtener información exitosa	El especialista geográfico selecciona el proceso del cual desea la información y oprime el botón Ver Información, seguidamente el sistema muestra un formulario con los datos necesario para la ejecución del proceso.	Formulario/Ver Información
	EC 1.2 Obtener información fallida	El especialista geográfico no selecciona ningún proceso disponible.	Formulario/Ver Información

Diseño de las pruebas para cada sección del caso de uso Obtener información

SC 1 Obtener información

Variables

Var 1: Seleccionar proceso

Tabla 15. SC 1 Obtener información

ID del escenario	Escenario	Var 1	Respuesta del sistema	Resultado de la prueba
EC 1.1	Obtener información de forma exitosa	V Seleccionar proceso	El sistema muestra en un formulario la información necesaria para la ejecución del proceso	Se mostró la información correctamente.

			seleccionado.	
EC 1.2	Obtener información fallida	I No seleccionar proceso	El sistema muestra un mensaje de error “No ha seleccionado ningún proceso”.	Muestra el mensaje de error.

4.5. Resultados obtenidos

Luego de haber llevado a cabo todo el proceso de pruebas en este epígrafe se hace indispensable documentar los resultados obtenidos. Con la aplicación de las pruebas de caja negra fue posible detectar que el sistema no muestra en ocasiones el resultado en el QGIS, pues es necesario para que se muestren contar con un servidor WPS correctamente instalado y configurado, además es necesario tener conocimientos básicos sobre los procesos para poder llenar debidamente los parámetros necesarios para la ejecución de los procesos. Se detectaron algunas faltas de ortografías en los mensajes que muestra la aplicación. Estas fueron las únicas deficiencias detectadas en la aplicación, por otra parte todas las funcionalidades probadas fueron satisfactorias.

4.6. Conclusiones parciales

Como resultado de la implementación, se logró desarrollar todas las funcionalidades del plugin qWPS y se integró satisfactoriamente con el QGIS. Se realizaron pruebas de caja negra al plugin y se comprobó que sus funcionalidades son operativas. Al analizar los resultados obtenidos por las pruebas, se puede decir que el plugin cumple con la calidad requerida, además cuenta con una interfaz intuitiva y amigable para el especialista geográfico.

CONCLUSIONES GENERALES

Como conclusiones se comentan algunas reflexiones sobre la experiencia de desarrollo, las contribuciones de qWPS y los principales problemas encontrados.

- ✓ Se implementó un cliente WPS para que QGIS pudiera consumir servicios WPS ampliando sus funcionalidades.
- ✓ El manejo adecuado de las herramientas seleccionadas para el desarrollo de la aplicación han facilitado el trabajo y mejorado los resultados, pero exige preparación.
- ✓ El estándar WPS puede resultar en un futuro un modo sencillo y eficaz de publicar y consumir procesos geoespaciales. No obstante, hoy en día el número de servidores WPS es prácticamente nulo y la tecnología aún no se ha desarrollado ni explotado debidamente.
- ✓ Se comprobó el funcionamiento del plugin a partir de las pruebas de caja negra, realizadas mediante los casos de prueba, verificando que los requisitos definidos han sido totalmente satisfactorios.
- ✓ Es necesario la instalación de un servido WPS para que pudieran ejecutarse correctamente los procesos.
- ✓ La instalación y configuración de un servidor WPS es bastante complicado y conlleva a un estudio previo de todas las tecnologías referentes, además de que este tiene una gran cantidad de dependencias.

RECOMENDACIONES

- ✓ Implementar un servidor WPS que permita una integración completa con el plugin en vistas a mejorar el desempeño y la calidad de los procesos.
- ✓ Realizar pruebas de aceptación por parte de un especialista con un servidor local PyWPS que soporte el estándar WPS versión 1.0.0.
- ✓ Implementar un proxy al software que permita la conexión a internet.

Bibliografía

1. **Aronoff, Stan.** *Geographical Information Systems: A management perspective.* s.l. : WDL, 1987.

2. **Worboys, M.F.** *GIS: A Computing Perspective.* London. : Taylor Francis.

3. **OSGeo.** QGIS. [Online] [Cited: Enero 10, 2011.] http://www.qgis.org/.

4. **Inc, Open Geospatial Consortium.** *OpenGIS Web Processing Service.* s.l. : Peter Schut, 2007. OGC 05-007r7.

5. **OGC.** Web Processing Service. [Online] 2007. [Cited: Enero 10, 2011.] http://www.opengeospatial.org/standards/wps.

6. **Masó, J. and Pons, Xavier.** *Del SIG de escritorio al entorno cliente-servidor con Web Processing Service.* Barcelona : s.n., 2010.

7. **Duque, Raúl Gonzáles.** *Python para todos.* España : s.n.

8. **Nokia.** Framework Qt. [Online] http://qt.nokia.com/.

9. Riverbank Computing. [Online] http://www.riverbankcomputing.co.uk/.

10. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* s.l. : Prentice Hall.

11. Visual Paradigm for UML. *Visual Paradigm for UML.* [Online] [Cited: 2 1, 2011.] http://www.visual-paradigm.com/.

12. **Appcelerator, Inc.** PYDEV. [Online] [Cited: 1 23, 2011.] http://www.pydev.org.

13. The Eclipse Foundation. *The Eclipse Foundation.* [Online] [Cited: 2 1, 2011.] http://www.eclipse.org.

14. **Fernández, Carlos Rafael Ballestero.** *Arquitectura de Software del Sistema de Gestión de Información Intranet 2 .* Ciudad de la Habana : Universidad de las Ciencias Informáticas, Julio 2007 .

15. **Letelier Torres, Patricio Orlando and Penadés, Mª Carmen.** *MÉTODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE: EXTREME PROGRAMMING (XP).* Universidad Politécnica de Valencia (UPV) : Departamento de Sistemas Informáticos y Computación , 2006. Vol. 5. ISSN 1666-1680.

16. **Beck, Kent.** *Extreme Programming Explained. Embrace Change.* s.l. : Addison-Wesley Professional, 2000. 0201616416.

17. **Jeffries, Ron, Anderson, Ann and Hendrickson, Chet.** *Extreme Programming Installed.* 2001.

18. **IBM.** Classic RUP form SOMA.es. [Online] [Cited: 12 2, 2010.] http://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/index.htm.

19. **Larman, Craig.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objeto.*

20. **Mc-Graw-Hill.** *Pressman, R.Ingenieria del software: un enfoque práctico.* 2002.

21. GeoSoft. [Online] [Cited: Octubre 30, 2010.] www.geosoft.com.

22. GeoSoft Latinoamerica. *GeoSoft Latinoamerica.* [Online] [Cited:] http://www.geosoft.com/global/latino/sp/index.asp.

23. **Pressman, Roger S.** *Ingenieria de Software un enfoque practico.* s.l. : Mc Graw Hill.

24. **Dijsktra, E.** *Co-coperating sequencial processes .* New York : F. Genuys , 1968.

25. **1471-2000(2000), IEEE Std.** *Recommended Practice for Architectural Description of Software Systems.* s.l. : IEEE Computer Society, September, 2000.

26. **Brook, FP.** *The Mythical Man Month: Essays on Software Engineering.* 1975. 0201835959.

27. **Brown, Malveau and McCorrmick Mowbray, Wiley.** *AntiPatterns. Refactoring Software, Architecture and Projects in Crisis.*

28. **Jacobson, I, Booch, G and Rumbaugh, J.** *El Proceso Unifeicado de Desarrollo de Software.* 84-7829-036-2.

29. **Monroe, R. T, et al., et al.** *Stylized Architecture, Desing Patterns, and Objects .* 1996.

30. **Reynoso, Carlos and Kicillof, Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft .* Universidad de Buenos Aires : s.n.

31. **Chorley, Ribble.** *Handling Geographic Information.* 1987.

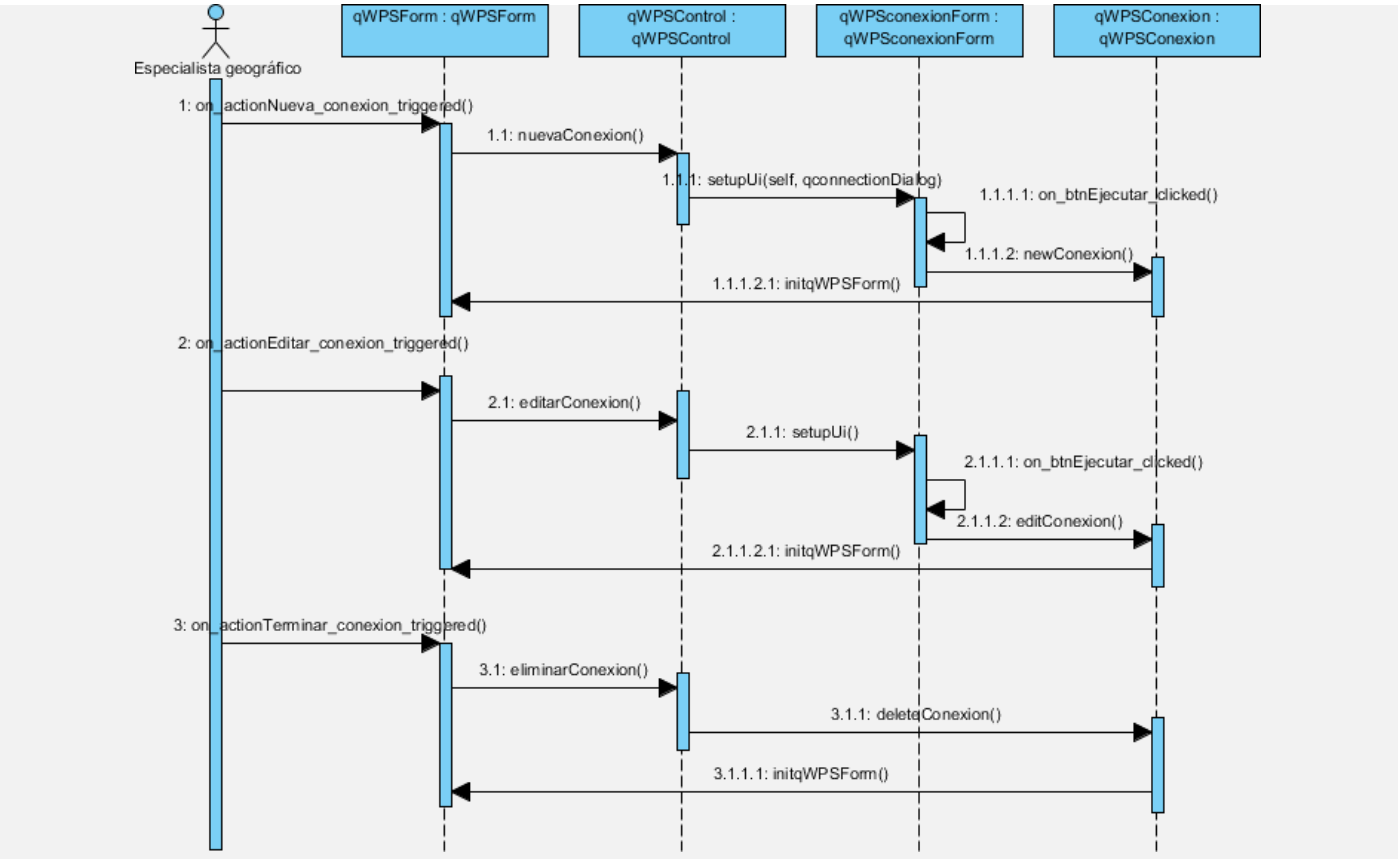
32. **Peng, Zhong-Ren y Ming-Hsiang Tsou.** *Internet GIS: Distributed.* s.l. : Wiley, 2003.

33. **OGC.** OpenGIS Web Map Service. [Online] 2006. [Cited: Enero 11, 2011.] http://www.opengeospatial.org/standards/wms.

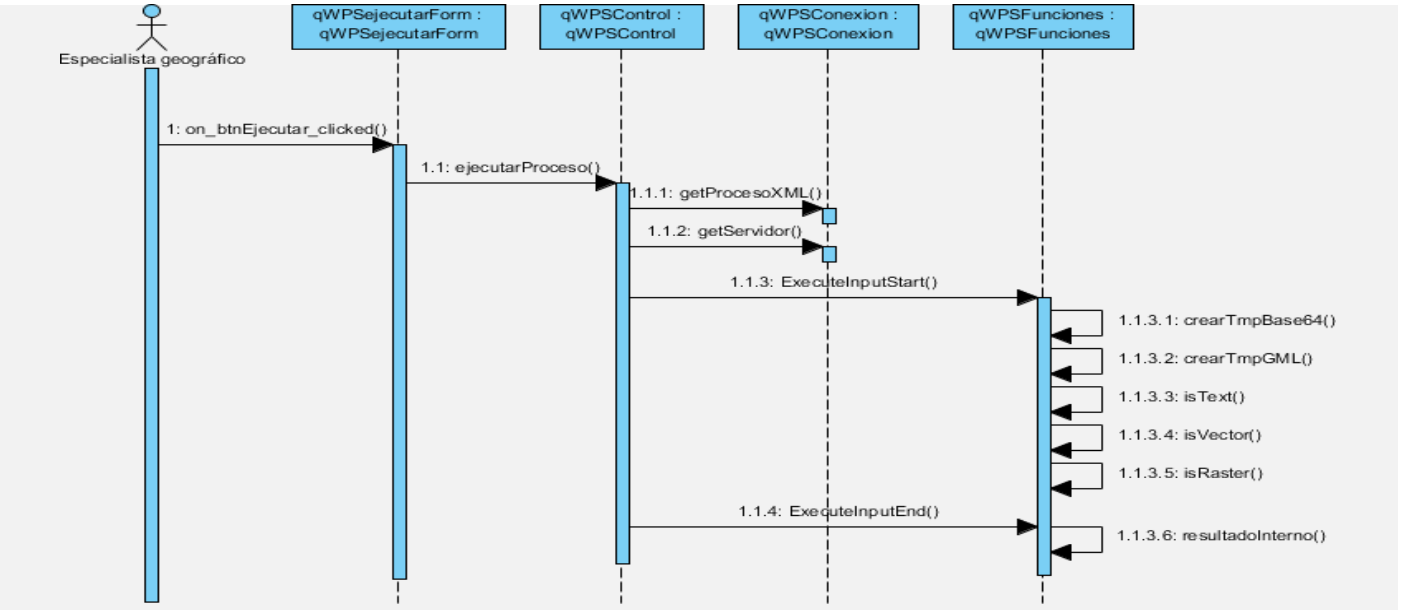
34. —. OpenGIS Geography Markup Language. [Online] 2003. [Cited: Enero 11, 2011.] http://www.opengeospatial.org/standards/gml.

ANEXOS

Anexo 1: Diagrama de secuencia caso de uso Gestionar conexión.



Anexo 2: Diagrama de secuencia caso de uso Ejecutar proceso.


Especialista geográfico
qWPSejecutarForm : qWPSejecutarForm
qWPSControl : qWPSControl
qWPSConexion : qWPSConexion
qWPSFunciones : qWPSFunciones
1: on_btnEjecutar_clicked()
1.1: ejecutarProceso()
1.1.1: getProcesoXML()
1.1.2: getServidor()
1.1.3: ExecuteInputStart()
1.1.3.1: crearTmpBase64()
1.1.3.2: crearTmpGML()
1.1.3.3: isText()
1.1.3.4: isVector()
1.1.3.5: isRaster()
1.1.4: ExecuteInputEnd()
1.1.3.6: resultadoInterno()

GLOSARIO

GML	Geography Markup Language (Lenguaje de Marcado Geográfico). Basado en XML, usado para el transporte y almacenamiento de información geográfica.
OGC	Open Geospatial Consortium. Organización internacional participada por empresas e instituciones del sector de la información geográfica enfocada a la definición de estándares de interoperabilidad (datos y procesos).
XML	Extensible Markup Language, es un formato sencillo, texto muy flexible. Desempeña un papel importante en el intercambio de una amplia variedad de datos en la Web y en otros lugares.
HTTP	Hypertext Transfer Protocol, es un protocolo de capa de aplicación para sistemas distribuidos, de colaboración, los sistemas de información hipermedia.
SIG	Sistemas de Información Geográfica / Geographic Information Systems
WFS	Web Feature Service. Servicio del OGC para acceder y manipular “Fenómenos” (entidades geográficas), que utiliza GML
WMS	Web Map Service. Servicio del OGC para la obtención de imágenes de mapas, habitualmente en formatos como PNG, GIF o JPEG.
WPS	Web Processing Service. Servicio del OGC para la ejecución remota de geoprocesos