

Esta obra está bajo una licencia Reconocimiento-No comercial-Sin obras derivadas 2.5 España de Creative Commons. Puede copiarlo, distribuirlo y transmitirlo públicamente siempre que cite al autor y la obra, no se haga un uso comercial y no se hagan copias derivadas. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.

Diseño y Desarrollo de un cliente Web Processing Service (WPS) para gvSIG

MEMORIA

María del Bueyo Ruiz Lasanta
Ingeniería Informática
Trabajo final de carrera
Curso 2008/09. Primer semestre

Resumen

Esta memoria resume el trabajo realizado para el Proyecto de Fin de Carrera en el área de Sistemas de Información Geográfica. El proyecto se ha centrado en el estudio de los estándares definidos por el Open Geospatial Consortium, y más concretamente en el estándar Web Processing Service (WPS). Así mismo, ha tenido una componente práctica que ha consistido en el diseño y desarrollo de un cliente capaz de consumir servicios Web creados según WPS e integrado en la plataforma gvSIG.

La cartografía es la ciencia que estudia los mapas geográficos y ha sido utilizada por la mayoría de los pueblos de nuestra civilización. Su evolución desde los años sesenta hacia lo que hoy conocemos con Sistemas de Información Geográfica, ha ido de la mano de las tendencias tecnológicas de cada momento. El actual desarrollo de la Web 2.0, también está teniendo un reflejo en estos sistemas. La definición de Servicios Web relativos a Información Geográfica está generando una amplia difusión de aplicaciones GIS y el acercamiento de las capacidades tradicionales de las aplicaciones de escritorio a entornos Web.

La aportación de nuevos estándares realizada por el Open Geospatial Consortium (OGC) con el fin de conseguir la interoperabilidad de los sistemas de geoprocesamiento, ha hecho posible desarrollar servicios de publicación de mapas e integrarlos con las herramientas de trabajo habituales en este campo. En este proyecto se ha estudiado la plataforma gvSIG y los servicios OGC que actualmente incorpora, bien para consumirlos como cliente o bien para ofrecerlos como servidor.

El estándar WPS ha supuesto un paso más en la evolución de estos servicios, pasando de compartir información a compartir procesamiento. El cliente WPS diseñado pretende procesar datos geográficos almacenados de gvSIG en un servidor ajeno a esta plataforma. Esta memoria detalla los aspectos técnicos que han permitido el desarrollo de dicho cliente, así como los inconvenientes surgidos durante el trabajo que han impedido realizar una integración completa con dicho estándar.

Índice de contenidos

Capítulo 1	Introducción.....	6
1.1	Objetivos.....	6
1.2	Estructura del Proyecto.....	7
1.3	Planificación de hitos y tareas.	8
Capítulo 2	Sistemas de Información Geográfica.....	14
2.1	Los SIG y su evolución.	15
2.2	Componentes de un SIG.	17
2.3	Funciones de los SIG.....	17
2.4	Topologías en los SIG.....	19
Capítulo 3	Introducción a los estándares del Open Geospatial Consortium.....	21
3.1	Aspectos generales.	22
3.2	OpenGIS® Web Services.....	24
Capítulo 4	Estándar Web Processing Service. Análisis comparativo de los diferentes proyectos	26
4.1	Web Processing Service 1.0.0.....	26
4.2	Aspectos generales de WPS.....	27
Capítulo 5	El software gvSIG. Instalación y configuración. Principales características	45
5.1	Introducción a gvSIG.....	45
5.2	Requisitos de Instalación.....	48
5.3	Nociones básicas sobre las funcionalidades de gvSIG.....	49
5.4	gvSIG y OGC.....	52
Capítulo 6	Diseño de un cliente Web Processing Services para gvSIG. Catálogo de requisitos, casos de uso y diseño de la aplicación.....	57
6.1	Catálogo de requisitos.....	57
6.2	Diseño del cliente WPS.....	59
Capítulo 7	Desarrollo de la aplicación. Cuestiones fundamentales de la implementación realizada.....	66
7.1	Extensión ConnectClient.....	68
7.2	Extensión GeoprocessRemoteManagerExtension.....	69
7.3	Mejoras propuestas.....	72
Capítulo 8	Conclusiones.....	73
	Lista de referencias.....	75
	Bibliografía	76
	Anexo I. Manual de usuario.....	77

Índice de figuras

Fig. 1. Flujo de trabajo de un SIG	18
Fig. 2. Modelos de representación de la información.....	19
Fig. 3. Pila de protocolos establecida en el marco de referencia del OGC	24
Fig. 4. Modelo de referencia de la arquitectura de las IDEs e INSPIRE.....	27
Fig. 5. Modelo UML del interface WPS	29
Fig. 6. Arquitectura de Deegree WPS	39
Fig. 7. Aspecto visual del cliente WPSint	40
Fig. 8. Modo de trabajo de PyWPS con GRASS GIS	41
Fig. 9. Procesamiento basado en varios servicios OGC desde un cliente uDIG.	41
Fig. 10. Arquitectura gvSIG	48
Fig. 11. Vista del proyecto gvSig en eclipse.	49
Fig. 12. Interfaz gvSIG	49
Fig. 13. Árbol de extensiones de la ventana de preferencias de gvSIG	52
Fig. 14. Conexión a un servidor WFS desde gvSIG.....	53
Fig. 15. Capa WMS obtenida del servidor de la D.G. del Catastro.....	53
Fig. 16. Ventana del gestor de proyectos de gvSIG con extensión de publicación.....	54
Fig. 17. Pantalla gvSIG para la publicación de servicios	56
Fig. 18. Casos de uso	62
Fig. 19. Diagrama de clases de la extensión ExtWPSClient	63
Fig. 20. Pantalla de eclipse con proyecto ExtWPSClient.....	66
Fig. 21. Pantalla de dependencias de la extensión ExtWPSClient en gvSIG.....	68
Fig. 22. Pantalla de conexión a un servidor WPS	69
Fig. 23. Presentación del árbol de Geoprocesos.....	70

Capítulo 1

Introducción

Este capítulo expone el marco en el que se ha desarrollado el presente Proyecto de Fin de Carrera (PFC). Se presentan los objetivos del proyecto, la planificación de las principales tareas, la metodología utilizada durante su ejecución y los productos obtenidos. Por último se incluye una breve descripción del contenido del resto de los capítulos de la memoria.

1.1 Objetivos

El objetivo general del PFC es abordar un trabajo de manera individual que permita llevar a la práctica los conocimientos, tanto técnicos como transversales, adquiridos en las diferentes asignaturas de la carrera. En concreto, en el proyecto han sido de gran utilidad los contenidos de Metodología de gestión de Proyectos TIC, de Ingeniería del software de componentes y sistemas distribuidos y de Compiladores II. Por otro lado, ha sido necesario el estudio de otros aspectos específicos para desarrollar este trabajo.

Este proyecto se enmarca en el área de Sistemas de información geográfica (SIG) y se centra en el estudio de los estándares definidos por el Open Geospatial Consortium, y en el diseño y desarrollo de un cliente Web Processing Service para gvSIG.

Los objetivos específicos de este proyecto son:

- Conocer que es un Sistemas de Información Geográfica (SIG).
- Conocer la herramienta gvSIG y configurar su entorno de desarrollo en el IDE Eclipse.
- Conocer los estándares más extendidos del Open Geospatial Consortium (OGC) y en especial el estándar Web Processing Service (WPS).
- Realizar un análisis comparativo de las diferentes implementaciones existentes en la actualidad del estándar WPS.
- Analizar, diseñar y desarrollar un cliente capaz de consumir servicios WPS.

1.2 Estructura del Proyecto

El proyecto se divide en una parte de estudio del estándar WPS y una parte práctica para llevar a cabo el desarrollo del cliente WPS para gvSIG.

Durante la ejecución del proyecto se han realizado tres entregas de documentos para las pruebas de evaluación continuas, con sus correspondientes informes de seguimiento, resumen de la ejecución de los trabajos realizados.

A continuación se detallan los productos obtenidos como resultado final del proyecto.

1.2.1 Memoria del proyecto.

Describe el trabajo realizado para el diseño y desarrollo del cliente WPS. El contenido de la memoria se ajustará a los siguientes capítulos:

- Capítulo 1. Introducción.
- Capítulo 2. Breve historia de los Sistemas de Información Geográfica.
- Capítulo 3. Introducción a los estándares del Open Geospatial Consortium.
- Capítulo 4. Estándar Web Processing Service. Análisis comparativo de los diferentes proyectos.
- Capítulo 5. El software gvSIG. Instalación y configuración. Principales características.
- Capítulo 6. Diseño de un cliente Web Processing Service para gvSIG. Catálogo de requisitos, casos de uso y diseño de la aplicación.
- Capítulo 7. Desarrollo de la aplicación. Cuestiones fundamentales de la implementación realizada. Plan de pruebas. Presentación de la aplicación.
- Capítulo 8. Resultados y conclusiones. Líneas futuro de desarrollo.
- Anexo. Manual de usuario de la extensión WPS para gvSIG.

1.2.2 Presentación audiovisual del proyecto.

Resumen de la memoria que se presenta un audiovisual de 20 minutos de duración máxima. Sintetiza el trabajo realizado, haciendo especial hincapié en los aspectos más relevantes del mismo y en los resultados obtenidos.

1.2.3 Extensión WPS para gvSIG, código fuente e instalación.

Se entrega el código fuente de la extensión desarrollada, comentado según lo establecido para generar su JavaDoc. Así mismo, se entrega los ficheros de creación de la misma.

1.3 Planificación de hitos y tareas.

Los principales hitos a cumplir y las tareas principales identificadas para abordar el trabajo conformaron la planificación inicial del proyecto. A continuación se presenta dicha planificación.

1.3.1 Hitos

Los principales hitos del proyecto han estado marcados por las fechas de entrega de las pruebas de evaluación continua. Las entregas y las fechas de estos hitos han sido:

Hito	Entrega	Fecha
PEC1	Plan de Trabajo	30/09/2008
PEC2	Capítulos 2, 3, 4 y 5 Informe de seguimiento	02/11/2008
PEC3	Capítulos 6 y 7 Informe de seguimiento	09/12/2008
PEC4	Presentar proyecto. Memoria y presentación	05/01/2009

1.3.2 Tareas

A continuación se describen brevemente las tareas abordadas durante el proyecto y la duración estimada.

1.3.2.1 Tarea 1ª – Definir proyecto.

Leer el enunciado del proyecto y extraer los puntos fundamentales. Tras esta primera lectura, se tendrán elementos de búsqueda suficientes para obtener información complementaria que permita dimensionar el proyecto y abordar la planificación del proyecto.

También se realizará la lectura del material complementario facilitado por la Universidad sobre el Proyecto Fin de Carrera.

La duración estimada de la tarea es de 5 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Leer enunciado	1 hora	18/09/2008	18/09/2008
Leer material del TFC	4 horas	19/09/2008	20/09/2008

1.3.2.2 Tarea 2º – Elaborar Plan de Trabajo.

Esta tarea tiene una doble finalidad: estructurar el contenido del proyecto e identificar los productos resultantes al final del mismo, y planificar las tareas y estimar el esfuerzo requerido para su realización.

La planificación de las tareas es difícil de realizar debido a que en su realización intervienen una gran cantidad de factores inciertos o desconocidos en esta fase inicial. En primer lugar se identificarán las tareas a realizar para proceder posteriormente a la redacción del Plan de Trabajo. Se entregará al consultor un borrador de dicho Plan y una vez recibidos sus comentarios se procederá a su corrección.

Como resultado de esta tarea se obtendrá el Plan de Trabajo del Proyecto.

La duración estimada de la tarea es de 15 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Planificar tareas	5 horas	20/09/2008	22/09/2008
Redactar Plan de Trabajo	6 horas	23/09/2008	25/09/2008
Corregir del Plan de Trabajo	4 horas	26/09/2008	29/09/2008

1.3.2.3 Tarea 3ª – Instalar software.

Esta tarea se simultaneará con las dos anteriores. Todo el software será descargado de Internet y se procederá posteriormente a su instalación y configuración. En aquellos casos en los que el software ya esté instalado, se estudiará si las versiones son las adecuadas y se procederá a su actualización si se estima necesario.

La duración estimada de la tarea es de 10 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Instalar software	10 horas	20/09/2008	30/09/2008

1.3.2.4 Tarea 4º – Estudiar estándar WPS y gvSIG.

Se comienza en esta tarea a profundizar en tema principal del proyecto, realizando en primer lugar una búsqueda de información de interés partiendo de la bibliografía descrita en el apartado 5 de este documento.

El estudio se iniciará con una introducción a los Sistemas de Información Geográfica, haciendo un repaso sobre la evolución de los mismos desde su nacimiento hasta su situación actual. Posteriormente, se conocerán los principales estándares definidos por el OGC, analizando con mayor detalle algunos de ellos como son los de Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Catalogue Service (CSW) y fundamentalmente el estándar de Web Processing Service (WPS).

Así mismo, se analizarán algunas iniciativas de las Administraciones Públicas y Entidades sin ánimo de lucro encaminadas a promover el uso de estos estándares, y más concretamente, el desarrollo de la plataforma gvSIG promovido desde la Conselleria de Infraestructuras y Transporte de la Generalitat Valenciana. En este punto, se estudiará la arquitectura de gvSIG y sus tres subsistemas principales, Andami, Fmap y subdriver.

Por último, se evaluarán diferentes proyectos que implementan el estándar WPS, realizando un análisis comparativo de los mismos y se procederá a la instalación y configuración de uno de estos proyectos.

Una vez finalizada la tarea se dispondrá de información suficiente para redactar los capítulos 2, 3, 4 y 5 de la Memoria del Proyecto, así como del resultado de la evaluación de las alternativas existentes actualmente para abordar el desarrollo la siguiente tarea.

La duración estimada de la tarea es de 44 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Buscar información	4 horas	01/10/2008	02/10/2008
Estudiar los SIG	2 horas	03/10/2008	03/10/2008
Estudiar estándares OGC	7 horas	03/10/2008	05/10/2008
Estudiar estándar WPS	12 horas	06/10/2008	11/10/2008
Familiarizarse con gvSIG	3 horas	12/10/2008	12/10/2008
Instalar proyecto	3 horas	13/10/2008	15/10/2008
Evaluar alternativas	3 horas	16/10/2008	16/10/2008

1.3.2.5 Tarea 5º – PEC 2

La entrega correspondiente a la segunda prueba de evaluación continua contiene un documento sobre el resultado de la tarea anterior junto con un informe de seguimiento de la evolución del proyecto.

La duración estimada de la tarea es de 18 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Preparar borrador PEC2	12 horas	17/10/2008	26/10/2008
Corregir PEC2	6 horas	31/11/2008	02/11/2008

1.3.2.6 Tarea 6º – Desarrollar cliente WPS

Esta tarea consiste en el desarrollo de un cliente WPS en el entorno gvSIG. Se abordaran las fases de análisis, diseño e implementación contempladas en el ciclo de vida del software.

En la fase de análisis, se establecerán los requisitos del cliente a desarrollar y se elaborará un catalogo detallado y conciso de los mismos. Así mismo, se especificarán los casos de uso asociados a los requerimientos funcionales.

El diseño de la aplicación se realizará partiendo de la arquitectura de gvSIG. En este momento se diseñarán las clases necesarias para la implementación del cliente teniendo en cuenta las especificaciones del servidor instalado en la tarea 4.

La implementación de la aplicación se realizará en Java, dentro del entorno gvSIG y utilizando como IDE eclipse.

La tarea concluirá con la prueba de la aplicación y con la elaboración del manual de usuario

La duración estimada de la tarea es de 57 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Elaborar catalogo de requisitos	4 horas	27/10/2008	30/10/2008
Diseñar casos de uso	2 horas	30/10/2008	30/10/2008
Diseñar de la aplicación	15 horas	03/11/2008	09/11/2008
Implementar de la aplicación	30 horas	10/11/2008	23/11/2008
Realizar pruebas	3 horas	24/11/2008	26/11/2008
Redactar Manual Usuario	3 horas	27/11/2008	27/11/2008

1.3.2.7 Tarea 7º – PEC 3

La entrega correspondiente a la tercera prueba de evaluación continua contiene la documentación del desarrollo del cliente realizado en la tarea anterior, y que corresponde a los capítulos 6 y 7 de la Memoria del Proyecto y un informe de seguimiento de la evolución del proyecto.

La duración estimada de la tarea es de 20 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Preparar borrador PEC3	15 horas	28/11/2008	04/12/2008
Corregir PEC3	5 horas	07/12/2008	09/12/2008

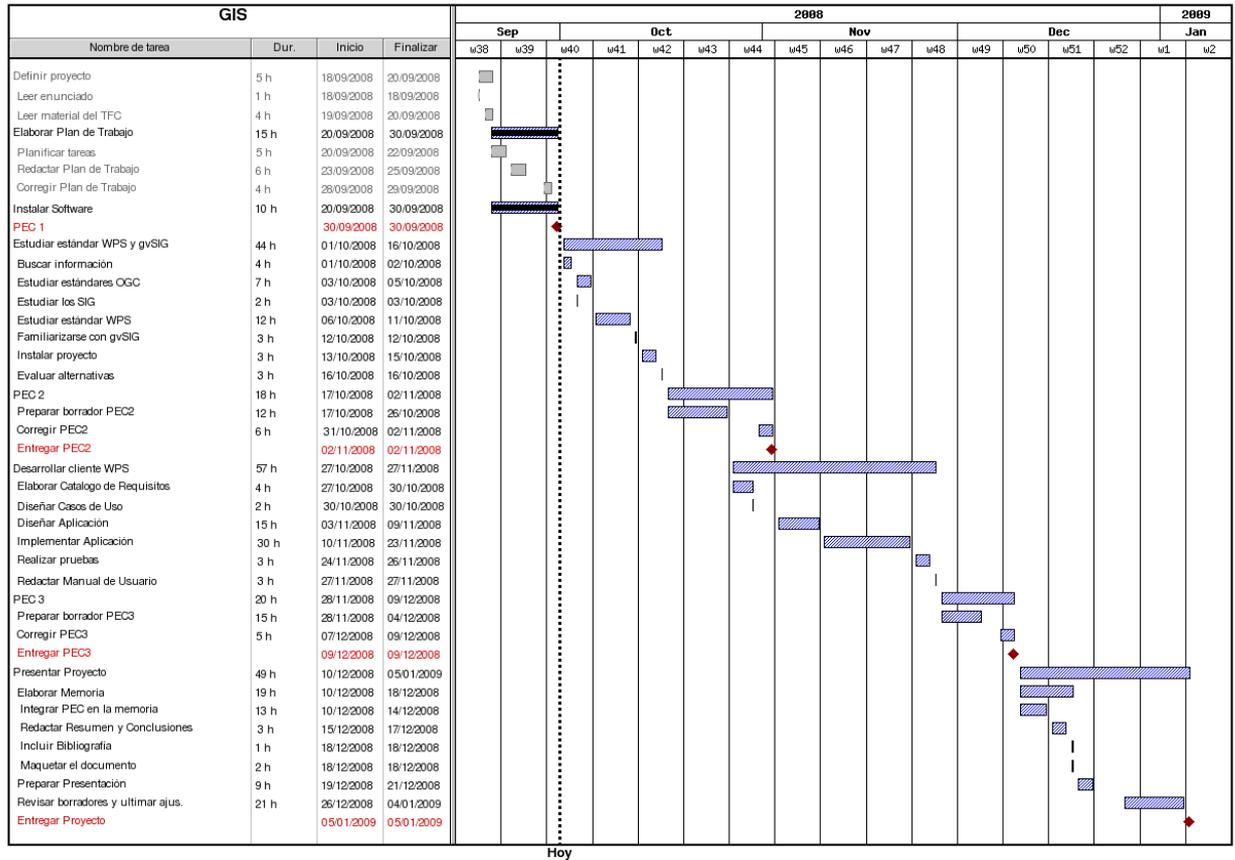
1.3.2.8 Tarea 8º – Presentar Proyecto

Esta tarea se centra en la redacción y maquetación de la Memoria del Proyecto, y en la presentación del proyecto mediante la elaboración un audiovisual. Así mismo, se ha previsto dedicar un número de horas suficiente para realizar las revisiones del material entregable y ultimar los ajustes del proyecto.

La duración estimada de la tarea es 49 horas.

	Duración	Fecha prevista de inicio	Fecha prevista de fin
Elaborar de la memoria			
Integrar las PEC de la memoria	13 horas	10/12/2008	14/12/2008
Redactar Resumen y Conclusiones	3 horas	15/12/2008	17/12/2008
Incluir Bibliografía	1 hora	18/12/2008	18/12/2008
Maquetar el documento	2 horas	18/12/2008	18/12/2008
Preparar presentación	9 horas	19/12/2008	21/12/2008
Revisar borradores y ultimar ajustes	21 horas	26/12/2008	04/01/2008

A continuación se muestra el gráfico de Gantt con la planificación completa del proyecto. La duración total del proyecto será de 208 horas. En el gráfico aparecen marcadas las tareas completadas en el momento de entregar el presente Plan de Trabajo.



1.3.3 Calendario de trabajo.

Este proyecto se ha abordado de manera individual, con una dedicación parcial, junto con las tareas de otra asignatura de la Ingeniería Informática, el trabajo laboral y cuestiones familiares.

La previsión de horas dedicadas al proyecto fue de 15 horas semanales atendiendo a la siguiente distribución:

- De lunes a miércoles: 3 horas en total.
- De jueves y domingo: 3 horas diarias.

Se prevé una dedicación uniforme durante la duración del proyecto, salvo la semana del 20 de octubre, que se reducirá a 6 horas debido a un viaje por motivos familiares. Con el fin de corregir posibles desviaciones en la planificación o a imprevistos personales, no se han tenido en cuenta una mayor disponibilidad durante el mes de diciembre debida a días festivos y de vacaciones laborales.

Capítulo 2

Sistemas de Información Geográfica

La Cartografía se define como el arte de trazar mapas geográficos y la ciencia que los estudia. La información espacial o geográfica ha sido utilizada por la mayoría de los pueblos de nuestra civilización y su evolución ha ido de la mano de los avances producidos en otras ciencias.

Según los historiadores, ya los fenicios, expertos navegantes, exploradores y estrategias militares, recopilaban información y los plasmaban en dibujos que pueden considerarse los primeros mapas. Por su parte, los griegos desarrollaron las matemáticas y la astronomía, realizaron mediciones de distancias sobre la superficie de la Tierra basadas en el conocido Teorema de Pitágoras, y registraron esta información en forma de mapas.

Estos pueblos y los posteriores, utilizaron esta información con fines logísticos y estratégicos. Sin embargo hasta el siglo XVIII no se reconoció la importancia de organizar y sistematizar la información geográfica. En ese momento, se crearon los primeros organismos comisionados exclusivamente para ejecutar la recopilación de información y producir mapas topográficos a nivel de países enteros.

Ya en el siglo XIX, se produce un avance en el conocimiento científico de la tierra, y se cartografió una gran cantidad de información de las características de la Tierra. Fue en 1854, cuando el Dr. John Snow comenzó a utilizar la técnica de la superposición de mapas temáticos sobre un mapa topográfico base, cartografiando la incidencia de los casos de cólera en un mapa del distrito de SOHO en Londres. Este método, que permitió a Snow localizar con precisión un pozo de agua contaminado como fuente causante del brote, siguió siendo utilizado hasta bien entrado el siglo XX.

A partir de la segunda mitad del siglo XX, se comienza a utilizar los mapas para la valoración de recursos y planificación, y se empieza a aplicar la tecnología del computador digital en la automatización del trabajo cartográfico. También se producen otros avances, como la aparición de la fotografía aérea y de las imágenes de satélites originando un desarrollo de herramientas tecnológicas de procesamiento espacial para estudios físicos, matemáticos y científicos. Sin embargo, el concepto de la

representación visual de esta información no era el mismo que el que se tenía en cartografía.

Las personas que iniciaron el desarrollo de estas tecnologías, desconocían realmente que lo que estaban creando eran Sistemas de Información Geográfica.

2.1 Los SIG y su evolución.

Durante los años sesenta, emergieron nuevas tendencias en la forma de utilizar los mapas para la valoración de recursos y planificación. Es entonces cuando se empieza a hablar de los Sistemas de Información Geográfica (SIG). Dándose cuenta de que las diferentes coberturas sobre la superficie de la tierra no eran independientes entre si, sino que guardaban algún tipo de relación, se hizo latente la necesidad de evaluarlos de una forma integrada y multidisciplinaria, y se comenzó a aplicar tecnología informática. En esta primera etapa, los trabajos se realizan de manera individual y para propósitos específicos.

A partir de ese momento, las instituciones y otras instancias gubernamentales ponen en marcha sus iniciativas de manera particular para resolver los problemas derivados de sus actividades relacionadas con el tratamiento de información geográfica. También las universidades intentan encontrar nuevos métodos para el tratamiento de la información espacial. Buscan el desarrollo de sistemas automáticos con los que realizar análisis de datos geográficos y no tanto la producción de cartografía. En ambos casos, el objetivo es conseguir unas aplicaciones que de forma automática resuelvan cuestiones que hasta la fecha se habían solventado de manera manual.

Estados Unidos ha tenido desde entonces un papel muy significativo en la evolución de los SIG, y de las tecnologías relacionadas, si bien son también destacables iniciativas de otros países como Canadá y Gran Bretaña.

Fue Canadá el primero en utilizar el término Sistema de Información Geográfica, cuando en 1964 el Departamento de Agricultura de Canadá financió el desarrollo de una aplicación específica para el inventariado y planificación del suelo de las vastas extensiones de territorio del país mediante la digitalización y análisis semiautomático de fotografía aérea. El CGIS (Canadian Geographic Information System) ha sido un referente mundial en la historia de los SIG. Algunas de las ideas que aportó están vigentes en software SIG actuales, como por ejemplo la estructuración de la

información en capas temáticas, la división de mapas digitales en hojas y ajustes en los bordes, la topología de arcos o las superposiciones topográficas.

En 1966, Estados Unidos se crea el Laboratory of Computer Graphics and Spatial Analysis (Universidad de Harvard USA). Su objetivo principal era el diseño y desarrollo de software específico para aplicaciones cartográficas. Su principal logro fue el SYMAP (SYagraphic MAPping) una técnica sinagráfica para elaboración de mapas. Este software era capaz de representar información espacial por medio de gráficos generados por ordenador.

Paralelamente, la administración estadounidense desarrolló diversos proyectos mediante programas propios y específicos, si bien comienza a hacer uso de software de terceros. Los proyectos SIG más significativos de esa época son United States Census Bureau (USCB) y United States Geological Survey (USGS), en ambos casos con continuidad en nuestros días.

De entre las iniciativas privadas, cabe señalar la fundación de ESRI (Environmental Systems Research Institute) en 1969 con la misión inicial de organizar y analizar la información geográfica. Su principal aportación es el desarrollo de ArcInfo.

Durante los años 70, el uso de ordenadores progreso rápidamente en el manejo de información cartográfica, y se afinaron muchos de los sistemas informáticos para distintas aplicaciones cartográficas. De la misma manera, se estaba avanzando en los sectores de la topografía, la fotogrametría y la percepción remota. En un principio, este rápido ritmo de desarrollo provocó duplicidad de esfuerzos en las distintas disciplinas relacionadas con la cartografía, a la vez que se adquiría experiencia, en la elaboración automatizada de información espacial.

A principio de los 80, los SIG se habían convertido en un modelo plenamente operativo, a medida que la tecnología se perfeccionaba, se hacía menos costosos y gozaba de una mayor aceptación. A principio de esta década, ESRI crea el primer sistema eficiente de superposición de polígonos y su primer software comercial, ArcInfo, que combina elementos geográficos tales como puntos, líneas y polígonos, con una herramienta de administración de bases de datos para asignar atributos a estos elementos. Por otro lado, el Laboratorio de Harvard, partiendo desarrolló ODYSSEY, un software SIG de tipo vectorial que incluye la digitalización semiautomática, la gestión de la base de datos y la elaboración interactiva de los mapas.

A partir de este momento, confluyen dos circunstancias importantes en la evolución de estos sistemas. Por una parte, comienzan a disminuir las iniciativas individuales y los desarrollos específicos para dar paso a iniciativas corporativas, iniciadas a instancias gubernamentales. Por otra parte, la mejora del rendimiento de los ordenadores, el abaratamiento del software y el uso generalizado de Internet, hacen posible la difusión de los SIG a nivel de usuario doméstico y la generalización de funcionalidades visibles desde cualquier navegador Web. Se consigue un acceso más universal y de bajo coste.

El creciente interés en la utilización de la información geográfica para el desarrollo sostenible, ha llevado a muchos países y organizaciones a adoptar una combinación de técnicas, políticas y mecanismos encaminados a compartir información espacial a través de sus grupos de trabajo. Estos mecanismos son conocidos como Infraestructuras de Datos Espaciales y pretende incentivar la interoperabilidad de conocimiento e información espacial.

El proyecto INSPIRE (INfrastructure for SPatial InforMation in Europe) es la iniciativa de la Unión Europea para la propuesta de normativas de carácter comunitario relacionados con la información geográfica.

2.2 Componentes de un SIG.

Los Sistemas de Información Geográfica son sistemas compuestos por hardware, software y procedimientos para capturar, manejar, manipular, analizar, modelizar y representar datos georreferenciados, con el objetivo de resolver problemas de gestión y planificación. (NCGIA, 1990).

Basándonos en esta definición, los componentes de un SIG son:

- El hardware: Equipos informáticos formados por ordenadores y sus periféricos.
- El software: Programas informáticos con funciones para visualizar, consultar y analizar los datos geográficos. Pueden tener estructura modular o integrada.
- Los datos: Constituyen una representación simplificada del mundo real que ha de recogerse y mantenerse actualizada.
- El personal: Comprende a los usuarios del sistema.

2.3 Funciones de los SIG

Se distingue seis grandes tipos de cuestiones a las que un SIG puede responder.

- Localización: ¿Qué hay en ...?
- Condición: ¿Dónde sucede que ..?
- Tendencias: ¿Qué a cambiado ...?
- Rutas: ¿Cuál es el camino óptimo ..?
- Pautas: ¿Qué pautas existen...?
- Modelos:¿Qué ocurriría si?

Para responder a estas cuestiones, los SIG cumplen una serie de funciones, mostradas en la figura 1, que se pueden agrupar en:

- Funciones de captura de la información. Permiten adquirir y depurar errores de información geográfica y temática preparándola para ser tratada por el ordenador.
- Funciones de gestión. Permiten la estructuración de la información original en diferentes capas de información coherente.
- Funciones de análisis. Facilitan el procesado de los datos y permiten extraer información no presente a simple vista, generar nuevos datos y realizar simulaciones de comportamientos. Son las que dan mayor potencialidad a los SIG.
- Funciones de salida. Permiten mostrar al usuario los datos incluidos en el sistema, el resultado de las consultas y del análisis realizado.

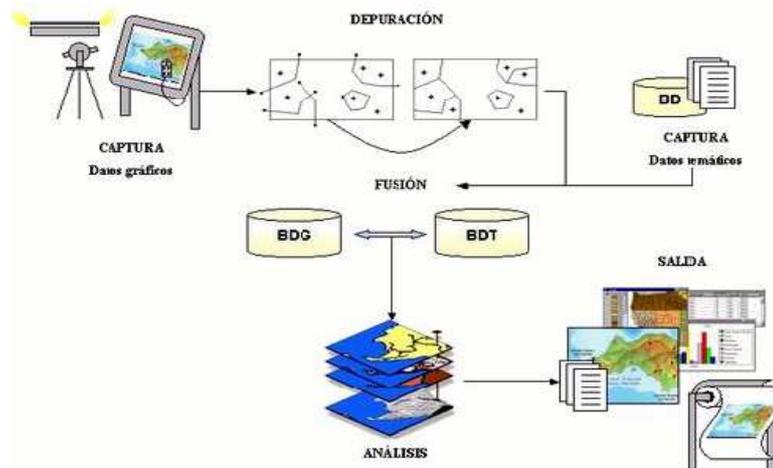


Fig. 1. Flujo de trabajo de un SIG

2.4 Topologías en los SIG

Los SIG deben ser capaces de representar y almacenar las entidades geográficas reales mediante la representación y almacenamiento de entidades gráficas. La figura 2 muestra los dos sistemas básicos que existen para recoger y representar la información geográfica en los SIG: el modelo raster y el modelo vectorial.

En los SIG vectoriales el foco de interés se sitúa en las entidades, en su posicionamiento sobre el espacio. Para modelizar las entidades del mundo real se utilizan objetos espaciales como puntos, líneas o polígonos. Así, para los elementos puntuales se almacenan un par de coordenadas, para los objetos lineales un conjunto de pares de coordenadas y para entidades poligonales, el conjunto de pares de coordenadas que definen el contorno de dicho polígono.

En los SIG raster, el espacio geográfico es dividido en sectores de forma regular denominados comúnmente píxel. De esta forma se establece una malla en la que cada píxel va a tomar el valor de la variable geográfica que se encuentre en esa posición. La resolución espacial va a marcar la exactitud de la información recogida, cuanto más pequeño es el píxel más exacta será la representación de la información. Este modelo se centra más en las propiedades del espacio más que en la precisión de la localización.

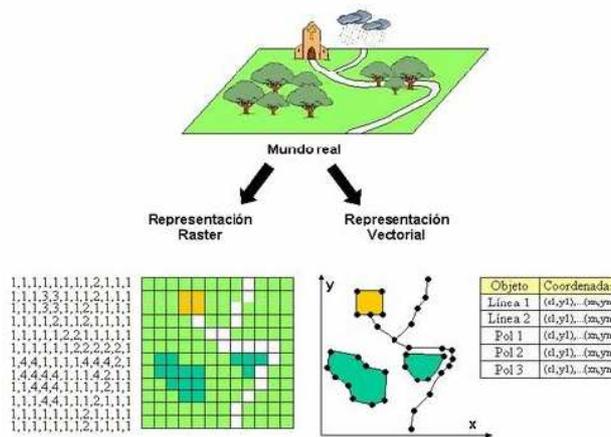


Fig. 2. Modelos de representación de la información.

De forma general se recomienda utilizar información raster para trabajar con grandes extensiones de terreno y a pequeñas escalas, siempre y cuando no se requiera una exactitud muy alta en los cálculos. Y se recomienda utilizar información vectorial para trabajar con extensiones pequeñas y a medianas o grandes escalas, y siempre que se

necesite alta precisión en los cálculos y definición de entidades. La siguiente tabla muestra una comparativa de ambos modelos.

Modelo Vectorial	Modelo raster
Ventajas	
Ocupa menos espacio de memoria	Facilidad de captura
Precisión elevada en la definición de entidades el cálculo de magnitudes geométricas	Estructura de datos simple
Representación adecuada de las relaciones topológicas	Sencillez en la manipulación y gestión de la información
Mejores salidas gráficas	
Inconvenientes	
Captura de datos más costosa	Menor precisión en el cálculo de áreas y longitudes
Estructura de datos más compleja	Ocupan mayor espacio de memoria
Mayor dificultad a la jora de realizar ciertas operaciones (Comparación de mapas)	Dificultad de representar ciertas relaciones topológicas

Capítulo 3

Introducción a los estándares del Open

Geospatial Consortium

Actualmente, las dos organizaciones de normalización principales dentro de los sistemas de información geográfica son el comité técnico 211 de la Organización internacional para la Estandarización (ISO TC211) y el Open Geospatial Consortium (OGC).

En el caso del ISO TC211, se trata de un cuerpo dependiente de la ISO dedicado a la temática de información geográfica y geomática. Las normas ISO resultado de su trabajo y relativas a la información geográfica, se encuentran en el rango de numeración 19.100. Estas normas derivan en su mayor parte de los documentos técnicos desarrollados por el OGC.

En cuanto al Open Geospatial Consortium, fue creado en 1994 y en estos momentos, agrupa a 366 organizaciones públicas y privadas. Algunas de las empresas que forman parte de la organización son Intergraph, ESRI, Oracle, Adobe, Google y la NASA

Su fin es la definición de estándares abiertos e interoperables dentro de los SIG. Persigue acuerdos entre las diferentes empresas del sector para posibilitar la interoperabilidad de sus sistemas de geoprocésamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios. Anteriormente fue conocido como Open GIS Consortium.

El OGC publica diferentes tipos de documentos, principalmente los siguientes:

- OpenGIS® Specifications: Documentos técnicos que detallan interfaces o codificación.
- Specification Profiles: Perfiles de una especificación OGC.
- Abstract Specification: Proporcionan los fundamentos conceptuales para actividades de desarrollo de especificaciones OGC.
- Best Practices Documents: Documentos de debate sobre la implementación de documentos OGC.

- Discussion Papers: Se objetivo es crear discusión en la industria de la información geoespacial sobre un tema específico.

3.1 Aspectos generales.

El OGC originalmente se centró en el uso de plataformas de cálculo distribuido, como CORBA, OLE/COM y SQL. Al extenderse el protocolo http y la Web, el OGC adapta su modelo a las tecnologías emergentes para permitir, mediante los OGC Web Services (OWS), ensamblar aplicaciones a partir de los servicios de procesamiento de datos geográficos y de los servicios de localización disponibles. El objetivo fundamental es que las aplicaciones se puedan crear dinámicamente a partir de servicios descubiertos y enlazados en tiempo de ejecución para ser usados en servicios de información de geográfica y en aplicaciones profesionales.

El modelo de interoperabilidad del OGC reconoce la existencia de caminos alternativos para invocar un servicio y por tanto describe los principios y los modelos básicos para la creación dinámica de sistemas débilmente acoplados. Para ello, proporciona un modelo de referencia para la construcción estos servicios, consistente con los términos y conceptos introducidos en los modelos de referencia para entornos abiertos de procesamiento distribuido (RM-ODP). Este modelo debe ser común para la publicación, localización, conexión y gestión de servicios Web.

Las construcciones conformes a este modelo de referencia deben:

- Soportar múltiples construcciones independientes de un tipo de servicio.
- Soportar la conexión de proveedores independientes para distintas clase de servicios.
- Localizar, en tiempo de ejecución una instancia específica de un servicio basándose en el tipo de servicio, los contenidos accesibles y las características de la calidad del servicio.
- Permitir el acceso a los metadatos que describen espacial y temporalmente los datos referenciados.
- Localizar los servicios que pueden ser utilizados con unos tipos de datos específicos.
- Posibilitar el acceso a los metadatos que describen los servicios, su localización en la Web, los medios para acceder y uso de los mismos.

- La invocación de servicios en tiempo de ejecución para realizar tareas comunes, utilizando los metadatos de descubrimiento.
- Posibilitar los cambios a medida (coordinación) de los servicios para conseguir un control de flujo de un proceso agregado.

Siguiendo el modelo de referencia RM-ODP, el OGC define cinco puntos de vista que abordan diferentes aspectos del sistema, que no se superponen unos a otros, y que permiten la separación de distintos aspectos de interés.

- El punto de vista de la empresa. Se centra en el objetivo, ámbito y política para ese sistema desde la perspectiva del propio negocio.
- El punto de vista de la información. Se centra en la semántica y el procesamiento de la información.
- El punto de vista de computacional. Refleja los detalles de componentes e interfaces sin reparar en la distribución.
- El punto de vista de la ingeniería. Se centra en los mecanismos y funciones que se requieren para secundar la interacción distribuida entre objetos en el sistema.
- El punto de vista de la tecnología. Se centra en la elección de la tecnología.

Para el trabajo que nos ocupa, tendremos en cuenta el punto de vista de la información como marco de información OpenGIS, el punto de vista informático como marco de servicios OpenGIS y el punto de vista de la ingeniería marco de protocolos.

El punto de vista de la información define esquemas conceptuales para la información geoespacial y métodos para definir esquemas de aplicación.

Tradicionalmente, los sistemas de información geográfica han venido tratando dos tipos fundamentales de datos, vectoriales y raster. El OGC se refiere a “features” en el caso de los datos vectoriales, y a “coverages” para los datos tipo raster.

Entrando en el aspecto informático, el OGC define los conceptos fundamentales de los servicios, interfaces y operación, e incluye una clasificación de servicios y una descripción del marco de servicios OGC.

Los servicios OGC con base en la Web están diseñados para facilitar información geográfica en Internet. De esta manera, comparten muchas capacidades y características de servicios Web en general, pero también tienen características singulares que se

adaptan a las necesidades geospaciales. En particular, los OWS por definición se describen a sí mismos y mantienen una operación GetCapabilities que detallan operaciones y contenidos posibles. Se pretende que la respuesta devuelta desde la operación GetCapabilities de un servicio suministre toda la información que un cliente necesite para hacer uso de ese servicio. Puede considerarse un modelo completo de información de servicios, expresado en UML, esquema XML o como casos individuales XML.

El punto de vista de la ingeniería presenta una serie de tecnologías y estándares en los que los servicios pueden ser construidos y desplegados. La figura 3 muestra la pila de protocolos establecida por el OGC. Los niveles de las capas más bajas, permiten la interoperabilidad y son un medio para los mecanismos de publicación, búsqueda y vinculación de los servicios, permitiendo a los componentes software trabajar de forma transparente.

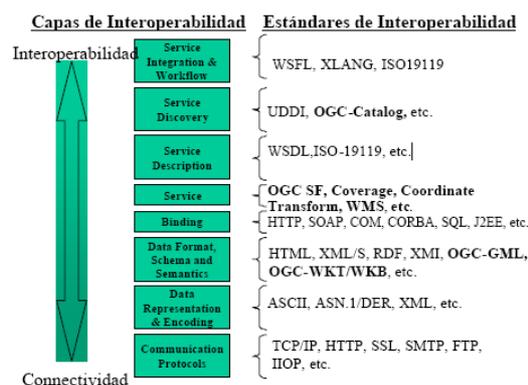


Fig. 3. Pila de protocolos establecida en el marco de referencia del OGC

3.2 OpenGIS® Web Services.

Las especificaciones más importantes surgidas del OGC son los OpenGIS® Web Services (OWS). Entre ellos cabe destacar los siguientes:

- Servicio de mapas (WMS): Produce mapas de forma dinámica a partir de información geográfica vectorial o raster presentando la información como imágenes digitales susceptibles de ser visualizadas en pantalla.
- Servicio de publicación de objetos (WFS): Permite recuperar y modificar datos espaciales en formato vectorial codificados en Geography Markup Language (GML).
- Servicio de coberturas (WCS) : Permite la obtención de datos geospaciales en forma de “coberturas” que representan fenómenos de variación espacial. A

diferencia de los servicios de mapas, estos proporcionan los datos con su semántica original, permitiendo además de su representación estática, su interpretación y extrapolación.

Estos estándares posibilitan el intercambio de información geográfica. En cuanto a la definición de estándares para el procesamiento distribuido, el OGC ha establecido el Web Processing Service (WPS) que veremos en profundidad en el siguiente capítulo.

Por otro lado, el OGC ha definido una interfaz común, denominada Catalog Interface (CI), que permite crear diversas aplicaciones, que cumplan el estándar, para poder descubrir, navegar y realizar operaciones de consulta contra los servidores de catálogo heterogéneos y distribuidos. También se ha estandarizado un servicio que proporciona un directorio en línea de servicios y productos, conocido como Directory Services (DS).

Es importante señalar que la OGC contempla el encadenamiento de estos servicios. Existen aspectos comunes a todos ellos. Así, las definiciones esenciales para un marco de servicio son:

- Un servicio es un conjunto de operaciones accesibles a través de una interfaz.
- Una interfaz es un conjunto de operaciones con denominación que caracteriza el comportamiento de una entidad.
- Una operación es la especificación de una transformación o consulta que se solicita a otro objeto. Cada operación tiene un nombre y una lista de parámetros.

En cuanto a las operaciones básicas que un servicio puede llevar a cabo se distinguen:

- Publicar. El proveedor de un servicio transmite metadatos que describen las capacidades del servicio y la dirección de red.
- Encontrar. Los solicitantes de los servicios utilizan esta operación para localizar tipos o casos de servicios específicos. Generalmente reciben los metadatos transmitidos por el proveedor de servicios.
- Vincular. Se utiliza cuando un solicitante y un proveedor negocian un servicio. El solicitante utiliza el metadatos del proveedor para vincularse con el servicio.

Capítulo 4

Estándar Web Processing Service. Análisis comparativo de los diferentes proyectos

El OpenGIS WPS se ha diseñado para estandarizar la forma de ofrecer cálculos GIS a través de Internet. Este estándar proporciona normas para describir cualquier cálculo, denominados aquí procesos, así como la forma de realizar las peticiones al proceso y de responder dicha petición. La norma también define cómo un cliente puede solicitar la ejecución de un proceso.

La especificación del Web Processing Service fue publicada por OGC en su primera versión en 2004 y en 2007 se liberó la versión 1.0.0. Este proyecto se basa en esta última versión, cuya especificación se encuentra en el propio portal del OGC (<http://www.opengeospatial.org/standards/wps>)

4.1 Web Processing Service 1.0.0

Un WPS proporciona acceso a operaciones o cálculos de datos geoespaciales, de distinta complejidad, mediante Servicios Web como interfaces. Puede contener operaciones para tratar tanto datos vectoriales como matriciales, provenientes de la red o del propio servidor.

Mediante el uso de servicios de geoprocésamiento distribuido, los usuarios pueden acceder y compartir rutinas de procesamiento remoto. El acceso a servicios de geoprocésamiento distribuidos incrementa la reutilización, disminuyendo el coste de mantenimiento y haciendo más eficiente el trabajo de los expertos en el contexto de las IDEs. La comunicación con estos servicios se realiza fácilmente mediante peticiones HTTP con XML, garantizando la interoperabilidad y la independencia entre sistemas. Si bien en un principio, WPS fue diseñado para trabajar con datos espacialmente referenciados, actualmente su especificación no limita el tipo de procesos únicamente a este tipo de datos, sino que es posible la ejecución de rutinas sobre datos vectoriales, raster, literales, u otro tipo de datos, entre ellos los que siguen el estándar GML.

La ventaja del procesamiento remoto para compartir rutinas y reutilizar métodos sin la necesidad de mantenimiento de aplicaciones de forma local, compite con la desventaja

de su rendimiento. En general, cuando trabajamos con datos espaciales, éstos suelen ser datos complejos y de gran tamaño, pudiendo hacer inviable el procesamiento remoto de este tipo de datos entre proveedores y consumidores de servicios. Por lo tanto, las técnicas de generalización complementan perfectamente a los servicios de geoprocésamiento ya que permiten la extracción de la información relevante que evite el tráfico de grandes volúmenes de datos.

El modelo propuesto por el OGC se basa en arquitecturas abiertas e interoperables, que permiten crear aplicaciones a partir de la concatenación de operaciones atómicas, tal y como muestra la figura 4, en contraste con las aplicaciones GIS monolíticas donde toda la funcionalidad está implementada dentro de la propia aplicación.

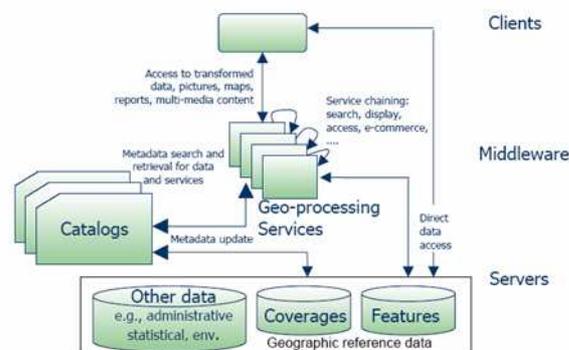


Fig. 4. Modelo de referencia de la arquitectura de las IDEs e INSPIRE

4.2 Aspectos generales de WPS

WPS, al igual que otros servicios Web OGC, es un interface genérico que no especifica ningún proceso específico. Cada implementación de WPS define los procesos que soporta, así como sus entradas y salidas asociadas. WPS puede verse como un modelo abstracto de un Servicio Web, para el cual deben desarrollarse perfiles.

Si bien, WPS permite desarrollar servicios para reutilizar significantes cantidades de código en el desarrollo de interfaces Web, facilitando al mismo tiempo la comprensión entre los desarrolladores de aplicaciones Web, el OGC propone el uso de perfiles estandarizados para conseguir una interoperabilidad completa. El empleo de un perfil permite la optimización del comportamiento de interoperabilidad de la interface del usuario cliente, así como del paradigma publicar/encontrar/vincular. Para alcanzar una alta interoperabilidad, cada proceso estará especificado en un perfil de aplicación de esa especificación.

Respecto a los mecanismos para encontrar y enlazar servicios, WPS siguen el modelo OGC de WMS y WFS, de manera que define una operación GetCapabilities que puede ser solicitada mediante el protocolo HTTP GET y HTTP POST y SOAP. WPS describe el interface del servicio que define como:

- Codificar la petición para la ejecución del proceso
- Codificar el response para la ejecución del proceso
- Embeber los datos y metadatos en las entradas y salidas de la ejecución del proceso.
- Referenciar los datos de entrada y salida accesibles vía Web.
- Devolver información del estado del proceso
- Devolver errores del proceso
- Obtener la respuesta de las salidas del proceso.

Para ello, WPS propone una serie de operaciones mediante las que se envían y se recibe la información y los datos de los procesos.

Los tipos de datos de entrada y salida que se utilizan pueden ser de tres tipos diferentes:

- LiteralData: Cadenas de caracteres, numérico entero y numérico doble.
- ComplexValue y ComplexValueRefence: Son raster, vectores y otros ficheros de datos, como mapas en varios formatos. En el caso de ComplexValueReference se trata de la URL donde se encuentran los datos.
- BoundingBox: Son pares de coordenadas

WPS permite para la recogida de datos de entrada dos métodos diferentes. Mediante la codificación de datos en la petición Execute, actuando como un servicio stand-alone, o haciendo referencia a los datos desde un recurso accesible vía Web, actuando como un servicio middleware que obtiene datos desde un recurso externo con el fin de ejecutar un proceso en la implementación local.

Además, la ejecución de los procesos puede realizarse de manera síncrona y asíncrona. Los cálculos geoespaciales pueden durar mucho tiempo, en termino de horas, días e incluso semanas. En estos casos, es posible realizar el seguimiento del progreso del proceso chequeando el estado del mismo.

Otro aspecto importante es el encadenamiento de servicios con WPS. Generalmente, un proceso WPS es una función atómica que realizar un cálculo geoespacial específico. El encadenamiento de procesos WPS facilita la creación de workflows repetitivos.

A partir de la versión 1.0.0 de WPS, el estándar es compatible con WSDL y SOAP. SOAP puede utilizarse para empaquetar las peticiones y las respuestas WPS y permite incluir certificado de seguridad al invocar los servicios WPS. En cuanto a WSDL, puede utilizarse para un proceso individual WPS, así como para una instancia WPS entera que puede incluir varios procesos.

4.2.1 Operaciones WPS

El servicio WPS proporciona tres operaciones: GetCapabilities, DescribeProcess y Execute. La implementación de estas operaciones es obligatoria por parte del servidor. Estas operaciones pueden utilizar HTTP GET con codificación KVP o HTTP POST con codificación XML o SOAP, dependiendo del tipo de codificación POST que vaya a soportar y que indicará el documento de capacidades.

El modelo UML de la figura 5 presenta el interface WPS. La clase WPS hereda la operación getCapabilities desde la interface OGCWebService, y añade las operaciones DescribeProcess y Execute.

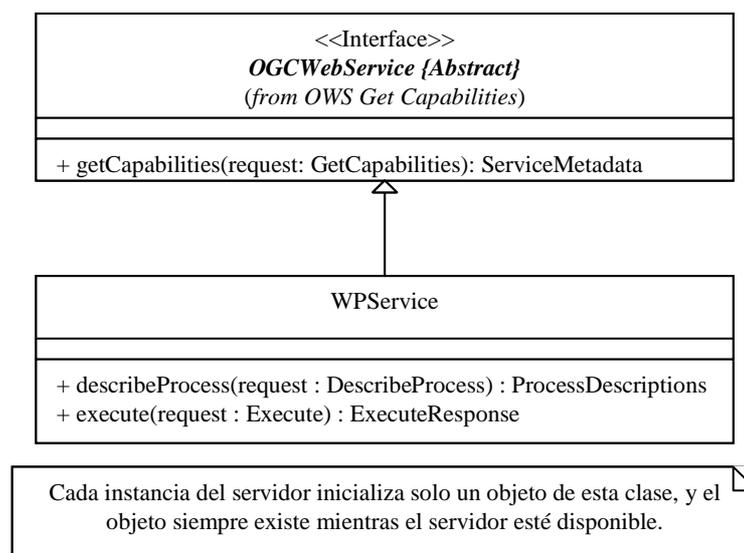


Fig. 5. Modelo UML del interface WPS

4.3.1.1. GetCapabilities

Esta operación permite obtener las características del servicio, así como la descripción general de cada una de los procesos ofrecidos por una instancia WPS. Esta respuesta se obtiene mediante un documento XML.

La petición debe ser HTTP GET, usando codificación KVP. Los servidores WPS no podrán implementar codificación KVP usando transferencia HTTP POST. Opcionalmente, los servidores WPS, pueden implementar transferencia HTTP POST en la petición de la operación GetCapabilities, utilizando sólo codificación XML

La siguiente tabla especifica la implementación de estos parámetros para clientes y servidores WPS de la petición de esta operación:

Componente	Definición	Multiplicidad y uso	Implementación cliente	Implementación servidor
Service	Identificador del tipo de servicio. Será siempre WPS	Uno (obligatorio)	Cada parámetro deberá ser implementado por todos los clientes, usando el valor especificado	Cada parámetro deberá ser implementado por todos los servidores, comprobando que cada parámetro se recibe con el valor específico.
Request	Nombre de la operación. Será siempre GetCapabilities	Uno (obligatorio)		
AcceptVersions		Cero o uno (opcional)	Podría implementarse por todos los clientes usando el valor especificado	Será implementada por todos los servidores, comprobando si se recibe con el valor especificado.
Language		Cero o uno (Opcional)	Podría ser implementarlo por todos los clientes	Podría ser implementado por servidores que ofrezcan capacidades multilingües.

El documento XML devuelto con la descripción de servicio debe contener:

1. Service: Identificación del servicio (Obligatorio).

2. Versión: Versión de la operación (Obligatorio).
3. Update Sequence: Versión del documento de metadatos del servicio, valor que se incrementa cuando se realizan cambios en el documento completo. Lang: Identificador de lenguaje. (Obligatorio)
4. Service Identification: Metadatos del servicio específico. Se trata de un esquema entre cuyos parámetros se incluye el tipo de servicio, la versión y el título. (Obligatorio).
5. Service Provider: Metadatos sobre la organización del servidor. (Obligatorio).
6. Operations Metadata: Metadatos de las operaciones especificadas por el servicio e implementadas por el servidor, incluyendo las URL para las peticiones de operaciones. Las operaciones de WPS son GetCapabilities, DescribeProcess y Execute. (Obligatorio).
7. Process Offerings: Descripción de cada uno de los procesos ofrecidos por el servicio. Para cada proceso se indicará su identificador y título, y opcionalmente podrá contener los parámetros “Abstract” con un resumen del proceso, “Metadata” con más metadatos sobre el proceso, “Profile” con el perfil que cumple, “WSDL” con la localización del documento WSDL, y “Process Versión” con la versión del proceso. (Obligatorio)
8. Languages: Contiene una lista de los lenguajes por defecto y opcionales que ofrece el servicio. (Obligatorio)
9. WSDL: La URL donde se puede recuperar el documento WSDL que describe el servicio completo.

La operación getCapabilities puede devolver una serie de excepciones ante errores de ejecución. Estas excepciones son:

Excepción	Significado	Valor
MissingParameterValue	La petición no incluye valor para un parámetro y el servidor no tiene declarado valor por defecto para dicho parámetro.	Nombre del parámetro ausente.
InvalidParameterValue	La petición contiene un valor de parámetro inválido.	Nombre del parámetro con valor inválido.

VersionNegotiationFailed	La lista de versiones en el parámetro “AcceptVersions” no incluye ninguna versión soportada por el servidor	Se omite.
NoApplicableCode	No hay código de excepción especificado por el servidor para la excepción.	Se omite.

4.2.1.1 DescribeProcess

Esta operación permite preguntar y recibir una descripción de uno o más procesos que pueden ser ejecutados por el servidor. Esta descripción incluye los parámetros de entrada y de salida del proceso y puede ser utilizada para construir automáticamente un interface que capture los valores de los parámetros.

Al igual que la operación GetCapabilities, la petición debe ser HTTP GET, usando codificación KVP y opcionalmente, puede implementarse transferencia HTTP POST utilizando sólo codificación XML. Los servidores WPS no podrán implementar codificación KVP usando transferencia HTTP POST.

Componente	Definición	Multiplicidad y uso
Service	Identificador del tipo de servicio. Será siempre WPS	Uno (obligatorio)
Request	Nombre de la operación. Será siempre DescribeProcess	Uno (obligatorio)
Version	Versión de la especificación de la operación	Uno (obligatorio)
Language	Identificador del lenguaje en la respuesta	Cero o uno (opcional)
Identifier	Identificación del proceso identificador, cuyo valor ha sido definido en la sección ProcessOffering del documento de capacidades	Uno o varios (obligatorio)

La respuesta contiene el identificador del servicio, la versión de la especificación, y el identificador de lenguaje, así como una estructura de datos con el formato y una descripción de cada proceso. Esta estructura de datos contiene:

1. Identifier: Identificador del proceso. Obligatorio
2. Title: Títulos del proceso. Obligatorio
3. Abstract: Descripción del proceso. Opcional.
4. Metadata: Incluye más metadatos sobre el proceso. Opcional.

5. Profile: El perfil que cumple el proceso. Opcional
6. processVersion: Versión del proceso. Obligatorio.
7. WSDL: Localización del documento WSDL que describe el proceso. Opcional
8. DataInputs: Lista de entradas obligatorias y opcionales del proceso. Opcional
9. ProcessOutputs: Lista de salidas requeridas y opcionales de la ejecución del proceso. Obligatorio.
10. storeSupported: Indica si la salida de datos compleja del proceso puede almacenarse por el servidor WPS como un recurso Web accesible. Opcional.
11. statusSupported: Indica si la respuesta de la operación Execute puede devolverse con la información del estado. Opcional.

La operación DescribeProcess puede devolver una serie de excepciones si se producen errores de ejecución. Estas excepciones son:

Excepción	Significado	Valor
MissingParameterValue	La petición no incluye valor para un parámetro y el servidor no tiene declarado valor por defecto para dicho parámetro.	Nombre del parámetro ausente.
InvalidParameterValue	La petición contiene un valor de parámetro inválido.	Nombre del parámetro con valor inválido.
NoApplicableCode	No hay código de excepción especificado por el servidor para la excepción.	Omitido.

4.2.1.2 Execute

La operación Execute permite ejecutar un proceso específico implementado por el servidor, usando los valores de los parámetros de entrada proporcionados y recibiendo los valores de salida.

Las entradas pueden incluirse directamente en la petición Execute, o pueden estar referenciadas en un recurso accesible vía Web. Las salidas pueden ser devueltas en formato de un documento XML, embebidas en el documento respuesta o almacenadas

en un recurso accesible vía Web. Si las salidas están almacenadas, la respuesta de la operación Execute siempre consistirá en documento XML que incluya una URL por cada salida almacenada. En el caso de que la respuesta Execute sea una salida única, no es necesario incluir un documento XML de respuesta, puede devolverse dentro de la propia salida.

Generalmente, el documento de respuesta se devuelve una vez haya terminado completamente la ejecución, sin embargo, un cliente puede pedir al servidor que le devuelva el documento una vez haya aceptado la petición de Execute y continuar trabajando de manera asíncrona. En este caso, la respuesta incluye una URL donde se dejará el documento respuesta durante y al terminar la ejecución. El servidor irá realizando actualizaciones de este documento a lo largo de la ejecución. Esto permite al cliente determinar el estado del proceso.

La petición de una operación Execute debe incluir los siguientes parámetros:

Componente	Definición	Multiplidad y uso
Service	Identificador del tipo de servicio. Será siempre WPS	Uno (obligatorio)
Request	Nombre de la operación. Será siempre Execute	Uno (obligatorio)
Version	Versión de la especificación de la operación	Uno (obligatorio)
Language	Identificador del lenguaje en la respuesta	Cero o uno (opcional)
Identifier	Identificación del proceso identificador, cuyo valor ha sido definido en la sección ProcessOffering del documento de capacidades	Uno o varios (obligatorio)
DataInputs	Lista de entradas proporcionadas para la ejecución de este proceso. Se trata de una estructura de datos tipo DataInput que contiene una o varias entradas y para cada una de ellas, al menos, un identificador, un título y un resumen.	Cero o uno (opcional)
Response Form	Define el tipo de respuesta del WPS, bien será una salida raw data o un documento XML. Si se omite, la respuesta será siempre un documento respuesta que incluya todas las salidas embebidas en la respuesta.	Cero o una (opcional)

Cuando la petición de la operación contiene múltiples entradas es habitual proporcionar dichas entradas mediante una o más URI (generalmente URLs), salvo si se trata de entradas de valores escalares simples. Esto no significa que se pueden ejecutar peticiones batch, cada vez que debe ejecutarse un proceso debe realizarse una petición

de la operación Execute, independientemente del número de veces que se ejecute dicho proceso.

La lista de entradas y salidas del proceso no está ordenada, lo que significa que cuando existe un número elevado de entradas o salidas en un proceso, puede saturar al servidor que intenta interpretar los datos recibidos y realizar su transformación. En estos casos, se recomienda empaquetar y ordenar las entradas/salidas de manera eficiente, e identificar estos requerimientos de ordenación óptima en la descripción del proceso.

La ejecución de una petición de la operación Execute puede realizarse implementando la transferencia mediante HTTP GET usando codificación KVP. En este caso la codificación de los campos de las estructuras de datos DataInput, ResponseDocument y RawDataOutput deben seguir las siguientes reglas:

- Las entradas se separan entre sí mediante punto y coma (;).
- El nombre de cada entrada se separa mediante el signo igual (=) del valor y sus atributos, y de igual forma, el nombre de un atributo de sus valores.
- El valor de una entrada se separa de sus atributos mediante el signo @, así como un atributo de otro.
- Los nombres de los campos y de los atributos son sensibles a las mayúsculas. Un nombre de campo o de atributo incorrecto provoca una excepción InvalidParameterException.
- La ausencia de un campo obligatorio provoca una excepción MissingParameterValue.
- Todos los valores de campos y atributos se codificarán usando el estándar de Internet para codificar URL.
- Las referencias usando HTTP POST no se soportan para la codificación KVP.
- Las peticiones HTTP usando codificación KVP, soportan el encadenamiento de peticiones.

Como en las anteriores operaciones, todos los servidores WPS deben implementar la transferencia de la petición de una operación Execute mediante HTTP POST usando exclusivamente codificación XML.

La forma de respuesta a una operación Execute depende del valor del parámetro “ResponseForm” de la petición de la operación.

Si el valor del parámetro “ResponseForm” es “RawDataOutput”, una vez terminada la ejecución del proceso y si se produce solamente una salida compleja, la respuesta consiste en dicha salida compleja en su forma original devuelta directamente al cliente. Este es el caso, por ejemplo, de un proceso WPS que devuelve una imagen GIF como salida, la imagen GIF se devuelve al cliente como respuesta a su petición.

Si el valor del parámetro “ResponseForm” es “ExecuteResponse”, el documento obtenido tiene la siguiente estructura:

1. Service: Identificador del servicio. Contiene siempre WPS. Obligatorio.
2. Version: Versión de la especificación de la operación. Obligatorio.
3. Lang: Identificación del lenguaje. Obligatorio.
4. Status Location: Referencia la localización del documento si en la petición de la operación, el parámetro ExecuteResponse incluye storeExecuteResponse=TRUE. Sólo es obligatorio en este caso.
5. Service Instance: URL GetCapabilities del servicio que ha sido invocado. Obligatorio.
6. Process: Descripción del proceso. Obligatorio.
7. Status: Estado de la ejecución del proceso. Indica la hora de creación del documento o la hora de finalización del proceso, así como si el proceso ha sido aceptado, si ha comenzado, si está parado, si ha terminado y se ha fallado. Obligatorio si en la petición de la operación, el parámetro ExecuteResponse incluye status=TRUE.
8. DataInputs: Lista de entradas que se han indicado en la ejecución del proceso. Se incluye solo si en la petición de la operación, el parámetro ExecuteResponse incluye lineage=TRUE.
9. Output Definitions: Lista de definición de salidas solicitadas. Se incluye solo si en la petición de la operación, el parámetro ExecuteResponse incluye lineage=TRUE.
10. Process Outputs : Lista de valores de salida desde la ejecución del proceso.

La operación Execute puede devolver una serie de excepciones si se producen errores de ejecución. Estas excepciones son:

Excepción	Significado	Valor
MissingParameterValue	La petición no incluye valor para un parámetro y el servidor no tiene declarado valor por defecto para dicho parámetro.	Nombre del parámetro ausente.
InvalidParameterValue	La petición contiene un valor de parámetro inválido.	Nombre del parámetro con valor inválido.
NoApplicableCode	No hay código de excepción especificado por el servidor para la excepción.	Nulo.
NotEnoughStorage	El servidor no tiene espacio libre suficiente para almacenar las entradas y salidas asociadas a la petición	Nulo.
ServerBusy	El servidor está ocupado para aceptar y encolar la petición en este momento.	Nulo.
FileSizeExceeded	El tamaño de la fila de uno de los parámetros de entrada es demasiado largo.	Identificador del parámetro que excede el tamaño máximo.
StorageNotSupported	La operación incluye “storage= true” pero el servidor no ofrece almacenamiento	Nulo.
VersiónNegotiationFailed	La versión de una entrada no está soportada por el servidor y la negociación de versiones falla.	Identifica la entrada que no puede ser accedida.

4.2.2 Implementaciones WPS

Existen varios desarrollos sobre la especificación OCG WPS, la mayoría de ellos están implementan la versión 0.4.0 del estándar.

El estándar WPS 1.0.0 contempla que cada especificación de implementación OWS incluya un juego de pruebas abstractas antes de ser sometida a la ISO/TC 211. Este juego de pruebas abstractas especifica, a un alto nivel, como las implementaciones de servidor y clientes de dicha especificación pueden probar su conformidad con la especificación WPS. El marco para dicho juego de pruebas se especifica en la ISO 19.105 sobre la conformidad y pruebas en la Información Geográfica.

Un juego de pruebas abstractas contiene múltiples pruebas abstractas, agrupados en uno o más módulos de prueba. Así, pueden diferenciarse módulos de pruebas de alto nivel sobre la conformidad en la implementación cliente y módulos de pruebas de alto nivel sobre conformidad en la implementación del servidor. Estos módulos pueden contener pruebas a más bajo nivel. Hasta la fecha, sólo se han definido módulos de pruebas a bajo nivel para el módulo de pruebas del servidor. Estos módulos son:

- Módulo de prueba de todas las operaciones implementadas. Se trata de pruebas abstractas para chequear las propiedades del servidor que son comunes en todas las operaciones implementadas.
- Módulo de prueba de la operación GetCapabilities. Se trata de pruebas para chequear propiedades que se han especificado en la operación GetCapabilities.
- Respuesta de otras operaciones. Se trata de pruebas para chequear las propiedades que aplican a todas las operaciones excepto GetCapabilities.

En los módulos de pruebas del cliente como del servidor, todas las operaciones especificadas e implementadas deben ser testeadas, incluyendo la transferencia HTTP GET y HTTP POST de cada petición de operación. En los módulos de prueba de las especificaciones, todas las operaciones especificadas deben ser probadas, incluyendo la transferencia GET y POST de las peticiones de las operaciones, al igual que todas las peticiones y respuesta de las operaciones, los parámetros especificados. Cualquier elemento no implementado en la especificación no debe ser chequeado.

A continuación se presentan alguno de los proyectos más representativos de servidores WPS.

4.2.2.1 Deegree Framework.

Es un Framework Java de código abierto, bajo licencia GNU LGPL, que implementa la mayoría de estándares de OGC, entre ellos WMS, WCS, WFS, CSW y WAS, así como sus clientes y componentes de seguridad.

Es muy utilizado en la administración alemana.

El interface por http se realiza por un Servlet registrado dentro de un motor como Tomcat o Jetty. El Servlet selecciona una clase encargada dependiendo de la petición recibida, delegando en ella la responsabilidad del servicio. Dependiendo del proceso requerido, el WPSservice decide que proceso es responsable de él.

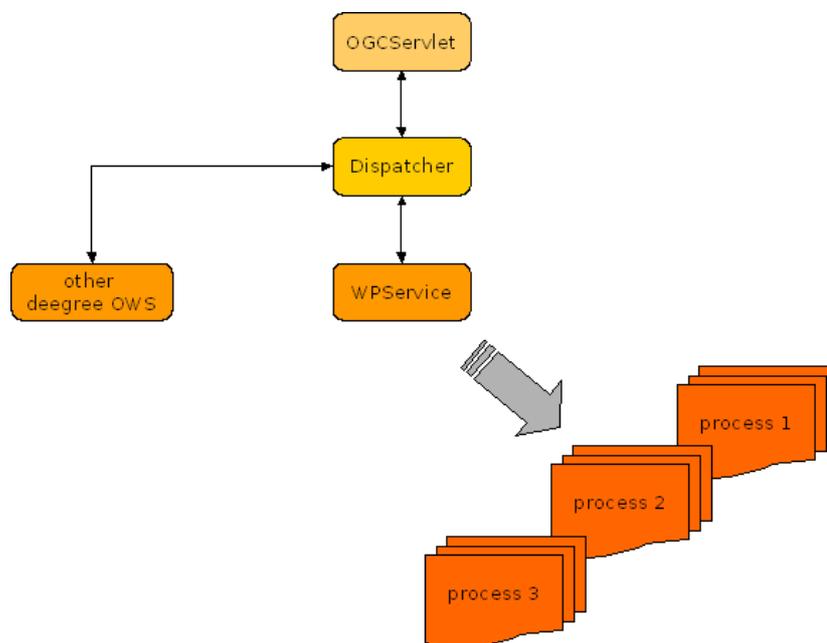


Fig. 6. Arquitectura de Deegree WPS

Deegree WPS está basado en la versión 0.4.0 del estándar, sin embargo no está publicada como una versión estable.

4.2.2.2 WPSint

Se trata de un Framework de código abierto compuesto por un plugin de Java que trata peticiones de WPS así como sus respuestas y por un cliente genérico WPS que puede acceder a cualquier WPS.

Este plugin simplifica el trabajo de creación de operaciones WPS y es accesible desde un navegador Web, una aplicación Web o un cliente de escritorio. A pesar de implementar la versión 0.4.0 del estándar, puede utilizarse mediante el interface

WSDL/SOAP. Actualmente, se está trabajando en una nueva versión para cumplir la versión 1.0.0 del estándar.

Se trata de una implementación en Java para Spring (Framework de aplicaciones Java/J2EE). Con la descarga del software se incluye los ficheros para desplegar el servidor y el cliente genérico, el código fuente, un plugin para eclipse y la documentación de API.

El plug-in está provisto de seguridad en la autenticación, autorización y control de acceso, mediante el sistema de seguridad ACEGI.

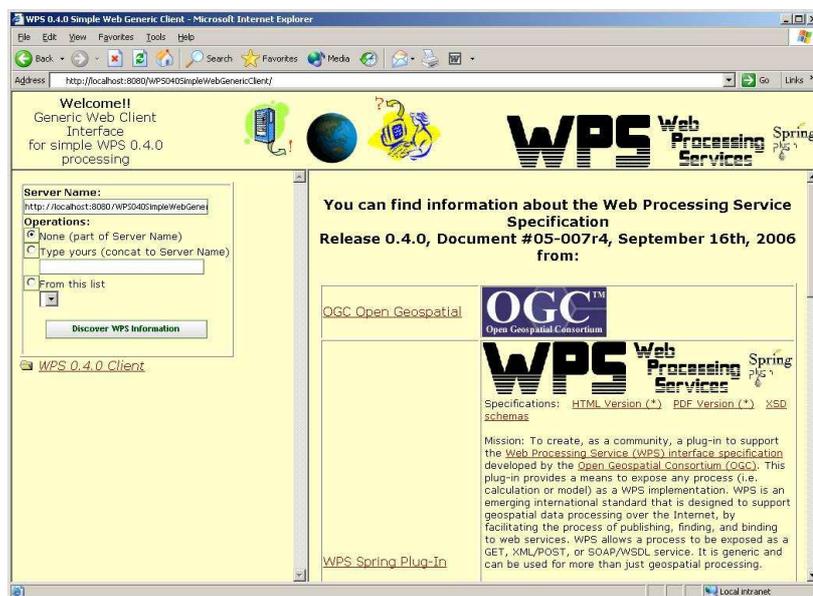


Fig. 7. Aspecto visual del cliente WPSint

4.2.2.3 pyWPS (Python Web Processing Server)

Es un proyecto desarrollado desde 2006 como proyecto soportado por BDU (Fundación Federal de Medio Ambiente Alemana), que implementa el estándar en su versión 0.4.0. Está escrito en el lenguaje de programación Python, con apoyo de GRASS (Geographic Resources Analysis System), Los procesos pueden estar escritos usando GRASS, pero también se puede utilizar con otros programas GIS, como el paquete R, GDAL o PROJ.

PyWPS es una aplicación de traducción entre el cliente (Navegadores Web, GIS de escritorio, utilidades de línea de comandos,..) y las utilidades de trabajo del servidor. PyWPS además de trasladar peticiones del estándar, monitoriza el progreso del cálculo e informa al usuario, y cuando el cálculo ha terminado, retorna el resultado. Sin embargo, no realiza ningún trabajo por si mismo, requiere de otras herramientas de trabajo.

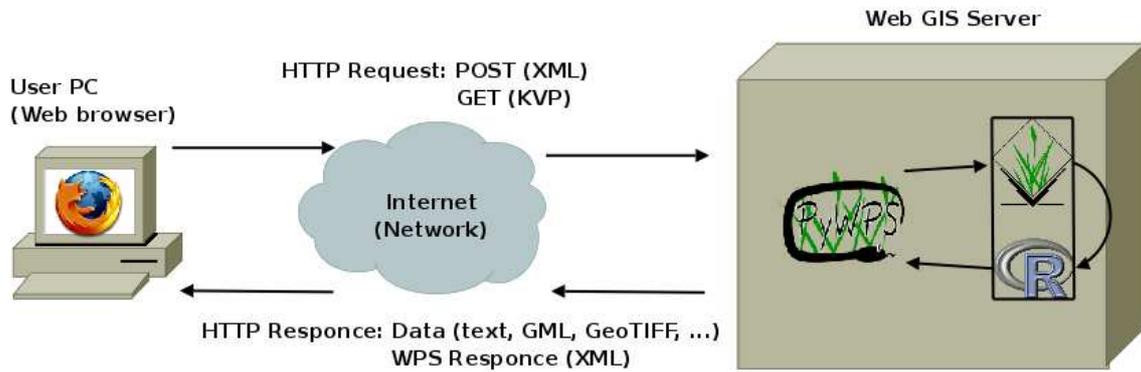


Fig. 8. Modo de trabajo de PyWPS con GRASS GIS

Actualmente, se está trabajando en la versión 3.0 con el propósito de soportar WPS 1.0.0 o una posible implementación de la siguiente versión del documento del estándar.

Varias aplicaciones cliente puede usar PyWPS, entre ellas el proyecto Embrio, y hay un plugin para OpenLayers.

4.2.2.4 52 North WPS

Se trata de una implementación de WPS 1.0.0 realizada por 52North dentro de un amplio proyecto de software libre, el cuál incorpora además de este componente específico para WPS, otras especificaciones del OGC. Está escrito en Java, como un plug-in para un contenedor Java Tomcat.

Además, proporciona los clientes uDig, mostrado en la figura 9, Open-Jump y lib.

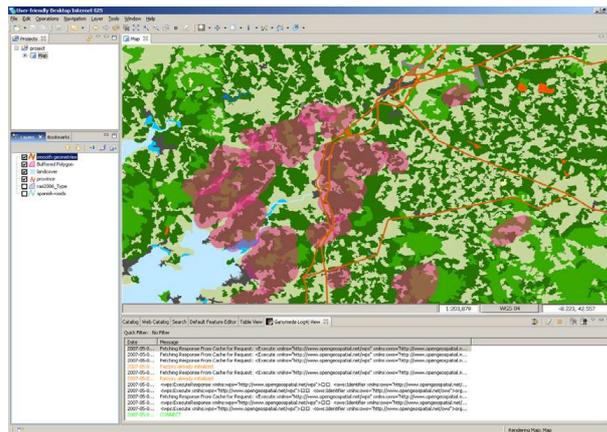


Fig. 9. Procesamiento basado en varios servicios OGC desde un cliente uDIG.

4.2.2.5 GeoConnection WPS

Es implementación de una de las versiones más antiguas del estándar, 0.2.0. El proyecto parece que dejará de estar soportado en breve.

4.2.3 El servidor 52 North WPS

El servidor 52n WPS permite el desarrollo de geoprocetos en la Web. Las principales características de este servidor son:

- Su implementación está basada completamente en código abierto de Java.
- Soporta todas las características de la especificación WPS versión 1.0.0.
- Creado sobre librerías robustas (Java Topology Suite, Geotools, xmlBeans, Servlet API, derby).
- Soporta todas las operaciones obligatorias de WPS 1.0.0 (más SOAP/WSDL beta).
- Soporta excepciones de manejo acorde a las especificaciones.
- Implementa un cliente básico para acceder al WPS (incluyendo la codificación XML completa).
- Plug-in para uDig y para JUMP.
- Almacena los resultados de ejecución.
- Soporta completamente GLM2 para ComplexData.
- Procesamiento asíncrono.
- Soporta raw data.
- Todas las respuestas se almacenan en ficheros del sistema pero pueden almacenarse en cualquier base de datos como Apache Derby.
- Soporta Maven completamente.
- Soporta HTTP-GET para la operación Execute.

4.2.4 Instalación del servidor.

Para el desarrollo del proyecto, se ha instalado un servidor WPS en la misma máquina local en la que posteriormente se realizará el desarrollo del cliente.

La instalación del servidor se ha llevado a cabo siguiendo las instrucciones publicadas por el propio 52north (<http://52north.org/>). Los requerimientos iniciales del sistema son Tomcat 5.5 y J2SE 5 o superior.

También es posible la instalación del código fuente tanto del servidor como del cliente. El proyecto utiliza Subversion para el control y la distribución de copias de dicho código. El repositorio de este código se encuentra en el propio portal 52north (<https://52north.org/svn/geoprocessing>).

Se puede utilizar eclipse para la instalación de estos fuentes, y requiere los plugins Subclipse, para la conexión con el repositorio Subversion, y m2eclipse si se desea compilar y poner en marcha el proyecto mediante Maven, para facilitar la búsqueda de las librerías externas que utiliza el programa y el empaquetado del mismo. Las librerías necesarias para la compilación del proyecto son la que aparecen en la tabla siguiente:

Grupo	Proyecto	Versión	Tipo
javax.xml.bind	jsr173_api	1.0	jar
log4j	log4j	1.2.14	jar
org.n52.wps	52n-wps-commons	1.0-rc3-snapshot	jar
org.n52.wps	<u>52n-wps-install-wizard</u>	1.0-rc3-snapshot	jar
org.n52.wps	<u>52n-wps-io</u>	1.0-rc3-snapshot	jar
org.n52.wps	<u>52n-wps-server</u>	1.0-rc3-snapshot	jar
xmlbeans	xbean	2.2.0	jar

En cuanto a la ejecución, se requiere de la librería:

Grupo	Proyecto	Versión	Tipo
net.sourceforge.pjl-comp-filter	pjl-comp-filter	1.6.4	jar

Una vez compilado y desplegado el servidor en el servidor de aplicaciones Tomcat, se debe realizar un testeado para comprobar su correcto funcionamiento.

Para ello, tal y como establece el OGC, en el repositorio SVN geoprocessing de 52n, está disponible el proyecto WPS-tester con los siguientes test:

- testGetCapabilities
- testDescribeProcess
- testExecute
- testExecuteStore
- testExecuteReference
- testExecuteStatus
- testExecuteRaw

El servidor describe sus capacidades en un documento XML que se obtiene mediante una petición del siguiente tipo:

http://localhost:8080/wps

Este servidor responde a las tres operaciones básicas de un WPS (getCapabilities, DescribeProcess y Execute), y atiende a sus peticiones mediante HTTP Get o mediante HTTP POST. Esta información puede verse en el campo OperationsMetadata del documento XML:

```
<ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8080/wps/WebProcessingService"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeProcess">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8080/wps/WebProcessingService"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Execute">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://localhost:8080/wps/WebProcessingService"/>
        <ows:Post xlink:href="http://localhost:8080/wps/WebProcessingService"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationsMetadata>
```

En cuanto a los procesos que implementa el servidor, se muestran en la entrada ProcessOffering.

```
<wps:ProcessOfferings>
  <wps:Process wps:processVersion="2">
    <ows:Identifier>org.n52.wps.server.algorithm.SimpleBufferAlgorithm</ows:Identifier>
    <ows:Title>Create a buffer around a polygon.</ows:Title>
  </wps:Process>
  <wps:Process wps:processVersion="2">
    <ows:Identifier>org.n52.wps.server.algorithm.simplify.TopologyPreservingSimplificationAlgorithm</ows:Identifier>
    <ows:Title>Topology preserving simplification algorithm</ows:Title>
  </wps:Process>
  <wps:Process wps:processVersion="2">
    <ows:Identifier>org.n52.wps.server.algorithm.simplify.DouglasPeuckerAlgorithm</ows:Identifier>
    <ows:Title>douglasPeucker algorithm</ows:Title>
  </wps:Process>
</wps:ProcessOfferings>
```

Mediante un cliente ligero se podrían realizarse peticiones para cualquiera de estos procesos. Sin embargo, el propósito del presente proyecto es realizar dichas peticiones desde un cliente pesado, en concreto desde gvSIG.

Capítulo 5

El software gvSIG. Instalación y configuración.

Principales características

La Generalitat Valenciana con la participación de la Universidad Jaume I y la cofinanciación de la Unión Europea, inicia en el año 2003 un proyecto para el desarrollo y la implementación de un nuevo programa para el manejo de información geográfica, con las directrices de que la solución fuera portable, modular, de código abierto, interoperable y sujeta a los estándares.

De esta iniciativa surge gvSIG, un Sistema de Información Geográfica multiplataforma, desarrollado como software libre, y que sigue los estándares del OGC, garantizando la interoperabilidad necesaria para ser utilizada como cliente de las Infraestructuras de Datos Espaciales (IDEs).

5.1 Introducción a gvSIG

La aplicación está desarrollada en Java, utiliza librerías estándar de GIS conocidas, como son Geotools o Java Topology Suite y contempla la posibilidad de crear extensiones en Java utilizando sus propias clases.

El hecho de diseñar el proyecto sobre una plataforma abierta, permite presentarlo como un framework base, al que se pueden añadir plugins que doten de nuevas funcionalidades al propio proyecto. Su arquitectura se muestra en la figura 10.

La plataforma gvSIG está en si misma construida a modo de capas, cada una de las cuales define sus puntos de extensión. A su vez, cada plugin puede definir sus propios puntos de extensión. Este modelo de plugin, está permitiendo añadir gran variedad de funcionalidades a la plataforma base de gvSIG, de forma que los artefactos de cada herramienta, como pueden ser los distintos tipos de capas, o botones, se presentan al usuario a través de la interfaz única del programa.

El núcleo de gvSIG está formado por tres subsistemas:

- Subsistema gvSIG: Representa la parte visual de la aplicación. Es la aplicación gráfica de los datos geográficos gestionando el interface de usuario.

El subsistema gvSIG se compone de:

- Gestor gráfico Andami: Proporciona los métodos necesarios para que los plugins puedan comunicarse tanto con la aplicación principal como entre ellos y soporta los métodos necesarios para gestionar el interface gráfico. Es el Framework de la aplicación, todo lo demás son extensiones.
 - Project: Es el contenedor de la estructura de documentos que hay cargados en un momento dado. Contiene también los datos básicos del proyecto.
 - Documents. Son los distintos tipos de documentos que soporta la aplicación, Vistas, Tablas y Mapas. Este componente proporciona también los puntos de extensión necesarios para incluir nuevos tipo de documentos.
 - View: Representación gráfica de la cartografía.
 - Layout: Representación gráfica de una vista en un soporte apto para imprimir.
 - Table: Es la representación gráfica de los datos alfanuméricos.
 - Layers: Son el conjunto de capas que pueden insertarse en una vista de gvSIG
- Subsistema Fmap. Es el motor de la aplicación. Incluye todas las clases necesarias para manejar objetos SIG, tanto para dibujar la cartografía como para acceder a los datos. Se compone de un gestor de herramientas, capas y orígenes de datos para las capas y las geometrías que se utilizan para representar los datos en las capas.

El subsistema Fmap se compone de:

- MapControl: Se encarga de dibujar y mantener la herramienta actual, conoce todas las herramientas que existen en la aplicación.
- MapContex: Es el contexto de la parte gráfica. Contiene los elementos necesarios para que el MapControl pueda realizar su labor.
- Behaivor: Es un comportamiento de una herramienta. También dice como se comporta la herramienta gráficamente. Controla el dibujado de la herramienta y el iniciador de los eventos de la herramienta.

- Listeners: Son los encargados de gestionar los eventos de las distintas herramientas ya sea propagándolos hacia quien corresponda o ejecutando las instrucciones necesarias.
- Layer: Contiene las características de la capa y las herramientas necesarias para su gestión.
- Geometrías: Son los distintos tipos de elementos gráficos que pueden ser representados dentro de una layer.
- DataSource y drivers: Contiene los métodos necesarios para la gestión de los datos tanto gráficos como alfanuméricos.
- Subsistema Subdriver: Es el subsistema que sirve de puente entre la aplicación y los datos. Contiene las clases necesarias para acceder a los datos, escribir datos en una fuente, así como las propiedades de acceso a fuentes remotas.

Los componentes de este subsistema son:

- RemoteService: Contiene las herramientas necesarias para unificar el acceso a datos remotos.
- Drivers: Gestiona los distintos tipos de datos soportados por gvSIG.
- DriverManager: Proporciona la carga y el acceso a los drivers disponibles en la aplicación, tanto alfanuméricos como espaciales.
- WriterManager: Proporciona la carga y el acceso a los writers disponibles en la aplicación, tanto alfanuméricos como espaciales.
- Writers: Permite las operaciones de escritura sobre los distintos tipos de formatos soportados.
- VectorialSources: Proporciona acceso a los datos con las geometrías.
- DataSources: Proporciona acceso a los datos alfanuméricos.
- RasterSouces: Proporciona acceso a los datos de tipo raster.

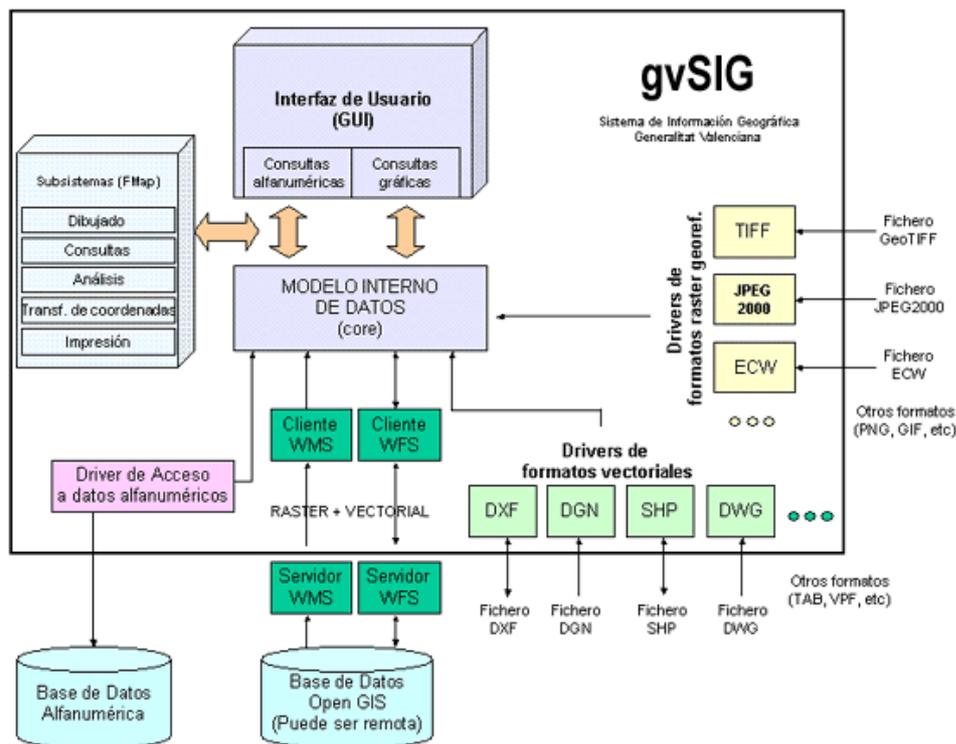


Fig. 10. Arquitectura gvSIG

5.2 Requisitos de Instalación.

gvSIG se distribuye bajo licencia GPL. Actualmente, hay disponibles distribuciones para los sistemas operativos Microsoft Windows, Linux y Mac OS X.

Los requisitos previos a la instalación de gvSIG son:

- JDK 1.5
- JAI 1.1.3
- JAIImage I/O 1.1

La descarga, tanto de la aplicación como del código fuente, puede realizarse desde el propio portal de gvSIG (<http://www.gvsig.gva.es/>).

Durante la instalación del software se puede indicar la instalación de algunas de las extensiones de gvSIG. Sin embargo, hay otras extensiones que deben ser instaladas manualmente. Para ello, simplemente es necesario descargar el fichero de la extensión correspondiente y ejecutar dicho fichero.

Por último, es necesario realizar la descarga de los ficheros fuentes de gvSIG para poder implantar el cliente WPS, objetivo del presente proyecto. Se utilizará Eclipse como IDE para el desarrollo del proyecto.

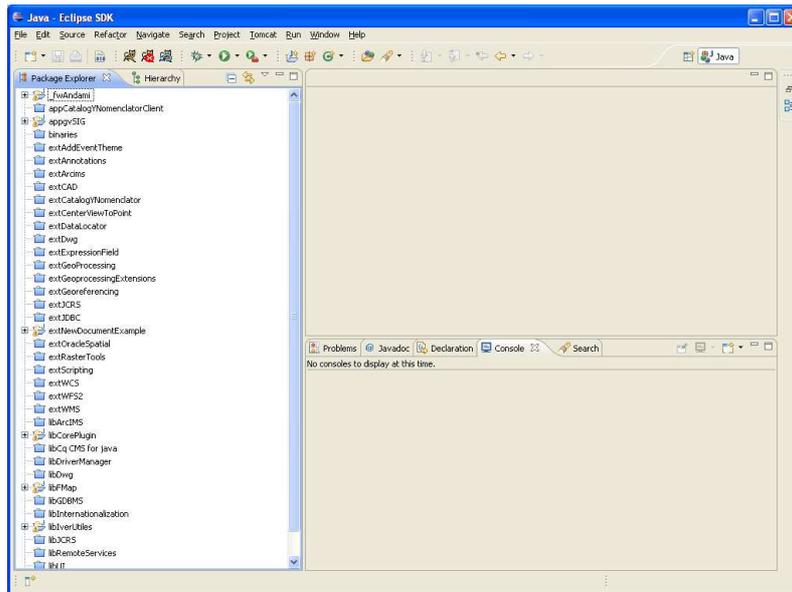


Fig. 11. Vista del proyecto gvSIG en eclipse.

5.3 Nociones básicas sobre las funcionalidades de gvSIG.

El programa gvSIG está orientado al manejo de información geográfica por usuarios finales. Se caracteriza por una interfaz amigable y sencilla, tal y como se muestra en la figura 12, con capacidad para acceder ágilmente a los formatos más usuales, raster y vectoriales, así como de integrar datos en una vista, tanto locales como remotos a través de un origen WMS, WFS, WCS o JDBC (Java Database Connectivity). Además, permite realizar análisis complejos sobre ellos.

La interfaz de gvSIG proporciona los elementos necesarios para comunicarse con el programa.

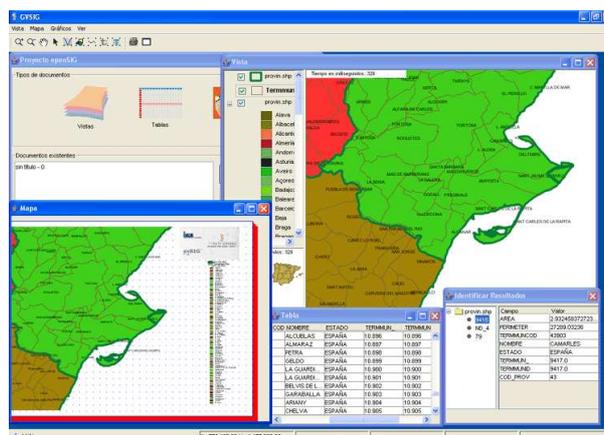


Fig. 12. Interfaz gvSIG

En gvSIG toda la actividad se localiza en un proyecto, el cual está formado por diferentes documentos, que pueden ser de tipo vistas, tablas y/o mapas. Los proyectos

son archivos de extensión .gvp. Estos archivos no contienen los datos espaciales y atributos asociados en forma de tablas, sino referencias al lugar donde se conservan las fuentes de dichos datos.

El programa permite configurar al usuario diversas características del mismo desde la ventana de preferencias. Entre dichas preferencias se encuentran las relativas a las extensiones que se utilizarán durante la ejecución del programa, y que establecen tanto la ubicación de las mismas como aquellas que desea utilizar. Una extensión es una ampliación del proyecto que incorpora una nueva funcionalidad.

5.3.1 Vistas

Las vistas son los documentos de gvSIG que constituyen el área de trabajo de la información cartográfica. Dentro de una vista pueden existir distintas capas de información. La relación de las capas que contiene una vista se muestra en la tabla de contenidos (ToC) en la parte izquierda de la pantalla. En dicha tabla, se indica para cada capa, si es o no visible y el orden de aparición en la Vista.

Las propiedades de una vista son su nombre, las unidades de cartografía y las unidades de distancia, el color de fondo, la proyección actual y el sistema de referencia.

Las capas pueden contener ficheros de varios formatos:

- **Datos SIG:** El formato estándar (de facto) de los SIG es el shape, que almacena tanto datos espaciales como atributos de los mismos. Se trata en realidad de tres o más archivos, con el mismo nombre y extensiones diferentes (dbf para el archivo que contiene la tabla de atributos, shp para el fichero de datos espaciales y shx para el archivo de índice de los datos espaciales).
- **Datos CAD:** Son archivos de dibujo vectorial. Los formatos soportados son dxf y dgn.
- **Datos WMS:** Son datos que se encuentran en otros servidores, accesibles vía Web y que cumplen el estándar WMS del OGC.
- **Datos WFS:** Son datos que se encuentran en otros servidores, accesibles vía Web y que cumplen el estándar WFS del OGC.
- **GML:** Son ficheros XML que pueden ser visualizados o exportados siguiendo las especificaciones del OGC.

- Imágenes: Archivos de tipo raster con formato tiff, jpg, ecw, musid, gif o bmp, entre otros. Estos ficheros pueden ser visualizado y modificados en el propio programa.

5.3.2 Tablas

Las tablas son los documentos que contienen la información alfanumérica. Las tablas se componen de filas o registro (que representan cada uno de los elementos de las bases de datos) y columnas o campos (que definen los distintos atributos de cada elemento).

Todas las capas de información vectorial tienen asociada una tabla de atributos. Cada elemento gráfico de una determinada capa tiene su correspondiente registro en dicha tabla de atributos. Se puede seleccionar elementos en una tabla y quedan seleccionados en la capa. El tratamiento de estas tablas es similar al que se puede realizar con cualquier software de operación de base de datos. Se pueden realizar estadísticas, filtros, ordenaciones, uniones, cargas de ficheros externos e incluso dispone de una calculadora de campos que permite realizar distintos tipos de cálculos sobre los campos de una tabla.

5.3.3 Mapas

Los documentos de mapa permiten diseñar y combinar en una página todos los elementos cartográficos que deseamos que aparezcan en un mapa impreso. En concreto, podemos añadir:

- Vistas
- Imágenes
- Barras de escala
- Textos
- Leyendas
- Objetos gráficos (Puntos, rectángulos, círculos, líneas, polilíneas o polígonos)
- Norte
- Texto
- Cajetines
- Escala

5.4 gvSIG y OGC

Teniendo en cuenta que entre los objetivos principales de gvSIG están el cumplimiento de directrices marcadas por el OGC y la extensibilidad del proyecto, se han incorporado al mismo algunos módulos que lo dotan de capacidad tanto para obtener información de un servidor externo que cumpla los estándares de WMS, WCS y WFS, como de publicar información en un servidor mediante este tipo de servicios.

5.4.1 Extensiones para añadir información desde servidores externos.

Son varias las extensiones disponibles en gvSIG que permiten incorporar capas mediante los protocolos establecidos por el OGC.

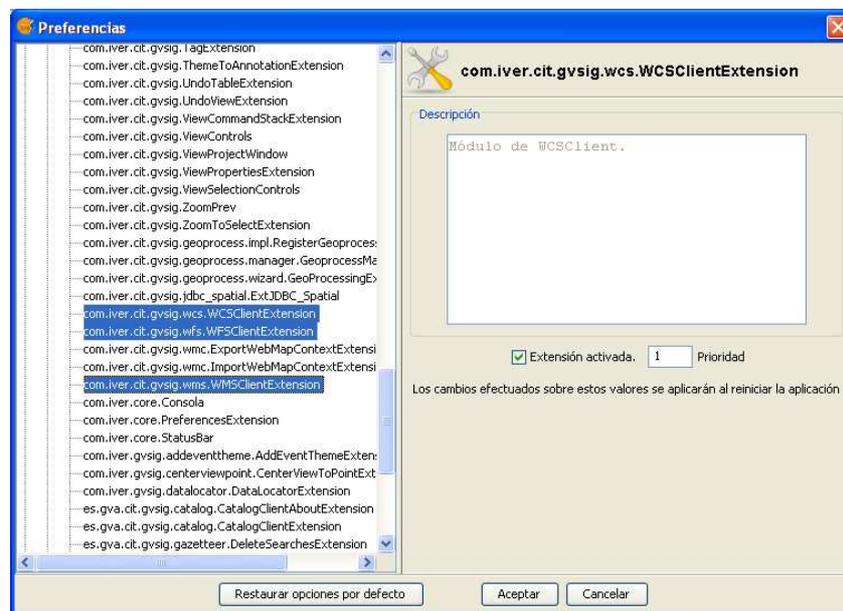


Fig. 13. Árbol de extensiones de la ventana de preferencias de gvSIG

Se puede añadir una capa a través del protocolo WFS. La figura 14 muestra como desde la ventana “Añadir capas” y seleccionando la pestaña WFS, puede conectarse con cualquier servidor que soporte este protocolo. gvSIG recupera de dicho servidor una capa vectorial en formato GML y se encarga de obtener las geometrías y los atributos asociados a cada “Feature” interpretando el contenido del archivo.

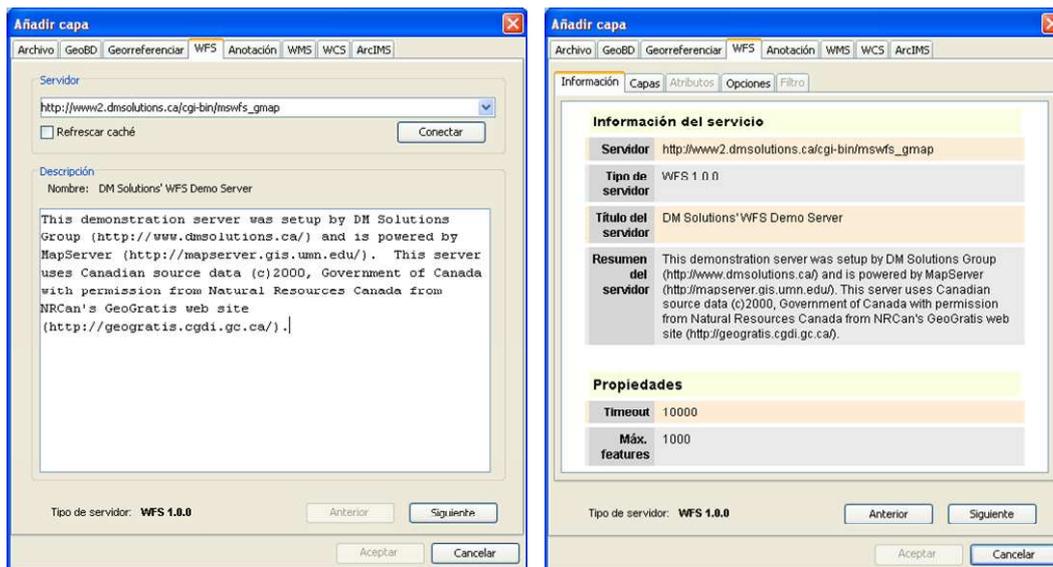


Fig. 14. Conexión a un servidor WFS desde gvSIG

Una vez establecida la conexión, es posible seleccionar la capa que se quiere cargar, los atributos de la capa, establecer filtros sobre dichos atributos, y configurar el número máximo de entidades que se pueden descargar, así como el tiempo de espera antes de descartar la conexión.

También es posible añadir una capa a través del protocolo WMS, como muestra la figura 15. El procedimiento para establecer la conexión con el servidor remoto es similar al anterior. En este caso, una vez seleccionadas las capas que se desean añadir, puede elegirse la forma de visualización de dichas capas, la dimensión, el formato, el sistema de coordenadas y/o la transparencia.

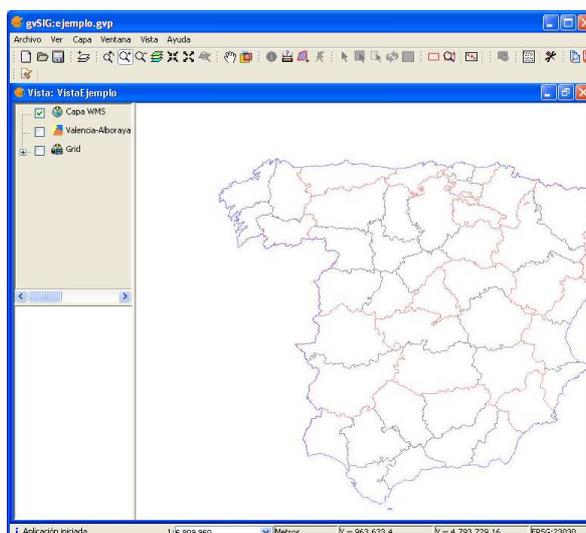


Fig. 15. Capa WMS obtenida del servidor de la D.G. del Catastro.

Así mismo, puede añadirse una capa a través del protocolo WCS. De nuevo, el proceso de conexión al servidor es similar a los anteriormente descritos. En este caso, se seleccionará la cobertura deseada, así como el formato de imagen con que se realizará la petición y también el sistema de referencia. Algunos servidores proporcionan la posibilidad de configurar las variables de Tiempo y Parámetros.

5.4.2 Extensión de publicación.

Esta extensión permite publicar la información tratada con gvSIG en una vista en un servidor que implemente servicios OGC. Concretamente, permite generar las configuraciones de los siguientes servicios:

- Mapserver 5.x WMS con fuentes de datos raster, shapefiles y PostGIS
- Mapserver 5.x WFS con fuentes de datos shapefiles y PostGIS
- Mapserver 5.x WCS con fuentes de datos raster.
- Geoserver 1.5.4 WFS con fuentes de datos PostGIS y shapefiles.

Como requisitos previos debe disponerse de una instalación de Mapserver y de Geoserver respectivamente.

Una vez instalada la extensión, aparece en el gestor de pantalla un nuevo tipo de documento, el documento Publicación, tal y como se aprecia en la figura 16.

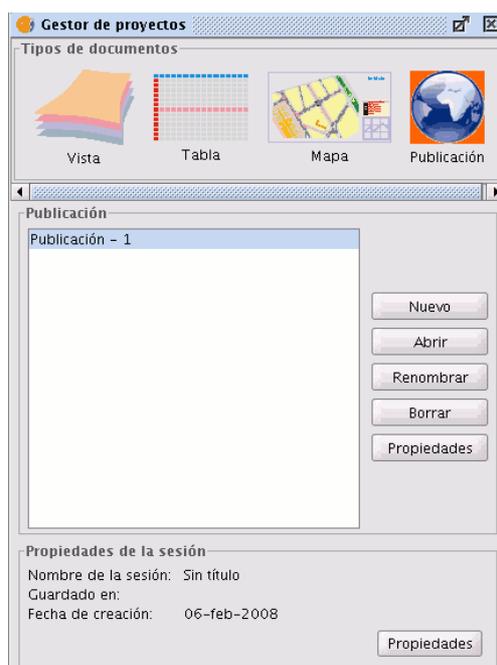


Fig. 16. Ventana del gestor de proyectos de gvSIG con extensión de publicación.

La publicación consiste en los siguientes pasos:

- Configuración de las vistas de gvSIG con las fuentes de datos y leyendas asociadas (para el caso de WMS) que se desea publicar. Estas deben ser capas raster o vectoriales.
- Creación de un documento de publicación.
- Selección del servidor y servicio a configurar.
- Configuración de los parámetros específicos del servidor
- Configuración de los parámetros específicos del servicio
- Adición de los recursos a publicar provenientes de las vistas de gvSIG.
- Configuración de los parámetros específicos de cada uno de los recursos.
- Generación de la configuración.

Por otra parte, habrá que tener en cuenta que las fuentes de datos deben estar accesibles tanto desde gvSIG como desde el servidor. Por ejemplo, los ficheros deberán tener permisos de lectura para el usuario que ejecuta el CGI de Mapserver.

Respecto a la creación del documento a publicar, una vez se dispone de las vistas cargadas en gvSIG, se crea un documento nuevo de tipo Publicación y se configura el servidor indicando su URL, así como el tipo de servidor y servicio. El siguiente paso es configurar los parámetros del servidor, que en función de su tipo son:

Servidor	Parámetros básicos	Parámetros avanzados
MapServer	Fichero de configuración	Depuración
		Directorio Temporal
		Directorio de datos
Geoserver	Usuario y contraseña	Máx. Features
	Directorio de configuración	Nivel de log

Una vez configurado el servidor, se indican, en la pestaña “Servicio”, los parámetros del servicio: Título, resumen y dirección URL.

Por último, en la pestaña “Recursos” que se muestra en la figura 17, añadiremos los recursos que queremos publicar, que en esta versión de gvSIG, sólo pueden ser las vistas y sus capas.

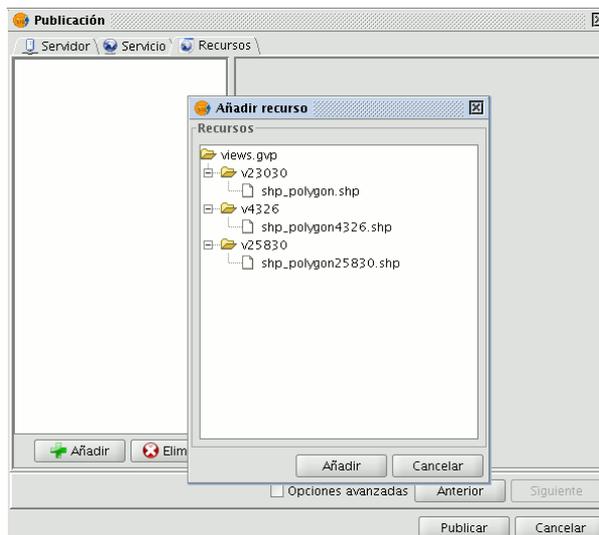


Fig. 17. Pantalla gvSIG para la publicación de servicios

Una vez pulsado el botón de Aceptar, el programa nos informa del resultado de la publicación.

Es importante tener en cuenta, que cuando se crea una publicación, la configuración generada contiene referencias a las fuentes de datos en forma de rutas absolutas a los ficheros, por tanto, una publicación sólo funcionará en un servidor que está en la misma máquina que ejecuta gvSIG. Esta limitación puede solventarse mediante un mecanismo de cambio de rutas absolutas.

Capítulo 6

Diseño de un cliente Web Processing Services para gvSIG. Catálogo de requisitos, casos de uso y diseño de la aplicación.

El servidor instalado 52nWPS dispone de un cliente implementado, al que podemos acceder desde un navegador de Internet. En este proyecto se pretende implementar un cliente gvSIG, incorporando, a las ya numerosas utilidades que actualmente ofrece este programa, la posibilidad de hacer uso de los servicios de procesamiento proporcionados por el servidor.

En el diseño de este cliente, se ha tomado como referencia los clientes incorporados en la instalación del servidor.

Recordemos que los procesos que puede solicitarse a dicho servidor se basan en los algoritmos SimpleBufferAlgorithm, TopologyPreservingSimplificationAlgorithm y DouglasPeuckerAlgorithm.

6.1 Catálogo de requisitos

El objetivo del proyecto es implementar en gvSIG en un cliente WPS.

El cliente podrá conectarse con un servidor WPS. En concreto, para esta práctica se utilizará un servidor 52nWPS.

El servidor WPS está instalado en la misma máquina local que el cliente.

La aplicación gvSIG trabaja sobre extensiones. Construiremos una nueva extensión gvSIG capaz de obtener los servicios de un servidor WPS.

El usuario de gvSIG tendrá disponible una opción para conectarse con otros servidores. Por defecto, aparecerá el servidor local.

Una vez obtenida respuesta del servidor, el cliente presentará al usuario los procesos que realiza dicho servidor. Para ello realizará una petición de GetCapabilities y mostrará una ventana específica para ello.

Así mismo, el usuario podrá seleccionar un proceso para consultar su descripción. Para ello, el cliente podrá realizar peticiones DescribeProcess de cada uno de los procesos obtenidos en la petición GetCapabilities.

El cliente recogerá la respuesta de una petición DescribeProcess.

El cliente realizará peticiones al servidor para que este ejecute cualquiera de los procesos que soporta. Las peticiones se realizarán para los procesos del servidor 52north con los parámetros de entrada obligatorios, que recogerá del propio gvSIG, bien desde alguna de las capas ya existentes, o bien mediante cuadros de diálogo presentados al usuario.

El cliente recibirá la respuesta del servidor y presentará los resultados al usuario, bien creando nuevas capas con la información procesada o bien mediante cuadros de diálogo que informen del resultado. Esta última opción se utilizará cuando la ejecución no haya sido satisfactoria.

Desde el cliente se podrá preguntar al servidor por el estado de la ejecución de su proceso. Esta información será mostrada en pantalla al usuario.

En concreto, para el servidor local se podrán realizar peticiones para extraer nueva información mediante los procesos siguientes:

- org.n52.wps.server.algorithm.SimpleBufferAlgorithm: Crea un área de influencia alrededor de un polígono simple. Acepta el polígono como un GML y proporciona otro GML de salida con las características del polígono de buffer.
- org.n52.wps.server.algorithm.simplify.DouglasPeuckerAlgorithm: Se trata de un algoritmo de simplificación de polígonos que utiliza la implementación JST. Proporciona un GML de salida con la geometría pulida.
- org.n52.wps.server.algorithm.simplify.TopologyPreservingSimplificationAlgorithm: Se trata igualmente de un algoritmo de simplificación que utiliza la implementación JST pero no soporta la topología aware.

La interfaz gráfica del cliente será acorde a la interfaz del programa gvSIG.

Se mostrarán todos los cuadros de diálogo necesarios para realizar las llamadas al servidor, y ejecutar sus procesos.

6.2 Diseño del cliente WPS

La arquitectura aplicada para el diseño del cliente WPS está basada en la establecida por el framework propio de gvSIG.

Las alternativas para diseñar el cliente WPS dentro del marco de gvSIG son varias. En este trabajo se han valorado dos:

- Crear una opción de menú del gvSIG que lance el cliente WPS.
- Envolver el cliente WPS en un geoproceto.

La primera alternativa simplifica la solución, sin embargo, teniendo en cuenta que un proceso WPS es en si mismo un geoproceto, con la única diferencia que su ejecución no se realiza en la misma máquina, se opta por analizar la viabilidad de la segunda alternativa.

Un geoproceto es un proceso que recibe una información espacial de entrada y como resultado genera una información espacial de salida. En concreto, recibe como parámetros de entrada una o más capas de información cartográfica, y como resultado de su ejecución produce una nueva capa, resultante de aplicar una serie de criterios lógicos, espaciales o numéricos sobre los elementos geográficos que forman parte de las capas con las que se está operando.

Los procesos realizados en un servidor WPS cumplen perfectamente la definición anterior y por tanto el cliente a desarrollar se implementará como parte del Gestor de Geoprocetos, con la especificidad necesaria para presentar al usuario procesos que no están implementados dentro del propio gvSIG sino en un servidor remoto.

Siguiendo las recomendaciones para el desarrollo de nuevas extensiones de gvSIG, se ha tomado como punto de partida los proyectos extGeoprocessing y extGeoprocessingExtension que implementa geoprocetos y mediante los cuales vamos a poder incorporar nuevos procesos.

GvSIG proporciona mecanismos para incorporar nuevos geoprocetos al Gestor de Procesos (geoprocessmanager). Un geoproceto se compone de los siguientes elementos:

- El propio geoproceto.
- Un panel gráfico.
- El fichero html descriptivo del geoproceto.

- El fichero con la imagen descriptiva del geoproceto.
- El controlador del geoproceto.

El GeoprocessManager tiene asociada una extensión en la que se construye el componente y se añaden los geoprocetos del core (unión, diferencia, ..). Así mismo, el GeoprocessManager construye un punto de extensión con el API ExtensionPoint del proyecto IverUtiles. Este punto de extensión, denominado “GesoprocessManager” se crea al inicializar la extensión Andami del GeoprocessManager.

El procedimiento para crear un nuevo geoproceto e incluirlo en el GeoprocessManager, consiste en construir una extensión Andami, inicializar la extensión añadiendo el geoproceto nuevo al punto de extensión y hacer que el GeoprocessManager recupere este punto de extensión.

A continuación se describen los principales aspectos que distinguen nuestro cliente del resto de geoprocetos:

- a) Registro de los procesos remotos. Los geoprocetos que incluye gvSIG están implementados internamente dentro del propio código, mediante la extensión extGeoprocessing y se registran como puntos de la extensión “GeoprocessManager”. Todos los geoprocetos registrados se presentan al usuario en una estructura tipo árbol. El registro de los geoprocetos básicos se realiza al inicializarse este panel, el resto se registran a posteriori, de forma dinámica. Los procesos que atiende el servidor remoto deberían ser incluidos también de manera dinámica, pero realizando el registro de los mismos una vez se haya obtenido la respuesta a la operación getCapabilites del servidor remoto seleccionado. El diseño de geoprocetos que proporciona gvSIG en esta versión no permite realizar dicho registro a posteriori, por lo que se ha optado por implementar específicamente los tres geoprocetos servidos por el servidor WPS.
- b) Parámetros de entrada del geoproceto. Los parámetros de entrada varía para cada proceso remoto. Se deben recoger del parámetro “DataInputs” de la petición DescribeProcess. El panel de recogida de estos datos debería solicitar de manera dinámica los parámetros de entrada para cada proceso del servidor remoto. Sin embargo, para realizar una implementación dinámica nos encontramos de nuevo nos encontramos limitados por el propio gvSIG. Éste proporciona la interface de alto nivel Igeoprocess que incluye las consideraciones relacionadas con los parámetros

de entrada y los tipos de geoprocursos overlay, mono-operando y bi-operando. La especificación de WPS no requiere que los procesos operen exclusivamente sobre datos georeferenciados, puede operar sobre cualquier tipo de dato del estándar GML, alguno de los cuales no puede ser asociados a capas gvSIG, y por tanto no puede vincularse dinámicamente los parámetros de entradas con las capas de entrada. Dado que la implementación se va a realizar de manera específica para cada geoprocuro y que conocemos de antemano los parámetros de entrada, se hará uso de la interface de gvSIG para el recubrimiento de los geoprocursos.

- c) Salida del proceso remoto. La salida de cada proceso remoto. Se debe recoger del parámetro "ProcessOutput" de la petición DescribeProcess. El panel de recogida de datos también tendrá que tener en cuenta si debe pedirle al usuario algún dato para recoger dicha salida. El tratamiento será similar ha comentado para los parámetros de entrada.
- d) La ejecución del proceso remoto. En el caso del resto de geoprocursos, su lógica está implementada en el propio gvSIG. En el caso de los procesos remotos, gvSIG no incorpora código para ello, es el propio servidor el que se encarga del procesamiento.

6.2.1 Casos de Uso del cliente WPS

La interacción del usuario con gvSIG para lanzar un cliente que cumpla los requisitos planteados no supone una gran complejidad. En el diseño descrito en el apartado anterior se plantean dos soluciones válidas y ambas estarán disponibles al usuario, tal y como muestra el diagrama de casos de uso del la figura 18. Se podrán realizar dos secuencias de acciones diferentes para lanzar el cliente WPS.

La primera secuencia se corresponde con el diseño de una opción de menú del gvSIG en la que se ha identificado los siguientes casos de uso:

- Conectar WPS: Se trata de un caso de uso muy sencillo, que consiste en la opción de menú que va a presentar el panel para la conexión con el servidor.
- Seleccionar servidor: El usuario deberá introducir la dirección URL del servidor al que se desea conectar.
- Conectar servidor: Consiste en una opción que ejecuta la conexión con el servidor WPS y devuelve los procesos que ofrece el servidor.

- Consultar Proceso: El usuario podrá solicitar al servidor las características de cada uno de los procesos.



Fig. 18. Casos de uso

La segunda secuencia de acciones se corresponde con el diseño de un geoproceso que ejecute uno de los procesos que ofrece el servidor WPS. Los casos de uso identificados son:

- Abrir capa: Consiste abrir una nueva capa mediante la opción de “añadir capa” que ya ofrece gvSIG.
- Llamar al gestor de geoprocesos: Consiste en abrir la ventana del gestor de geoprocesos mediante la opción “Gestor Geoprocesos”.
- Abrir Geoproceso WPS: El usuario seleccionará el geoproceso WPS del árbol de geoprocesos mostrado y desde el botón “Abrir Geoproceso” podrá llamar a dicho geoproceso.
- Ejecutar proceso: El usuario introducirá los parámetros de entrada y la capa donde desea que se muestren los resultados y lanzará la ejecución del geoprocesos.
- Visualizar salida: Consiste en mostrar en pantalla la capa que ha recogido los datos de salida.

- Ver estado de procesos: Consiste en mostrar el estado de la ejecución del geoproceso.

Los dos últimos casos de uso, visualizar salida y ver estado proceso WPS, no estarán implementados en esta versión del cliente.

6.2.2 Diseño de clases del sistema.

El geoproceso WPS requiere de la creación de nuevas clases, algunas de ellas serán una especialización de otras del proyecto ExtGeoprocess, así como de la modificación de algunas clases ya existentes. El diagrama de clases de la figura 19 contiene todas estas clases.

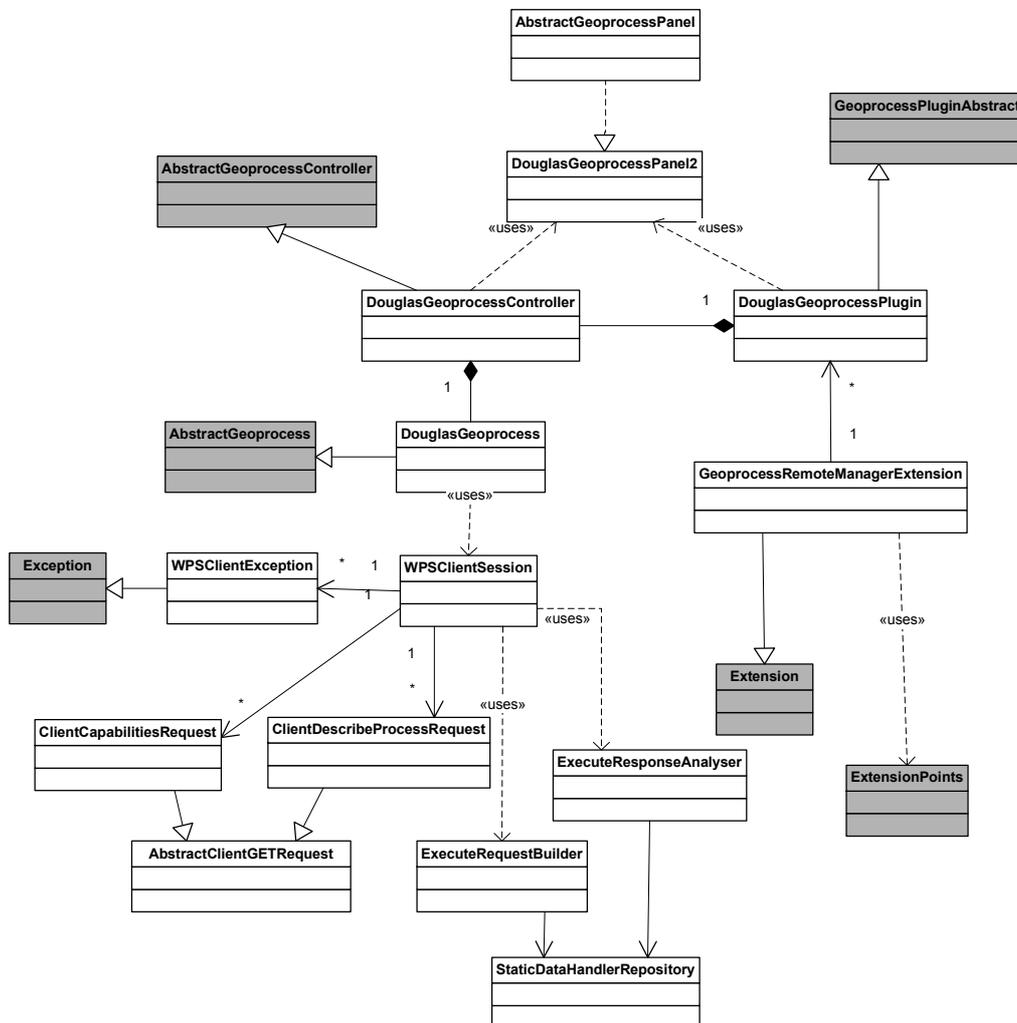


Fig. 19. Diagrama de clases de la extensión ExtWPSClient

A continuación se detallan las clases diseñadas para el cliente WPS y que se han añadido al actual del modelo de clases gvSIG.

DouglasGeoprocess: Representa el geoproceto. Se encarga de envolver el cliente WPS. Proporciona métodos para el paso de parámetros de una ejecución, para el chequeo de las precondiciones del geoproceto y para ordenar la ejecución del geoproceto al servidor remoto, así como para obtener la salida.

GeoprocessRemoteManagerExtension: Se trata de la clase que crea la extensión en Andami. Al inicializarse se añade el geoproceto DouglasPeucker como un punto de extensión de GeoprocessManager.

DouglasGeoprocessPlugin: Se trata de una especialización de la clase GeoprocessPluginAbstract. Proporciona todos los accesos a los componentes del DouglasGeoprocess y se encarga de recuperar el panel asociado, la imagen, el controlador y la estructura de nodos/hojas para la presentación en el árbol.

DouglasGeoprocessController: Se trata de una especialización de la clase AbstractGeoprocessController. Proporciona métodos para establecer el panel asociado al DouglasGeoprocess, para obtener el propio DouglasGeoprocess y para ordenar su ejecución al servidor.

DouglasGeoprocessPanel2: Proporciona un panel para la recogida de parámetros de entrada y salida que necesita el proceso. Se trata de una especialización de la clase AbstractGeoprocessPanel que modela los métodos comunes a todos los paneles de los geoprocetos.

WPSClientSession: Crea un cliente WPS que será el encargado de acceder al servidor y gestionar las peticiones de operaciones de getCapabilities, DescribeProcess y Execute, así como de recoger las respuestas. Está implementado con el patrón singleton, por lo que no podrá haber más de un cliente WPS ejecutándose

WPSClientException: Se trata de la clase encargada de gestionar las excepciones del cliente.

AbstractClientGETRequest: Se trata de una clase abstracta que construye la cadena de Request con los parámetros comunes para realizar una petición de operación vía GET.

ClientCapabilitiesRequest: Es una especialización de la clase AbstractClientGETRequest que incorpora el parámetro relativo a la operación a ejecutar, para el caso de la operación getCapabilities.

ClientDescribeProcessRequest: Es una especialización de la clase AbstractClientGETRequest que incorpora a la cadena de petición Request los parámetros relativos a la operación solicitadas, indicando que se trata de DescribeProcess, y los identificadores de los procesos para los cuales se solicita la descripción.

ExecuteRequestBuilder: Es la clase encargada de construir el Request de la sentencia Execute.

ExecuteResponseAnalyser: Es la clase encargada de devolver la lista de datos de salida obtenidos de la ejecución.

StaticDataHandlerRepository: Inicializa una factoría para la generación y el análisis de los datos estáticos. Se utiliza para analizar el contenido del parámetro OutputProcess de la operación execute.

Este modelo presenta algunas restricciones al cliente implementado, en concreto se han detectado las siguientes limitaciones:

- La capa de salida de cualquier geoproceto sólo puede ser shape.
- Las peticiones de las operaciones getCapabilities y DescribeProcess sólo están implementadas vía GET.
- La petición de la operación execute sólo está implementada vía POST.

Capítulo 7

Desarrollo de la aplicación. Cuestiones

fundamentales de la implementación realizada.

La codificación de la aplicación parte de la versión gvSIG 1.1.1 (<http://www.gvsig.gva.es/index.php?id=gvsig11&L=0>). Se han seguido las instrucciones del fichero leeme.txt que acompaña a los ficheros fuentes para su instalación. El único problema encontrado ha sido debido a la instalación de los fuentes en un directorio cuyo nombre llevaba espacios el blanco. La instalación desde el repositorio de Subversión también se ha intentado pero resulta más laboriosa que la primera opción.

El desarrollo se ha realizado como un nuevo plugin de gvSIG. Para ello se ha creado en eclipse un nuevo proyecto llamado ExtWPSClient. La estructura del proyecto, que se muestra en la figura 20, sigue las directrices marcadas en la documentación para desarrolladores del gvSIG (<https://gvsig.org/web/docdev/>).

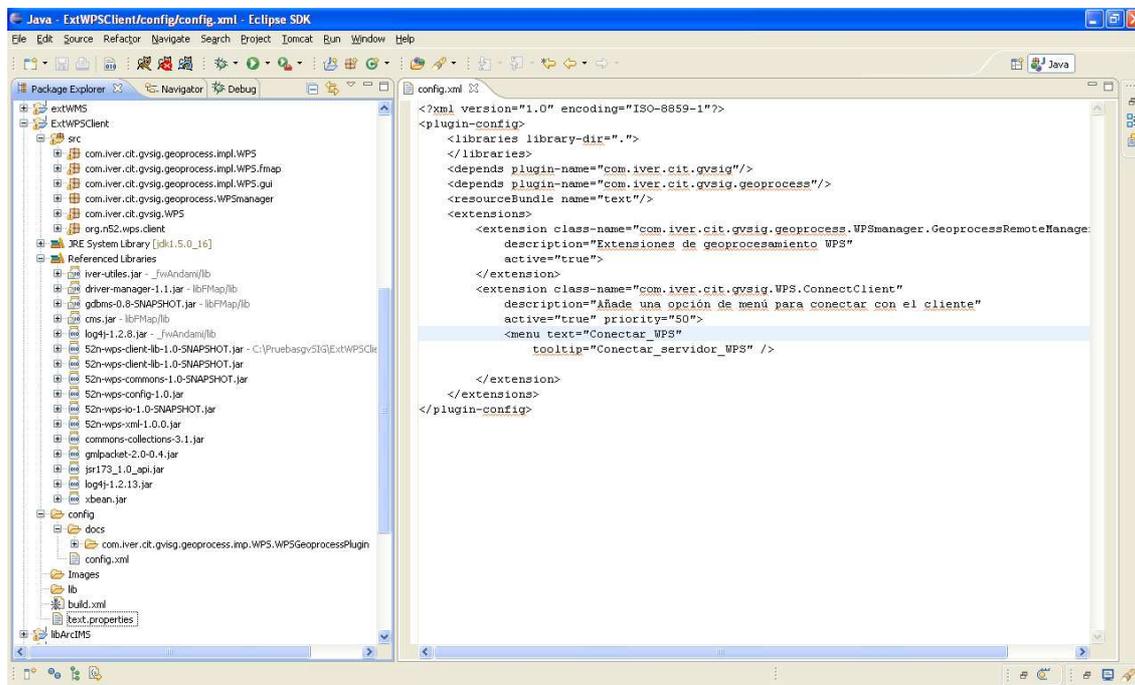


Fig. 20. Pantalla de eclipse con proyecto ExtWPSClient

El proyecto tiene las siguientes carpetas y ficheros:

- src: Carpeta que contiene los paquetes y las clases de la extensión.
- lib: Carpeta que contiene el fichero .jar que contiene las clases del plugin y todas las librerías utilizadas por él.
- config: Carpeta que contiene un fichero config.xml donde se especifican el nombre y las dependencias del plugin, se registran las extensiones, sus menús, y las clases que implementan el cliente WPS.
- El fichero build.xml: Compila el proyecto, contiene tanto el nombre del plugin, como las librerías y ficheros que necesita para funcionar. Se ejecuta como una tarea ant que genera la extensión en Andami, permitiendo la ejecución del plugin dentro de gvSIG. Al ejecutar este fichero se crea una carpeta en el directorio `_fwAndami\gvSIG\extensiones\com.iver.gvsig.WPSClient`
- El fichero text.properties, necesario para indicar los textos en castellano en gvSIG dado que se trata de una herramienta multilingüe que lleva implementado un sistema de traducciones

Se han definido dos extensiones: la extensión `com.iver.cit.gvsig.WPS.ConnectClient` que crea una opción de menú para lanzar el cliente WPS y la extensión `com.iver.cit.gvsig.geoprocess.WPSmanager.GeoprocessRemoteManagerExtension` que envuelve crea un proceso para al algoritmo de simplificación Douglas-Peucker. En los dos apartados siguientes se detallan ambas extensiones.

La solución implementada tiene una serie de dependencias con otros proyectos y con librerías externas, mostradas en la figura 21, que deben ser indicadas en el fichero build.xml. De esta manera, se copiarán en el directorio `_fwAndami\gvSIG\extensiones\com.iver.cit.gvsig\lib` al realizar la compilación del proyecto, todas las librerías necesarias para su correcta ejecución.

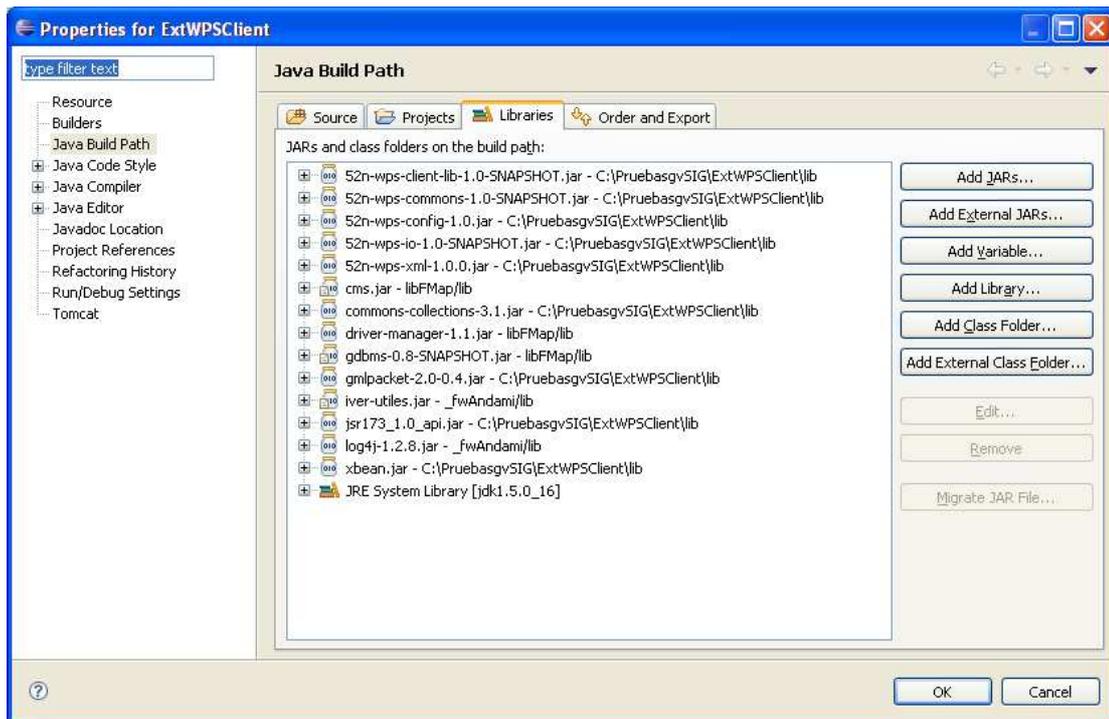


Fig. 21. Pantalla de dependencias de la extensión ExtWPSClient en gvSIG

7.1 Extensión ConnectClient.

A partir de la estructura del nuevo plugin ExtWPSClient, se han creado los siguientes paquetes y clases:

- `com.iver.cit.gvsig.WPS`. Contiene las clases de la extensión.
 - ✓ `ConnectClient.java`: Implementa una extensión para la conexión al servidor WPS.
 - ✓ `panelConectWPS.java`: Define el panel que permite seleccionar el servidor y mostrar la lista de procesos y las características de cada uno de ellos. La figura 21 muestra el aspecto visual de dicho panel.
- `org.n52.wps.client`: Contendrá las clases encargadas de la comunicación con el servidor WPS para las operaciones de `getCapabilities` y `DescribeProcess`. Estas clases coinciden con algunas de las definidas en el cliente lib que incluye el servidor WPS, y que ya se han comentado en el apartado del diseño de clases. En concreto, para esta extensión se utilizan las clases: `WPSClientSession`, `WPSClientException`, `ClientCapabilitiesRequest`, `ClientDescribeProcessRequest` y `AbstractClientGETRequest`.

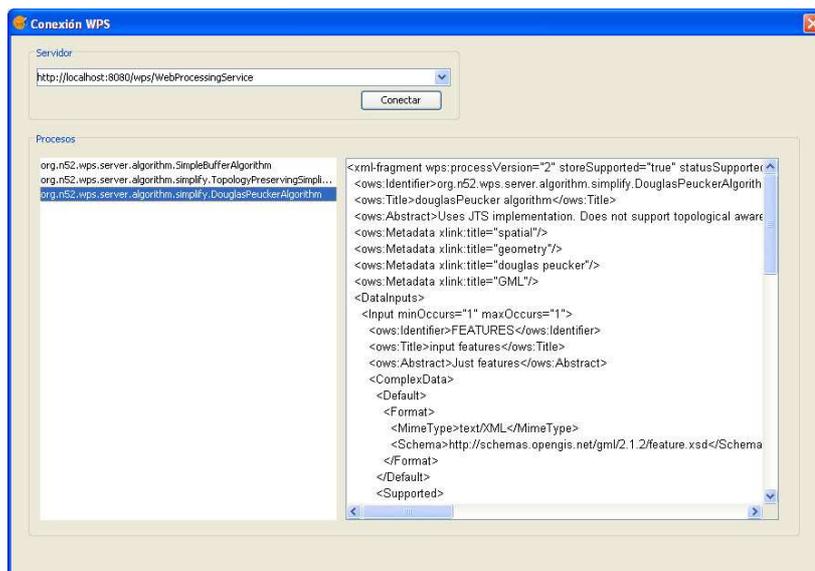


Fig. 22. Pantalla de conexión a un servidor WPS

El fichero config.xml contiene la siguiente definición para esta extensión:

```
<extension class-name="com.iver.cit.gvsig.WPS.ConnectClient"
  description="Añade una opción de menú para conectar con el cliente"
  active="true" priority="50">
  <menu text="Conectar_WPS"
    tooltip="Conectar_servidor_WPS" />
</extension>
```

El menú “Conectar WPS” está siempre visible y activo en gvSIG, puede ejecutarse desde cualquier punto del programa. La pantalla para de conexión se muestra en la figura 22.

7.2 Extensión GeoprocessRemoteManagerExtension.

Tal y como se ha comentado anteriormente, en esta versión del cliente, se ha implementado el geoproceto Douglas-Peucker. Sin embargo, se ha creado una estructura de paquetes genérica para los procesos WPS de manera que las diferentes clase de cada geoproceto se agrupen en los mismos paquetes. En concreto, los paquetes creados son:

- com.iver.cit.gvsig.geoprocess.imp.WPS: Contendrá las clases plugin y controller de los geoprocetos.
- com.iver.cit.gvsig.geoprocess.WPSmanager: Contendrá la clase que registra los geoprocetos WPS.
- Com.iver.cit.gvsig.geoprocess.imp.fmap: Contendrá las clases encargadas de envolver los geoprocetos WPS.

- com.iver.cit.gvsig.geoprocess.impl.WPS.gui: Contendrá las clases que implementan los paneles gráficos.

7.2.1 Registro del geoprocreso

La clase encargada de registrar el nuevo geoprocreso como un punto de extensión de GeoprocessManager es GeoprocessRemoteManagerExtension. En esta clase se ha incluido el siguiente método para incluir un punto de extensión al GeoprocessManager:

```
public void initialize() {
    ExtensionPoints extensionPoints = ExtensionPointsSingleton.getInstance();
    extensionPoints.add("GeoprocessManager", "WPS", WPSGeoprocessPlugin.class);
}
```

De esta manera, se realiza el registro al inicializarse el Gestor de Geoprocresos, junto con el resto de geoprocresos. El punto de extensión lleva asociado la clase DouglasGeoprocessPlugin.class que se encarga de definir la estructura del geoprocreso dentro del árbol de geoprocresos, tal y como se muestra en la Fig. 20, así mismo, se encarga de inicializar el controlador DouglasGeoprocessController() y el panel de recogida de parámetros DouglasGeoprocessPanel2, que permitirá al usuario lanzar la ejecución de proceso una vez haya introducido las capas de entrada y salida y el parámetro de tolerancia.

En caso de añadir nuevos geoprocresos al cliente, estos serán registrados en esta misma clase.

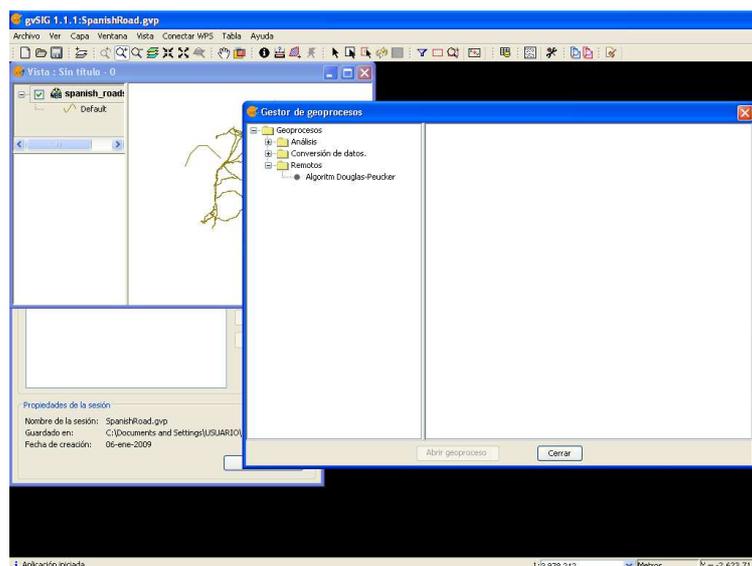


Fig. 23. Presentación del árbol de Geoprocresos.

7.2.2 Panel gráfico.

Tal y como se ha comentado en el diagrama de clases, DouglasGeoprocessPanel2 es la clase que define el panel gráfico del geoprocreso. Se trata de una clase que extiende a la clase abstracta AbstractGeoprocessGridbagPanel, y en la que se especializan los métodos addSpecificDesign, para incluir el campo de pantalla “Tolerancia”, y getTolerancia, para devolver el valor recogido por pantalla al controlador del proceso.

7.2.3 Controlador del geoprocreso.

El controlador del geoprocreso es el encargado de lanzar el panel DouglasGeoprocessPanel2 y de lanzar el DouglasGeoprocess, así como de presentar los resultados.

7.2.4 El geoprocreso Douglas-Peucker.

A diferencia de otros geoprocresos implementados en gvSIG, esta clase no realiza el procesamiento de los datos, se limita a solicita al servidor remoto su ejecución. Para ello, previamente debe inicializar una instancia de la clase WPSClientSession, comprobar que se cumplen las precondiciones marcadas por la interface Igeoprocess sobre las capas de entrada y salida, y compone la petición de execute y así como recoger la respuesta del servidor. Estas dos últimas operaciones se realizan a través de las clases ExecuteRequestBuilder y ExecuteResponseAnalyser.

7.2.5 Parámetros InputData y OutputData.

Los datos de entrada y de salida soportados por WPS pueden tener tres formatos distintos: ComplexData, LiteralData y BoundingBox. En el geoprocreso Douglas-Peucker intervienen los dos primeros.

Antes de solicitar la ejecución del geoprocreso al servidor, deben realizarse las transformaciones de la capa de entrada a datos de tipo ComplexData, que son los que acepta el servidor. Este punto no ha sido resuelto en este cliente, y por tanto la ejecución del geoprocreso no puede realizarse.

Concretamente, para enviar la capa al geoprocreso se recoge previamente las geometrías que forman la capa en un objeto de la clase FeatureCollection, gvSIG incorpora esta clase en su librería org.cresques.px.gml.FeatureCollection. El servidor está esperando recibir un objeto de la clase FeatureCollection de la librería org.geotools. A pesar de

tratarse de dos clases similares, parecen no ser compatibles, lo que produce en tiempo de ejecución un error al realizar el casting de dichas clases.

7.3 Mejoras propuestas

A la vista de lo expuesto anteriormente, es necesario incorporar una serie de mejoras al cliente desarrollado para conseguir ejecutar correctamente la operación de execute en el servidor WPS, así como para generalizar la implementación, de manera que pueda ser utilizada con otros servidores WPS.

Se cree necesario revisar el modelo de clases diseñado incorporando:

- Una interface para traducir los datos de gvSIG a los datos GML, acorde a lo establecido por el estándar WPS.
- Una interface de geoprocesos compatible con cualquier tipo de dato, sin necesidad de que se trata de datos georeferenciados, de manera que se pueda generar el panel de recogida de parámetros en función de la respuesta recibida a una operación DescribeProcess.
- Implementar el registro dinámico de las extensiones, que permita incorporar nuevos geoprocesos una vez que se haya llamado al gestor de procesos. De esta manera, se podría incluir un nuevo botón en este gestor para cargar geoprocesos desde servidores WPS.

Capítulo 8

Conclusiones.

La realización de este proyecto ha permitido tener una primera toma de contacto con los Sistemas de Información Geográfica, de los cuales no se tenía apenas conocimiento. Todos los aspectos trabajados, desde su comienzo con el estudio de las principales características de este sistema hasta los problemas encontrados durante el desarrollo del código, han sido enriquecedores tanto desde un punto de vista técnico como de gestión de proyectos.

En este sentido, cabe destacar que se han conocido algunos de los estándares OGC más utilizados actualmente en los SIG, y fundamentalmente las posibilidades que ofrece en estos momentos el estándar WPS. Prueba de ello es que actualmente ya se han desarrollado varios proyectos que implementan servidores WPS, así como diferentes clientes para los principales programas GIS.

La posibilidad de trabajar con un sistema de las características de gvSIG ha permitido poner en práctica los conocimientos de Java adquiridos en otras asignaturas de la carrera, trabajando en un entorno real y con una dimensión que hasta el momento no se había practicado. Así mismo, trabajar con estructuras de datos geográficos, dada su complejidad, ha puesto de manifiesto la utilidad de los estándares XML. Sin embargo, este ha sido el punto del proyecto que más dificultad ha supuesto, y que deberá de ser trabajado más detalladamente para continuar trabajando con los GIS.

Por último, señalar los beneficios que supone trabajar con arquitecturas abiertas e interoperables para la difusión de la información geográfica. La formación de una Infraestructura Europea de Datos Espaciales no podría llevarse a cabo trabajando sin la utilización de éstas.

En cuanto a las conclusiones desde un punto de vista de la gestión de proyectos, el proyecto se ha abordado con una planificación inicial que ha permitido cumplir los diferentes hitos marcados, si bien hay que señalar que la escasa experiencia en proyectos de este tipo, derivó en una desviación en las tareas prácticas del proyecto, prolongándose hasta la entrega final del proyecto las tareas de programación.

Lista de referencias

A continuación se resumen las referencias a otros textos utilizadas a lo largo de esta memoria:

UOC, Tablón de la asignatura en el Campus Virtual.

- Enunciado de la práctica

Generalitat Valenciana. Conselleria de Infraestructuras y transporte. Web de gvSIG.

- Página principal: <http://www.gvsig.gva.es>
- Código fuente de gvSIG:
<http://www.gvsig.gva.es/index.php?id=gvsig11&L=0&K=1>
- Documentación desarrolladores: <https://gvsig.org/web/docdev/>

Open Geospatial Consortium

- Página principal: <http://www.opengeospatial.org/>
- Especificación Web Processing Service 1.0.0.:
<http://www.opengeospatial.org/standards/wps>

Implementaciones WPS:

- Deegree Framework: <http://www.deegree.org/>
- WPSint: <http://wpsint.tigris.org/>
- pyWPS: <http://pywps.wald.intevation.org/index.html>
- 52North:
 - Página principal: <http://52north.org/>
 - Código fuente: <https://52north.org/svn/geoprocessing>
 - Instalación:
http://52north.org/index.php?option=com_jdownloads&Itemid=73&task=viewcategory&catid=22

Bibliografía

Gutiérrez Puebla, Javier; Gould, Michael (2000), SIG: Sistemas de Información Geográfica, Madrid, Ed. Síntesis.

Schut, Peter (2007); OpenGIS® Web Processing Service [OGC 05-007r7] en <http://www.opengeospatial.org/standards/wps>

Whiteside, Arliss (2007) OGC Web Services Common Specification [OGC 06-121r3] en www.opengeospatial.org/standards/common

Granell, C.; Díaz, L.; Esbrí, M.A.; Gould, M; Lladós, A; Contribuciones de una IDE a la e-Ciencia: Proyecto AWARE, en jidee06.uji.es/down/s31_Granell.pdf

Web processing with OGC WPS Specification, Theodor Foerster, en 52north.org/twiki/bin/view/Processing/GIdaysWPSworkshop

Percivall, George (2003), OGC Reference Model [OGC 03-040] en portal.opengeospatial.org/files/?artifact_id=3836

Manson Callejo, Miguel Angel; Servicios Web para la interoperabilidad entre programas definidos por el OGC, en <http://www.cp-idea.org/nuevoSitio/documentos/accesoDistribucion/OpenGis-WebServices.pdf>

Cepicky, Jachym, OGC Web Processing Service and it's usage, en gis.vsb.cz/GIS_Ostrava/GIS_Ova_2008/sbornik/Lists/Papers/025.pdf

Sancho-Jiménez, Gonzalo, Béjar, Rubén; Gayán, David; Muro-Medrano; Pedro R.; Un prototipo para acceder a comandos de Grass a través de servicios Web siguiendo la especificación de Web Processing Services” en www.orzancongres.com/administracion/upload/imgPrograma/P-016.pdf

Anexo I. Manual de usuario

La extensión ExtWPSClient incorpora a la plataforma gvSIG un cliente capaz de comunicarse con un servidor WPS. Este cliente puede ser utilizado desde una opción de menú de gvSIG para su conexión con el servidor y la obtención de los procesos que dicho servidor atiende, y otra a modo de Geoproceso, que permitirá ejecutar un proceso en el servidor WPS instalado en la máquina local.

1. Requisitos de la solución software

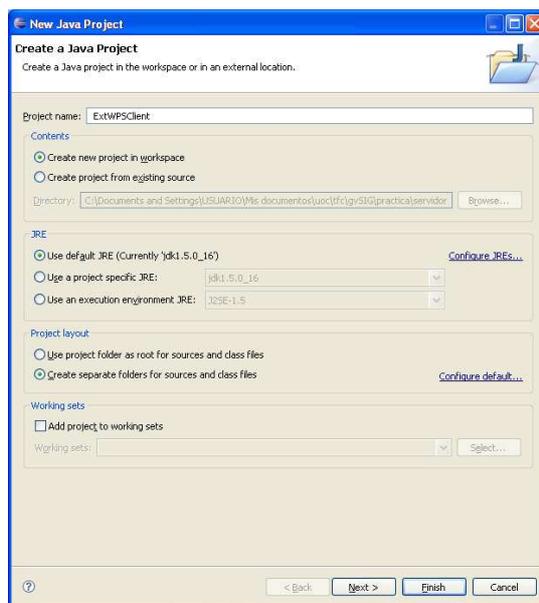
La solución software es una extensión sobre la plataforma gvSIG Desktop. Esta extensión no requiere de nuevos requisitos hardware a los indicados para la plataforma mencionada.

En cuanto a los requerimientos software, es necesario tener instalado el IDE Eclipse (<http://www.eclipse.org/downloads/>), un servidor Apache Tomcat 5.5. y un servidor WPS 52north (http://52north.org/index.php?option=com_jdownloads&Itemid=73&task=viewcategory&catid=22).

2. Instalación de los fuentes del cliente ExtWPSClient.

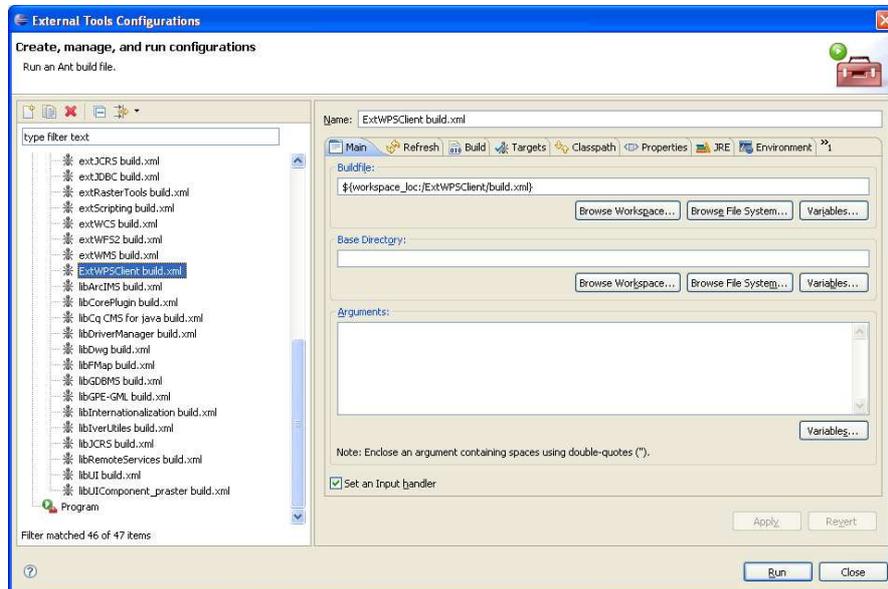
En primer lugar debe realizarse una instalación los fuentes de la plataforma (<http://www.gvsig.gva.es/index.php?id=gvsig11&L=0&K=1>), que puede realizarse siguiendo las instrucciones del fichero Leeme.txt incluido en la descarga.

Crear un nuevo proyecto Java en Eclipse llamado ExtWPSClient desde la opción File → New → Project, tal y como se muestra en la siguiente pantalla:



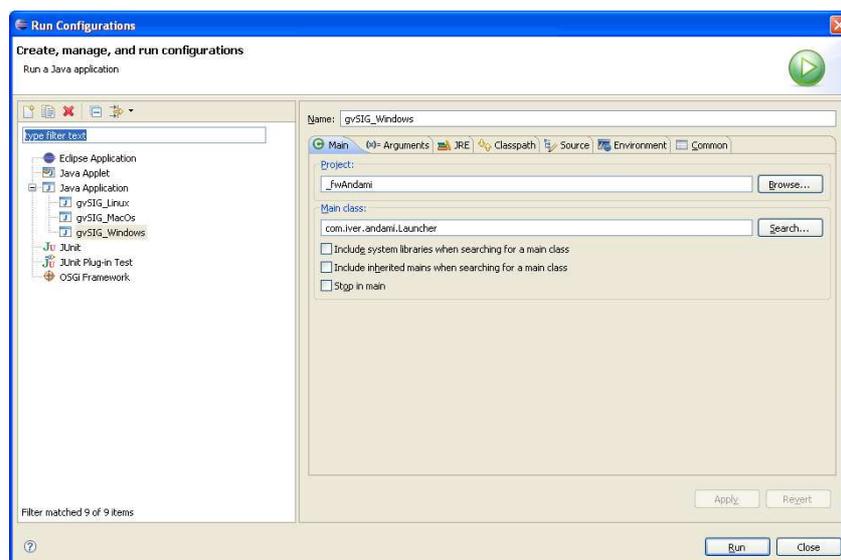
A continuación, importar el fichero ExtWPSClient.zip que contiene los ficheros fuentes de la solución desarrollada. Este fichero se acompaña en la entrega del proyecto.

Para compilar el proyecto, ejecutar el fichero build.xml desde la opción Run → External Tools Configurations → ExtWPSClient build.xml, tal y como se muestra en pantalla:



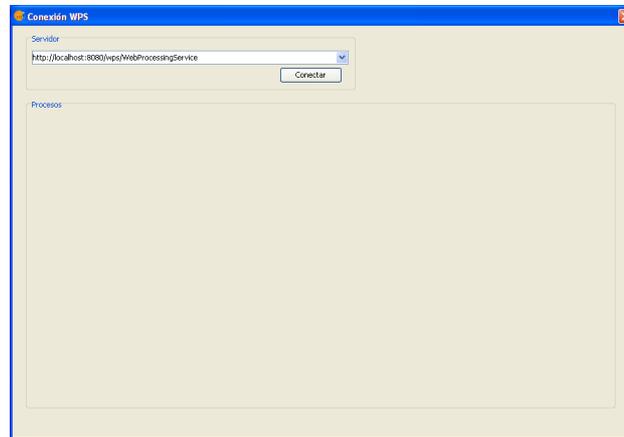
3. Ejecución de gvSIG con el cliente WPS.

La ejecución del cliente puede lanzarse desde el propio eclipse, desde la opción Run → Run Configuration, y seleccionado la configuración apropiada al entorno utilizado. En la pantalla siguientes se muestra la selección de la configuración de Windows.



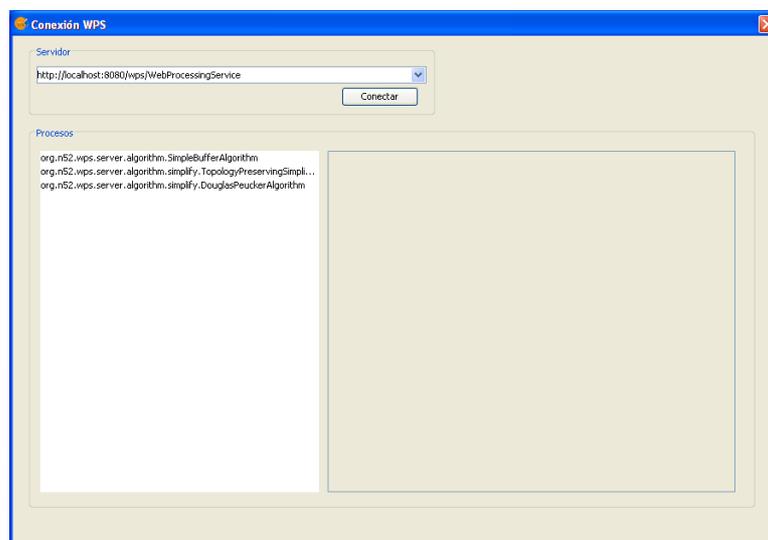
4. Conexión con el servidor WPS.

Una vez arrancado gvSIG, aparece en el menú superior de la pantalla la opción “Conectar WPS”. Este menú está visible en toda la aplicación y permite seleccionar un servidor, por defecto aparece el servidor local.



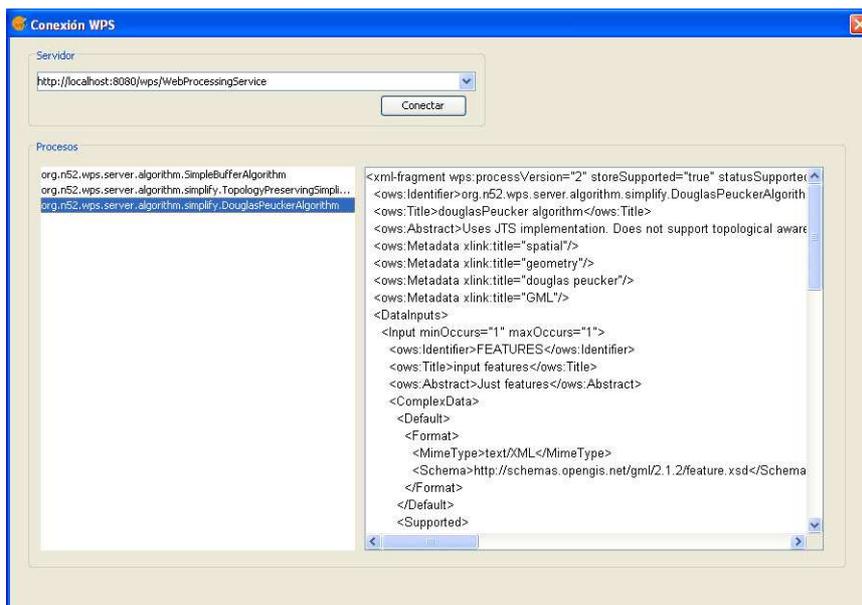
Para ejecutar esta opción, se debe tenerse levantado el servidor WPS.

Pulsando el botón conectar, se realiza una conexión al servidor y obtener la respuesta de una operación getCapabilities.



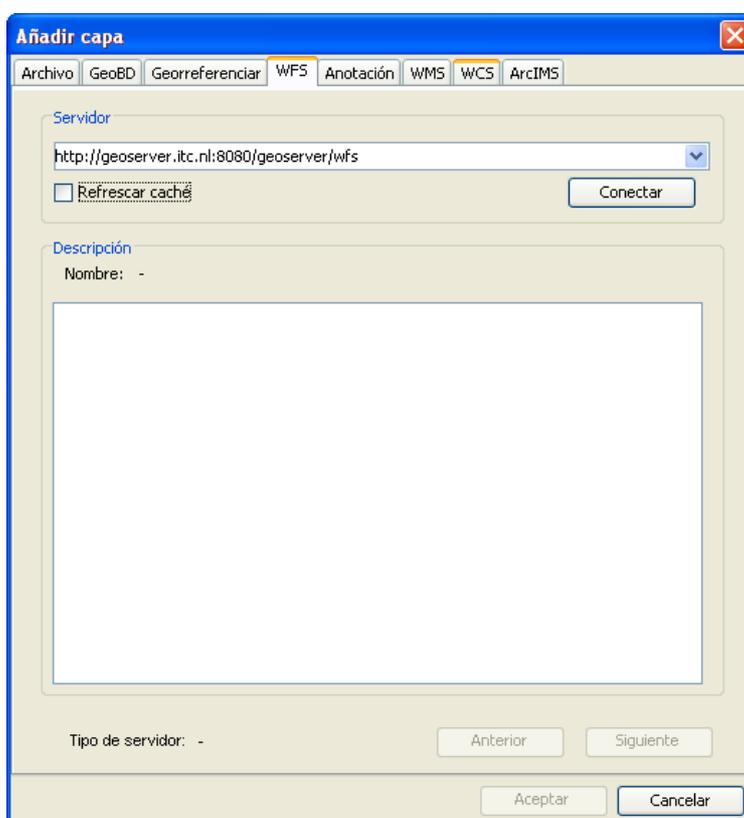
Si no puede realizarse la conexión, se muestra un cuadro de dialogo con el error.

Ahora se puede seleccionar un proceso de la lista de procesos que ofrece el servidor y obtener la respuesta a una operación DescribeProcess.

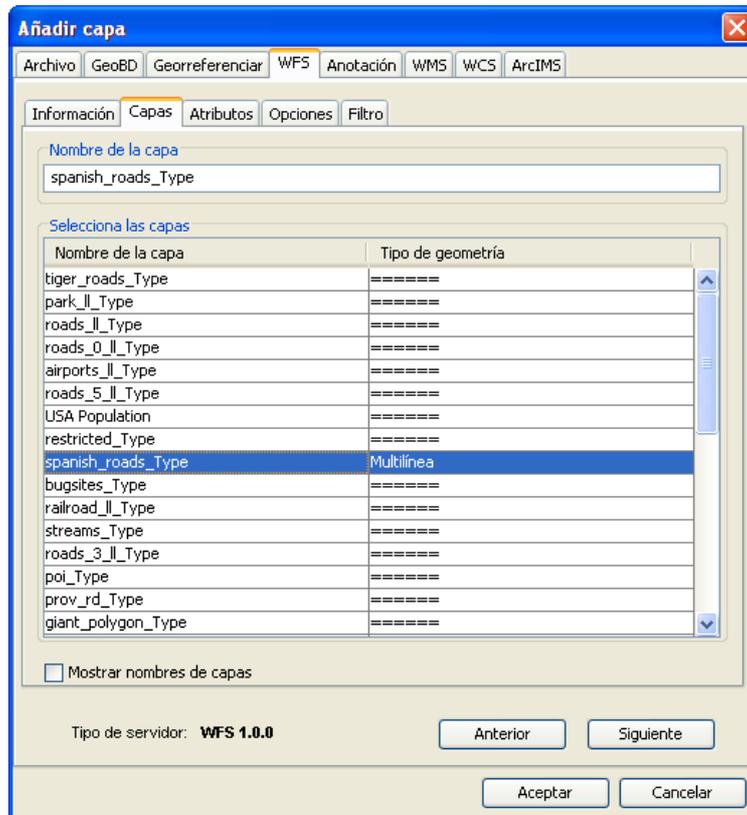


5. Ejecutar el geoproceto de simplificación del Algoritmo Douglas-Peucker.

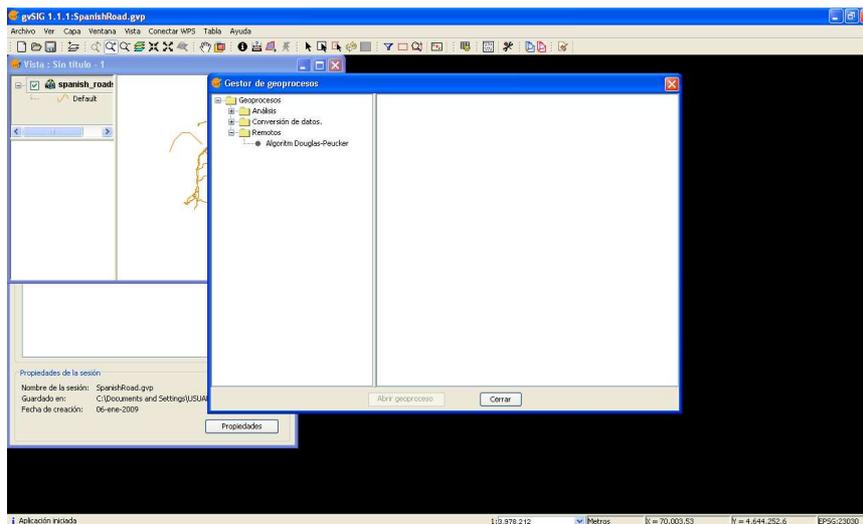
En primer lugar se debe cargar la capa a la que queremos aplicar el algoritmo de simplificación. Por ejemplo, podemos cargar desde un servidor WFS una capa de carreteras. Pulsamos la opción de “añadir capa” y seleccionamos la pestaña WFS:



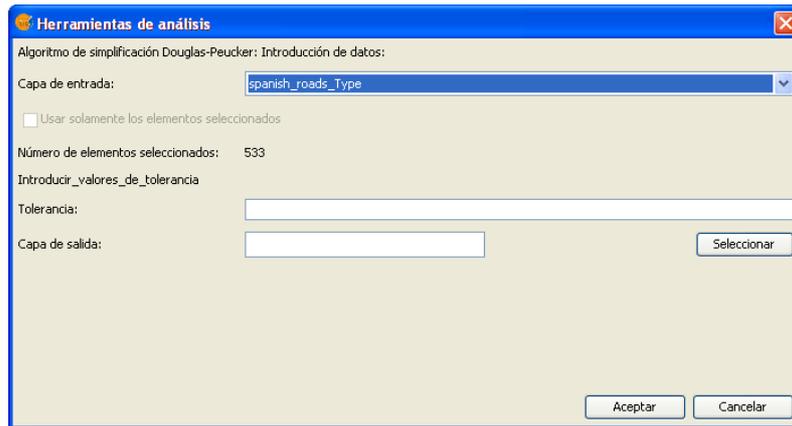
Desde el servidor <http://geoserver.itc.nl:8080/geoserver/wfs>, obtenemos la capa spanish_road_Type:



A continuación, desde la opción Vista del menú principal, llamamos al Gestor de Geoprocesos:



Seleccionamos el geoproceso Remoto, Algoritmo Douglas-Peucker y nos aparece la siguiente pantalla:



Vemos que se muestra por defecto la capa de entrada que hemos cargado anteriormente y que será la que se simplificará en el servidor WPS. Además, debemos introducir la tolerancia para realizar la simplificación y la capa de salida donde queremos que nos presente los resultado.

A partir de aquí, el cliente no está operativo.