Licencia Creative Commons

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.

UNIVERSIDAD OBERTA DE CATALUNYA

Ingeniería Técnica en Informática de Gestión

DISEÑO Y DESARROLLO DE UN CLIENTE WPS EN EL ENTORNO GVSIG

Memoria del Trabajo Final de Carrera

Alumno: Víctor Velarde Gutiérrez

Dirigido por: Jesús De Diego Alarcón

CURSO 2008-09 (Septiembre)

UNIVERSITAT OBERTA DE CATALUNYA

Resumen

Este documento es el resultado del Trabajo Final de Carrera (TFC) para la Ingeniería Técnica en Informática de Gestión por la UOC, realizado en el área de Sistemas de Información Geográfica durante el periodo de septiembre de 2008 a enero de 2009. En él se ha construido una extensión para gvSIG denominada **extWPS**, que permite ejecutar procesos a través de *Web Processing Service* (WPS), el protocolo promovido por el Open Geospatial Consortium (OGC). El software, elaborado en Java, se basa en el conjunto de librerías de la iniciativa 52north y tiene licencia GPL. Hace posible, una vez instalado sobre gvSIG, conectarse a un servidor WPS, ver los geoprocesos disponibles en él y ejecutarlos de manera remota.

Los fuentes, documentación e instalador de la extensión se encuentran disponibles en el repositorio web de Subversion siguiente: http://svn.assembla.com/svn/TFC_SIG_vvelarde

Palabras clave: *SIG*, *GIS*, *gvSIG*, *WPS*, *52north*.

Índice de Contenidos

Introducción 5

1. Presentación 6

1.1. Objetivos 6

1.2. Planificación 7

1.3. Entregables 10

2. Contexto tecnológico 12

2.1. Sistemas de Información Geográfica 12

2.2. Interoperabilidad y Open Geospatial Consortium 15

2.3. Web Processing Service y 52north 16

2.4. gvSIG y su carácter extensible 20

3. Desarrollo de la extensión 24

3.1. Configuración del entorno 24

3.2. Análisis de requisitos 25

3.3. Diseño 29

3.4. Codificación 35

3.4. Pruebas 37

3.5. Instalador 38

4. Conclusiones y recomendaciones 40

Glosario básico 44

ANEXO I. Bibliografía 45

ANEXO II. Plan de gestión de riesgos 46

Índice de Tablas

Tabla 1. Planificación de tareas del TFC 9

Tabla 2. Preguntas básicas para resolver con SIG. 13

Tabla 3. Herramientas del entorno de trabajo 24

Tabla 4. Caso de uso 01 – Obtener información de un servicio WPS 27

Tabla 5. Caso de uso 02 – Obtener información de un geoproceso 28

Tabla 6. Caso de uso 03 – Ejecutar un geoproceso 28

Índice de Ilustraciones

Ilustración 1. Componentes tecnológicos del proyecto 6

Ilustración 2. Fases y actividades principales del proyecto 9

Ilustración 3. Modelos vectorial y raster en un SIG 13

Ilustración 4. Servicios OWS principales 16

Ilustración 5. Operaciones del interfaz WPS 17

Ilustración 6. Cliente WPS en uDig (52north) 19

Ilustración 7. Capacidades OGC de gvSIG v1.1.2 21

Ilustración 8. Framework base de gvSIG 22

Ilustración 9. Espacio de trabajo Assembla 25

Ilustración 10. Casos de uso de extWPS 26

Ilustración 11. Pantalla 1ª del asistente "Añadir Capa" WPS en gvSIG (Servicio) ... 31

Ilustración 12. Pantalla 2ª del asistente "Añadir Capa" WPS en gvSIG (Geoproceso) 32

Ilustración 13. Diagrama general de paquetes y clases de extWPS 33

Ilustración 14. Javadocs de extWPS 35

Ilustración 15. Pruebas de extWPS con JUnit 38

Ilustración 16. Instalador de la extensión extWPS 39

Introducción

Esta memoria se estructura en 4 grandes apartados:

1. Una presentación del proyecto.
2. Un apartado de contexto tecnológico.
3. Una descripción del proceso de desarrollo de la extensión.
4. Unas conclusiones finales.

1. Presentación del proyecto

En este apartado se presenta el Trabajo Final de Carrera. Se describen los objetivos generales y específicos que se persiguen, el trabajo de planificación y el listado de los entregables generados.

2. Contexto tecnológico del proyecto

Aquí se dan las claves principales del ámbito en el que se enmarca el proyecto. Si el lector es ajeno a este área puede resultarle útil, proporcionando un repaso general sobre los Sistemas de Información Geográfica, la interoperabilidad (OGC y el protocolo WPS) y, finalmente, realizando una presentación del software gvSIG.

3. El desarrollo de la extensión para WPS

Se explica cómo se ha realizado el desarrollo del software, haciendo especial énfasis en las prácticas y las herramientas utilizadas para construir la extensión.

4. Conclusiones y recomendaciones

El documento termina con un apartado de conclusiones, que recoge también los principales problemas surgidos durante su desarrollo y posibles líneas futuras de trabajo.

Finalmente, se incluyen dos anexos: uno con referencias bibliográficas (*Anexo I*) y otro con el plan de gestión de riesgos del proyecto (*Anexo II*).

1. Presentación

Este apartado se divide en tres: *1.1 Objetivos*, *1.2 Planificación* y *1.3 Entregables*. Con el primero de ellos el lector conocerá qué se solicitó elaborar como TFC en el enunciado original y cuál ha sido el resultado de este trabajo. En el segundo, cuál fue la planificación del trabajo en fases y tareas.

Finalmente, en el último apartado de esta introducción encontrará un listado con los distintos entregables elaborados durante el proyecto.

1.1. Objetivos

El TFC en la UOC es una asignatura pensada para realizar un trabajo de síntesis de los conocimientos adquiridos en otras asignaturas de la carrera. Requiere ponerlos en práctica conjuntamente en un trabajo concreto, eminentemente práctico y centrado en el ejercicio profesional de la informática.

En este caso, el TFC pertenece al área de Sistemas de Información Geográfica y se apoya, especialmente, en los conocimientos derivados de las asignaturas de bases de datos, programación e ingeniería del software. Al inicio del proyecto, con su enunciado, se plantearon como sus objetivos:

a) analizar, diseñar y programar un cliente de *Web Processing Service* (WPS) para el programa gvSIG Desktop (objetivo central).

b) conocer gvSIG y configurar su entorno de desarrollo con Eclipse.

c) conocer los estándares OGC, en especial WPS.

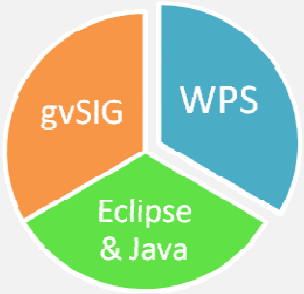

Ilustración 1. Componentes tecnológicos del proyecto

De esta manera, los componentes tecnológicos principales del proyecto han sido Java + gvSIG + WPS (Ilustración 1). Al comienzo se ofrecieron también algunas recomendaciones sobre software para trabajar con WPS y es ahí donde surgió la idea de trabajar con las librerías de 52north.

Con el proyecto finalizado se puede afirmar que los objetivos planteados han sido alcanzados satisfactoriamente. Como resultado del TFC se ha elaborado una extensión para gvSIG que se denomina **extWPS**. Una vez instalada sobre el software base en una versión 1.1.X, ésta permite ejecutar geoprocesos remotos a través del estándar WPS. ExtWPS ha sido programada en Java, utilizando el IDE Eclipse y se añade al resto de las extensiones que ya existían en gvSIG para otros estándares OGC como WMS, WFS o WCS.

En el proyecto se han utilizado múltiples recursos: las librerías de 52north, materiales formativos online sobre gvSIG, listas de correo y foros, documentos públicos del OGC, materiales de la UOC, bibliografía general de SIG, análisis de programas como uDig y Jump… pero, sobre todo, el principal recurso ha sido el estudio del propio código fuente de gvSIG, algo posible gracias a su carácter abierto y gratuito.

El software generado en este TFC se ha elaborado de manera iterativa e incremental, con varias versiones de la extensión hasta llegar a la versión de entrega (**v1.0**). La metodología de análisis, diseño y programación ha sido orientada a objetos (OO) y el análisis de requisitos se basó en casos de uso. Se han aplicado igualmente buenas prácticas como la reutilización de componentes y patrones de diseño (especialmente MVC[1]). También hubo programación de pruebas unitarias y el uso de diversas herramientas complementarias a Eclipse (JUnit, Subversion, Ant, Maven, IzPack…), todas ellas pertenecientes al ecosistema libre en Java.

1.2. Planificación

Como primer paso se realizó una exploración inicial del ámbito del proyecto y, en base a ésta, se elaboró un **Plan de Trabajo**. En dicho plan se

[1] MVC = Modelo/Vista/Controlador. Patrón basado en la separación de responsabilidades en el que la vista se ocupa exclusivamente de la presentación y el controlador de trasladar los inputs del usuario al modelo. Éste último permanece independiente de los otros dos y representa el dominio de trabajo y sus reglas.

estableció la descomposición general del TFC en fases, y éstas a su vez en tareas. De manera complementaria se fijaron los hitos de entrega, el calendario y horario de trabajo, los recursos materiales necesarios y un plan de contingencias para situaciones de riesgo.

Las cuatro fases establecidas en el Plan de Trabajo fueron: Inicio, Elaboración, Construcción y Entrega, cada una de ellas coincidiendo con las cuatro PEC[2]. Y a lo largo de ellas se llevaron a cabo cinco actividades principales:

- *Exploración*: Se realizó la investigación inicial (lectura de foros, pruebas tentativas con software…), que permitió fijar las líneas principales del proyecto y las necesidades de formación a cubrir.
- *Planificación*: Establecida en la fase inicial pero revisada mediante el diagrama de Gantt durante todo el proyecto.
- *Configuración del entorno*: Preparación del software base (IDE Eclipse para Java, gvSIG y 52north, repositorios de Subversion…).
- *Formación*: Antes de que se iniciara propiamente el desarrollo de extWPS, se realizó una formación específica en WPS, la aplicación gvSIG y el desarrollo de extensiones.
- *Desarrollo de software*: Se siguió un enfoque iterativo, finalizando con la entrega de un instalador .jar para extWPS.

La distribución temporal de dichas fases y actividades principales puede verse a continuación (Ilustración 2).

[2] PEC = Prueba de Evaluación Continua, propia del sistema de evaluación de la UOC. Cada una de ellas está asociada con una entrega formal del alumno y una evaluación individual del trabajo por parte del consultor.

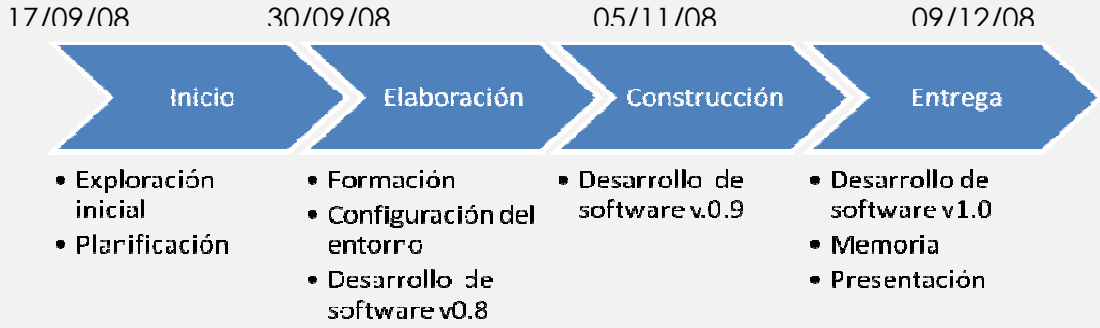

Ilustración 2. Fases y actividades principales del proyecto

Por otro lado, cada una de estas grandes actividades se fue dividiendo en tareas concretas, con una asignación de horas y fechas de inicio/fin, tal y como recoge la Tabla 1.

Tabla 1. Planificación de tareas del TFC

Nombre de tarea	Horas	Inicio	Fin
I. Fase de Inicio: Elaboración del Plan de Trabajo (PEC1)	**31h**	**17/09/08**	**30/09/08**
Lectura Plan Docente + revisión de calendario	3h	17/09/08	17/09/08
Lectura documentación UOC Gestión de Proyectos	3h	18/09/08	18/09/08
Análisis / Investigación inicial sobre Enunciado del Proyecto	5h	20/09/08	21/09/08
Elaboración del Plan de Trabajo	12h	21/09/08	25/09/08
Entrega Borrador del Plan de Trabajo	-	26/09/08	26/09/08
Ampliación / modificaciones del Plan de Trabajo	8h	27/09/08	29/09/08
Entrega PEC1 (Plan de Trabajo)	-	30/09/08	30/09/08
II. Fase de Elaboración (PEC2)	**116h**	**30/09/08**	**04/11/08**
Configuración del entorno de trabajo	**12h**	**30/09/08**	**02/10/08**
Descarga de software e inventario de versiones	1,5h	30/09/08	30/09/08
Espacio de trabajo virtual en *Assembla*	1,5h	30/09/08	30/09/08
Configuración Eclipse / 52 north / gvSIG	6h	01/10/08	02/10/08
Preparación de equipo de respaldo con VirtualBox	3h	02/10/08	02/10/08
Entorno de trabajo configurado 100%	-	02/10/08	02/10/08
Formación tecnológica	**30h**	**04/10/08**	**12/10/08**
Formación en gvSIG Desktop	10h	04/10/08	05/10/08
Formación en desarrollo de extensiones para gvSIG	10h	06/10/08	09/10/08
Formación en WPS (servidor 52north y API Java)	10h	09/10/08	12/10/08
Construcción Software (Ciclo I)	**74h**	**12/10/08**	**04/11/08**

Análisis de requisitos	17h	12/10/08	18/10/08
Diseño de clases e interfaz	16h	18/10/08	22/10/08
Implementación y pruebas	32h	23/10/08	02/11/08
Preparación de entrega	9h	02/11/08	04/11/08
Entrega PEC2 (v. Alfa)	**-**	**04/11/08**	**04/11/08**
III. Fase de Construcción (PEC3)	**102h**	**05/11/08**	**09/12/08**
Construcción Software (Ciclo II)	**92h**	**05/11/08**	**04/12/08**
Análisis de requisitos	11h	05/11/08	08/11/08
Diseño de clases e interfaz	8h	09/11/08	10/11/08
Implementación y pruebas	58h	11/11/08	29/11/08
Documentación de desarrollo	15h	30/11/08	04/12/08
Preparación de entrega	**10h**	**06/12/08**	**08/12/08**
Entrega PEC3 (v. Beta)	**-**	**09/12/08**	**09/12/08**
IV. Fase de Entrega (PEC4)	**113h**	**09/12/08**	**23/01/09**
Construcción Software (Ciclo III)	**14h**	**09/12/08**	**13/12/08**
Últimas revisiones y Cierre de software (v1.0)	14h	09/12/08	13/12/08
Memoria	**38h**	**09/12/08**	**21/12/08**
Lectura documentación UOC Redacción textos científicos	3h	09/12/08	09/12/08
Redacción del documento	35h	10/12/08	21/12/08
Entrega Borrador	**-**	**21/12/08**	**21/12/08**
Correcciones / revisión	**11h**	**21/12/08**	**24/12/08**
Presentación	**22h**	**22/12/08**	**29/12/08**
Lectura documentación UOC Presentaciones	3h	22/12/08	22/12/08
Elaboración de guión de la presentación	1,5h	23/12/08	23/12/08
Recopilación de materiales	1,5h	23/12/08	23/12/08
Construcción mediante PowerPoint + grabación de pantalla	16h	23/12/08	29/12/08
Ajustes / Revisión Final de Entrega	**19h**	**29/12/08**	**05/01/09**
Entrega PEC4 (TFC Final)	**-**	**05/01/09**	**05/01/09**
Debate Virtual Final	**9h**	**21/01/09**	**23/01/09**
PROYECTO TFC	**362h**	**17/09/08**	**23/01/09**

1.3. Entregables

Se han generado los siguientes materiales de entrega dentro del TFC.

A. Memoria de proyecto

El presente documento. Refleja los detalles técnicos del proyecto, siendo el resultado de varias versiones borrador previas.

B. Presentación multimedia

Un fichero en formato .avi con diapositivas y audio que presenta de manera sintética el trabajo realizado, incluyendo una demo del software en funcionamiento.

C. Datos y desarrollos finales v1.0

Dentro de este conjunto se incluyen:

- Instalador .jar de la extensión cliente WPS (extWPS 1.0). Incorpora binarios, documentación y fuentes completos.
- Manual de Instalación v1.0.
- Manual de Usuario v1.0.

Además de estos componentes, que conforman la entrega final del proyecto, hubo varias entregas intermedias durante las distintas PEC: Plan de trabajo (PEC1), informes de avance del proyecto y varios borradores del software y la memoria (PECs 2, 3 y 4).

Todos los entregables pueden obtenerse en:

http://svn.assembla.com/svn/TFC_SIG_vvelarde

2. Contexto tecnológico

En este apartado se proporciona una descripción del marco general en el que se encuadra el proyecto. Parte importante de lo aquí incluido es resultado de la fase de exploración y formación del proyecto.

En primer lugar, se describen los Sistemas de Información Geográfica (**SIG**) y su utilidad en el manejo de información espacial (apartado 2.1). En el apartado 2.2 se explica en qué consiste el protocolo **WPS** del Open Geospatial Consortium (OGC) y el papel de la interoperabilidad dentro del contexto de cambio tecnológico actual.

Finalmente, en el apartado 2.3 se presenta el software **gvSIG**, haciendo especial hincapié en su arquitectura y su modelo de desarrollo. De manera complementaria, a lo largo del apartado se proporcionan algunas referencias bibliográficas de interés para profundizar en el tema.

2.1. Sistemas de Información Geográfica

Existen muchas definiciones sobre los Sistemas de Información Geográfica (SIG[3]), desde ámbitos científicos y técnicos variados y muchas veces con enfoques complementarios (Geografía, Topografía, Informática, Geología...). En general conviven tres acepciones: un SIG como programa informático, un SIG como un tipo de Sistema de Información y, finalmente, los SIG como sector o disciplina científica especializada en la información geográfica.

Una breve pero relevante definición es: "*Un SIG es un sistema de información de base informática que permite la captura, modelización, manipulación, recuperación, análisis y presentación de datos referenciados geográficamente*" (WORBOYS F., 2001).

Precisamente en el último aspecto, los datos *georreferenciados*, es donde está el factor de mayor peso y originalidad de estos sistemas. En esencia, un **dato georreferenciado** posee una referencia espacial directa o indirecta sobre la Tierra, lo cual permite al sistema representarlo sobre un mapa o globo y tener en cuenta su posición para realizar consultas, entre otras muchas operaciones.

[3] Existen otros términos para referirse a estas tecnologías, como p.ej. Sistemas de Información Territorial (SIT). No obstante **SIG** y su equivalente anglosajón **GIS** (*Geographic Information Systems*) son los más extendidos; en este escrito se utilizará la variante española.

La referencia espacial más habitual se consigue mediante la asignación a una entidad concreta, por ejemplo un núcleo de población, una carretera o un polígono industrial, de coordenadas geográficas o proyectadas. Su forma se abstrae como un punto, una línea o un polígono, de acuerdo a la escala de interés. Esto es lo que se conoce como *modelo vectorial*.

Otra opción frecuentemente usada en los SIG es el *modelo raster*. En este caso, el acento se traslada de la entidad, cuya forma no se refleja expresamente, hacia el valor del dato en el espacio de trabajo, el cual es dividido en una malla de teselas regulares. Cada una de estas teselas o píxeles tiene asociado un valor de fenómeno (de manera indirecta, los valores podrán servir para inferir, con una precisión dependiente de la resolución, áreas y distribuciones). En la siguiente ilustración se muestra la esencia del modelo vectorial (arriba) y el raster (abajo) con polígonos, puntos y líneas.

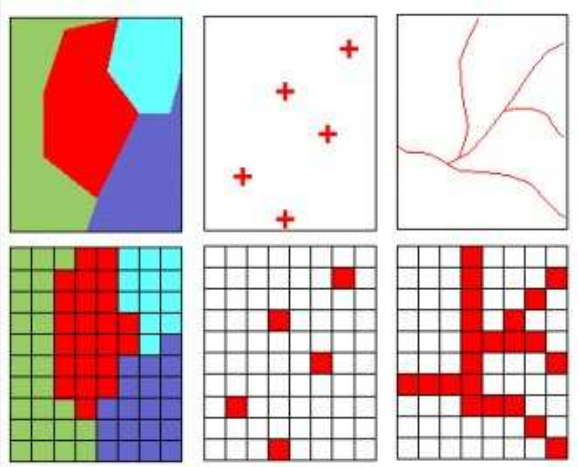

Ilustración 3. Modelos vectorial y raster en un SIG

En general, hay una mejor adecuación del modelo raster al tratamiento de datos medioambientales, con límites "difusos" y valores continuos (p.ej. temperatura, altitud del terreno, concentración de metales…). Y el modelo vectorial define mejor entidades concretas y con límites tangibles, construidas por el hombre, como p.ej. carreteras, fronteras de países y regiones, etc.

En cualquier caso, las funcionalidades conseguidas con los SIG sirven para responder preguntas de índole espacial, tales como las que recoge la Tabla 2.

Tabla 2. Preguntas básicas para resolver con SIG.

Ámbito	Pregunta	Ejemplo
1. Localización	*¿Qué hay en…?*	¿Cuánta población en edad escolar hay en una sección censal?
2. Condición	*¿Dónde sucede que…?*	¿Dónde hay un lago para pescar a menos de 50km de nuestra casa y rodeado de bosques?
3. Tendencias	*¿Qué ha cambiado…?*	¿Cuántas hectáreas de bosque quemado en zonas rurales hay en una comarca y cuál es la diferencia con la situación hace 20 años?
4. Rutas	*¿Cuál es el camino óptimo…?*	¿Cómo llegar desde el lugar de un accidente al hospital más cercano?
5. Pautas	*¿Qué pautas existen…?*	¿Qué patrón de distribución existe de casos de cáncer alrededor de una central nuclear?
6. Modelos	*¿Qué ocurriría si…?*	¿Qué efecto se produciría en la costa de una región si el nivel del mar creciera 2 metros?

Fuente: GUTIÉRREZ PUEBLA, J. y GOULD, M. (1994)

Las soluciones SIG más difundidas a nivel internacional son las de las casas comerciales ESRI, INTERGRAPH y BENTLEY, que acaparan en torno al 50% de las ventas en la industria, con productos como ArcGIS, Geomedia y Microstation (Informe Daratech[4], 2006). Además existen otras muchas soluciones que amplían la oferta, tanto propietarias (p.ej MapInfo) como libres (GRASS, Jump, uDig, el propio gvSIG…).

Y todo este software se presenta de muy diversas maneras al usuario y al programador: productos para escritorio, para dispositivos móviles, para servidores… en forma de paquetes cerrados o personalizables, componentes de desarrollo, APIs a servicios remotos, etc.

[4] El informe de Daratech *GIS/Geospatial Markets & Opportunities* se elabora anualmente. Sólo está disponible de manera íntegra bajo subscripción anual de pago, pero algunas de sus líneas fundamentales son publicadas en notas abiertas de prensa como: http://www.directionsmag.com/article.php?article_id=403&trv=1 (año 2003) o http://www.gisuser.com/content/view/9446/ y http://www.gisuser.com/content/view/9492/ (2006).

En cualquiera de sus formas, es evidente que los SIG son una tecnología informática emergente, y que tienen una enorme utilidad en ámbitos como medioambiente, urbanismo, catastro, la gestión de redes, emergencias, geomarketing, etc.

2.2. Interoperabilidad y Open Geospatial Consortium

En los últimos años se está produciendo una evolución muy notable de los SIG y las tecnologías geoespaciales en general. Antes eran usadas solo por un reducido grupo de científicos y profesionales de la gestión del territorio, pero ahora su uso es mucho más generalizado y abierto. En gran medida se han visto impulsadas por Internet, la eclosión de la llamada Web 2.0 e iniciativas rompedoras como las de Google[5].

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

Para permitir este nuevo ecosistema, crecientemente variado y complejo, es imprescindible trabajar en la interoperabilidad, y es ahí donde el *Open Geospatial Consortium* (OGC) cumple su papel. El OGC es un consorcio internacional participado por empresas, universidades y organismos gubernamentales que lidera la generación de estándares en el área geoespacial. Dentro de estos estándares destacan los **OGC Web Services (OWS)**, un conjunto de servicios web definidos utilizando estándares de Internet no propietarios, particularmente *World Wide Web* (WWW), *HTTP*, *Uniform Resource Locators* (URLs), tipos *Multipurpose Internet Mail Extensions* (MIME) y el lenguaje XML. A continuación se incluye un esquema con los OWS más extendidos y sus operaciones principales (Ilustración 4).

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

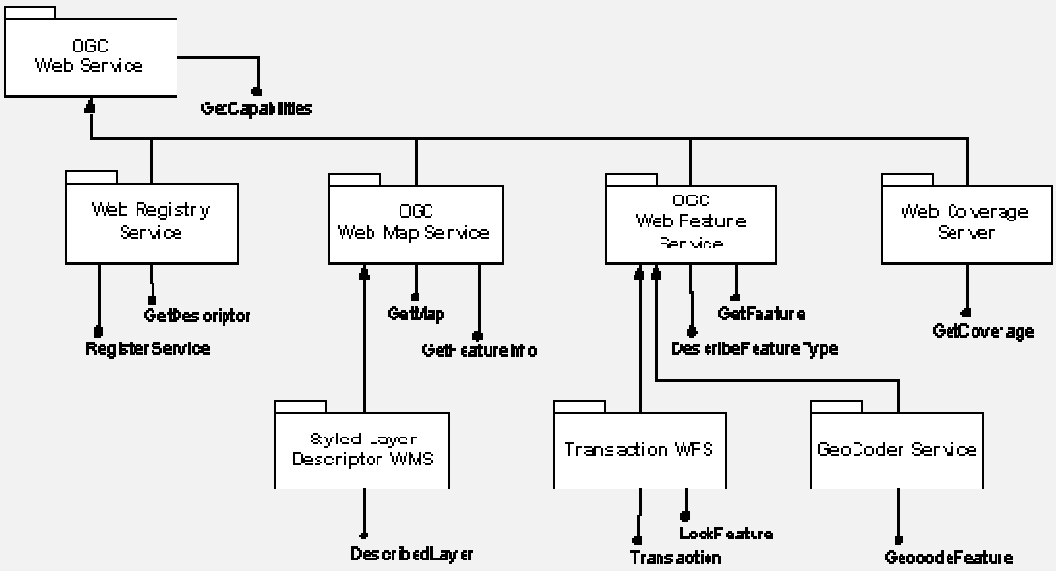

Ilustración 4. Servicios OWS principales

En este área se han logrado ya importantes avances como el **WMS** (*Web Map Service*) o **WFS** (*Web Feature Service*), que han favorecido enormemente la publicación de cartografía y datos a través de Internet por parte de organismos productores. Y recientemente ha sido propuesto como estándar **WPS** (*Web Processing Service*), apareciendo así un protocolo para publicar y consumir geoprocesos remotos que puede suponer un paso fundamental en el avance de los SIG[6].

2.3. Web Processing Service y 52north

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

En dicho documento se señala cómo WPS es un interfaz estandarizado de servicio para la publicación de **procesos geoespaciales** y su descubrimiento y consumo por parte de clientes. Dentro de la categoría de procesos geoespaciales se incluye cualquier *algoritmo, cálculo o modelo que opere sobre*

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

datos con referencia espacial, desde operaciones simples como p.ej. un buffer hasta modelo complejos como cálculo de zonas de inundabilidad (OGC, 2007).

El interfaz de un servicio conforme a WPS debe presentar obligatoriamente tres operaciones, mostradas en la Ilustración 5.

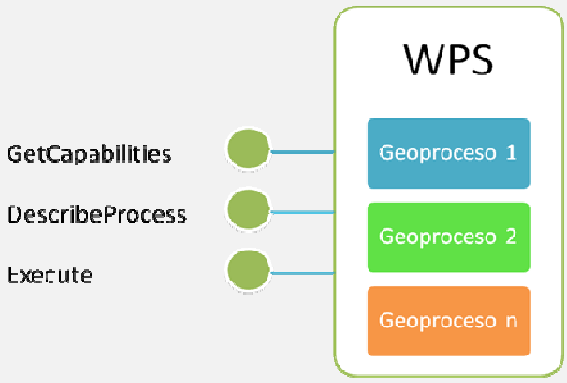

Ilustración 5. Operaciones del interfaz WPS

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

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

WPS puede funcionar de manera integrada con **otros servicios OGC**, y así tomar p.ej. como capa de entrada una entidad servida vía WFS, o encadenar varias operaciones alojadas en otros servidores WPS, ofreciendo una fachada

más sencilla ante el cliente. También pueden formar parte de flujos de negocio complejos, orquestados junto con otros servicios mediante BPEL.

Desde un punto de vista de *implementaciones de WPS* existen varias iniciativas, destacando PyWPS, deegree y 52north[7]. Mientras que la primera se basa en el lenguaje Python, las otras dos son marcos de trabajo para Java.

En la iniciativa **52north** es donde se han encontrado los recursos de mayor interés y dentro de ella en su *Geoprocessing Community*[8]. La comunidad de geoprocesos es un espacio donde pueden encontrarse abundantes recursos para WPS, tales como un repositorio de Subversion con fuentes Java para servidor y cliente, un Wiki con presentaciones y publicaciones sobre WPS, tutoriales, demos… y una lista de correo para dar soporte a los desarrolladores (https://52north.org/mailman/listinfo/geoprocessingservices). Liderando esta lista y el trabajo con el código están fundamentalmente dos personas: Bastian Schäffer (schaeffer@52north.or) y Theodor Foerster (foerster @52north.or).

Un primer punto de contacto recomendable con 52north es la lectura de los documentos técnicos del Wiki, para a continuación realizar pruebas de ejecución con el servidor WPS publicado y un cliente WPS (como los existentes para uDig o Jump). En el caso de *Jump*, el cliente WPS es una opción específica que aparece como un menú nuevo en la ventana principal; con *uDig*, se integra como un origen de datos más, dentro de la opción *Add Data* (Ilustración 6).

---

[7] PyWPS: http://pywps.wald.intevation.org/ | degree: http://deegree.org/ | 52north: http://52north.org/

[8] 52north se organiza en grupos de interés denominados *Communities*. Existen 4 comunidades centrales: *Security & Geo-RM*, *Sensor Web*, *ILWIS* y *Geoprocessing*, cada una de ellas con su espacio propio. El modelo de licencia del software de 52north es dual. Para la parte de librerías central (CORE), la utilizada directamente en este proyecto, existe la opción GPL y también un modelo de licencia comercial. Para la parte de proyectos complementarios, conocida como INCUBATOR, aplica solo GPL.

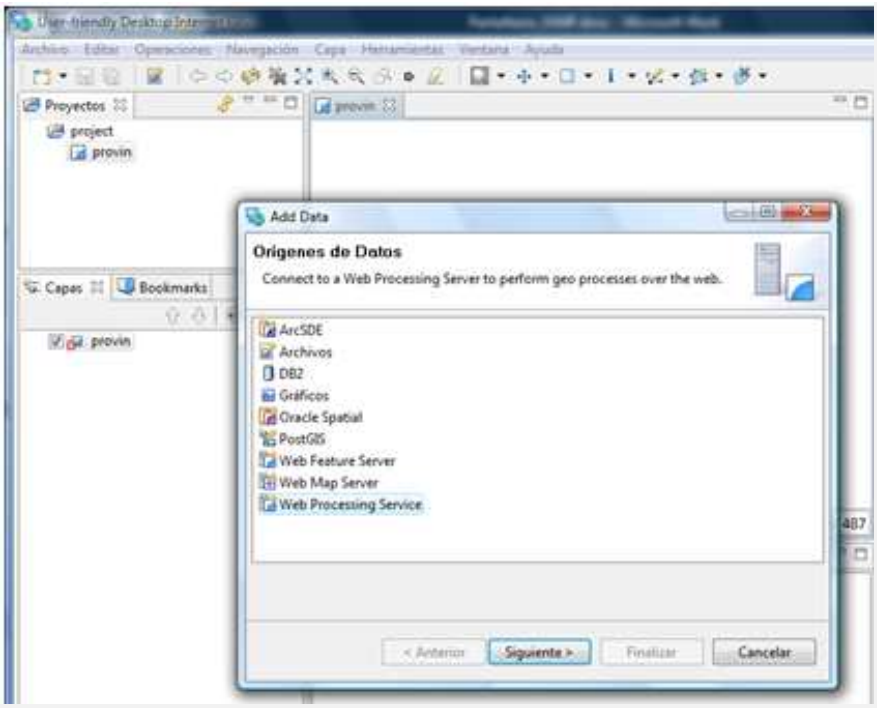

Ilustración 6. Cliente WPS en uDig (52north)

El componente servidor WPS de 52north se despliega de manera sencilla sobre un *Apache Tomcat*, proporcionando un entorno local para pruebas idóneo: un servicio WPS 1.0 con geoprocesos simples (buffer, eliminación de pseudonodos, simplificación mediante Douglas-Peucker…). Para la parte cliente es preciso instalar primero el software de escritorio que usemos como base (p.ej. uDig) y luego instalar el plugin WPS correspondiente.

Un punto muy favorable sobre 52north es que todos los fuentes están disponibles de manera abierta en su repositorio de Subversion (http://52north.org/svn/geoprocessing/). Con Eclipse + los plugins para Maven se puede descargar, estudiar y compilar el código completo para **WPS de 52north**.

Los componentes principales del proyecto WPS de 52north son:

- *52n-wps-server*.- componente servidor
- *52n-wps-io*.- manipulación de parámetros de entrada / salida de los geoprocesos.
- *52n-wps-commons*.- librería para cuestiones de configuración.

- ***52n-wps-client-lib***.- librería para la parte de cliente WPS.

En este proyecto se ha tomado como referente el tag más reciente del proyecto en el momento del desarrollo, la **versión 1.0-RC2**, y se ha trabajado sobre todo con *52n-wps-client-lib*. Con esta versión se procedió a la compilación en local y así obtener las librerías que fueron luego las utilizadas en el desarrollo de la extensión.

Es destacable que el proyecto 52north utiliza algunas librerías geoespaciales de Java muy extendidas, especialmente *Geotools* y *Java Topology Tools*, tal y como se refleja en sus documentos de dependencias (ficheros pom.xml del sistema *Maven*).

2.4. gvSIG y su carácter extensible

GvSIG es un software surgido como iniciativa de la Consejería de Infraestructuras y Transporte de la Generalitat Valenciana (Comunidad Valenciana, España). Está orientado principalmente a un perfil de usuario acostumbrado a manejar información geográfica (como p.ej. funcionarios de la administración regional, profesionales de la ordenación del territorio…) pero presenta un interfaz bastante amigable y sencillo.

Se trata de un software **gratuito** que permite gestionar y analizar datos espaciales, y además existen traducciones para varios idiomas. Su posición se está fortaleciendo notablemente frente a otros SIG de escritorio, especialmente software propietario, y de hecho han comenzado iniciativas del proyecto en otros entornos como los dispositivos móviles (gvSIG Mobile).

En la actualidad gvSIG dispone de varias extensiones (Sextante, análisis de redes, 3D…) y se caracteriza por manejar los formatos más usuales de datos vectoriales y raster (p.ej. ficheros shapefile, bases de datos PostGIS…).

Además gvSIG cumple con un número destacado de estándares OGC como WMS, WFS o WCS, lo cual lo convierten en un cliente adecuado para IDEs (Infraestructuras de Datos Espaciales); esto puede observarse en los orígenes de datos que admite para añadir como capa (Ilustración 7).

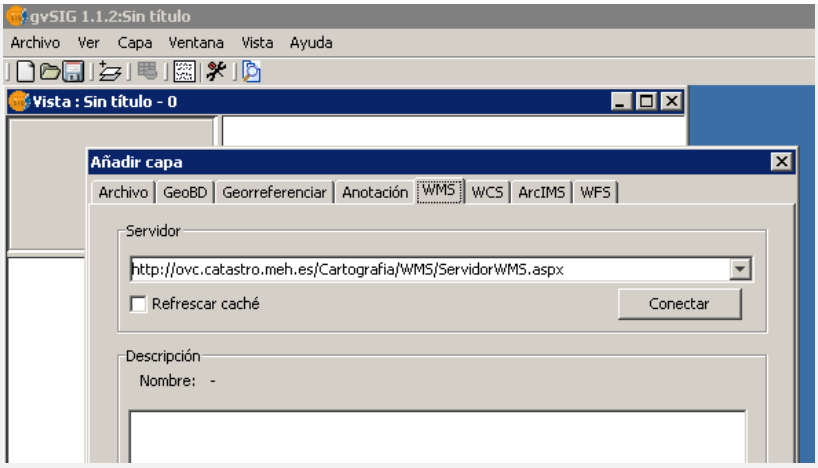

Ilustración 7. Capacidades OGC de gvSIG v1.1.2

Tanto para usar el sistema como para programar una extensión para él, los cursos básicos del software disponibles en Internet suponen una buena ayuda. Además del manual oficial de usuario, se recomienda el curso online CAMARGO LUQUE, F.R (2007)[9] de *Introducción a gvSIG*.

Desde la perspectiva de desarrollador, gvSIG puede verse como un software SIG libre construido en **Java**, cuyos aspectos principales son:

- Código con licencia **GPL** y disponible vía Subversion[10].
- **Comunidad** de desarrolladores creciente[11].
- Extensible mediante **plugins**, gracias a su *framework* de base (Andami).

Andami es la base de todo el sistema gráfico de gvSIG: marco general MDI, ventanas, menús, botones, barras de estado… y está montado sobre Java Swing. Andami es también el responsable de coordinar y cargar los *plugins y extensiones*, que son los que van proporcionando las funcionalidades al sistema de manera modular. Y para facilitar tareas recurrentes, proporciona toda una

[9] Disponible en http://downloads.gvsig.org/pub/images/Course-gvSIG-1_0-and-1_1-F_Camargo-es.zip

[10] http://subversion.gvsig.org/gvSIG/

[11] Es especialmente destacable la lista de desarrolladores: http://runas.cap.gva.es/mailman/listinfo/gvsig_desarrolladores

serie de servicios horizontales a los plugins (log, traducciones mediante ficheros de texto plano, acceso a recursos como ficheros xml, etc.)

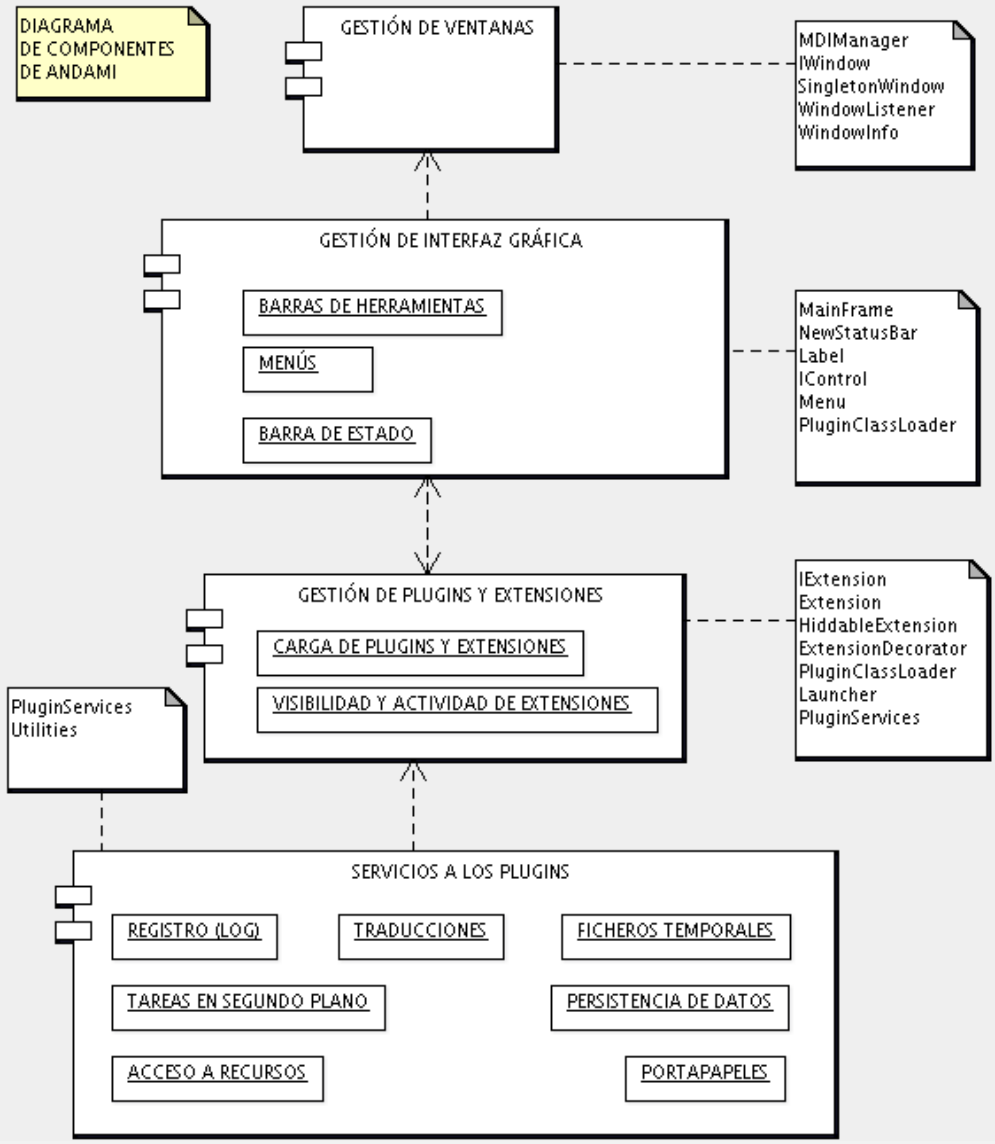

Ilustración 8. Framework base de gvSIG

El propio gvSIG no es sino una aplicación apoyada sobre Andami, evidentemente la más importante y la que proporciona las funcionalidades "geo". Sobre esta base se han ido desarrollando las extensiones, que para un buen funcionamiento deben seguir unas pautas preestablecidas.

Para conocer cuáles son esas pautas de desarrollo hay varios recursos online de interés. De ellos, los más reseñables son:

- La documentación y los tutoriales de programación de la página principal de desarrollo de gvSIG:

  https://gvsig.org/web/docdev/

- Los videos del *Seminario técnico de desarrollo con gvSIG*, impartido por Jorge Gaspar Sanz Salinas en la Universidad Politécnica de Valencia, disponibles en:

  http://cartosig.upv.es/es/recursos-sig/seminario-tecnico-de-desarrollo-en-gvsig

Fruto de este estudio, se llegó a la consecución de 2 hitos importantes para iniciar cualquier desarrollo de una extensión sobre gvSIG:

1. Descargar de Subversion y compilar correctamente en Eclipse los fuentes de gvSIG (en nuestro caso se utilizó la versión 1.1.2).
2. Desarrollar correctamente una extensión demo, llamada *extWorkshop*. Esto permite comprender los mecanismos de compilación con Ant y la configuración de extensiones para gvSIG.

3. Desarrollo de la extensión

Tras la fase inicial del proyecto y especialmente al comienzo de la fase de elaboración, se fue configurando el entorno de desarrollo. Una vez hecho esto, se gestó una versión de la extensión para WPS (**extWPS**) en cada PEC. Cada versión implicó tareas de análisis de requisitos, diseño de clases e interfaz, programación y pruebas.

El fruto de ambas actividades, configuración de entorno y desarrollo, es lo que se recoge en los siguientes apartados.

3.1. Configuración del entorno

Para lograr el desarrollo de la extensión extWPS se configuró un entorno que respondiera a las distintas necesidades del proyecto, seleccionando para cada una de ellas aquellas herramientas y versiones consideradas más adecuadas. La selección se refleja en la Tabla 3.

Tabla 3. Herramientas del entorno de trabajo

Ámbito	*Herramienta/s*
Planificación y control de tareas	Microsoft Project 2007
Software SIG	gvSIG Desktop v1.1.2 / JUMP + cliente WPS de 52north / uDig v1.1.1 + cliente WPS de 52 north
Framework Java	J2SE Development Kit (v5.0 Update 16), con las librerías gráficas JAI v1.1.3 y JAI ImageIO v1.1
IDE para programación en Java	Eclipse para Java (v3.4.0 Ganymede) con los plugins: VisualEditor, Subclipse, UML2 y Maven.
Repositorio de código fuente	Subversion (en la forja *Assembla*), con cliente TortoiseSVN v1.5.3
Servidor WPS para pruebas	Servicio WPS de 52north v1.0-RC2, desplegado sobre un Apache Tomcat v5.5.27
Código fuente de gvSIG	gvSIG v1.1.2 src
Librerías WPS cliente	Librería cliente WPS de 52north v1.0-RC2, recompilada en local
Herramientas para labores ofimáticas (documentos,	Microsoft Word / Project 2007. Camtasia Studio v5.0

presentaciones…)

De todos los elementos del entorno de trabajo hay uno que resultó especialmente útil: Assembla. **Assembla** es un espacio de trabajo online que integra herramientas para la gestión, colaboración y programación ágil. Las herramientas disponibles son Subversion/Git, Mercurial y Trac; además tiene espacio para Wiki, hitos, etc. Hasta 200 Mb es un espacio gratuito.

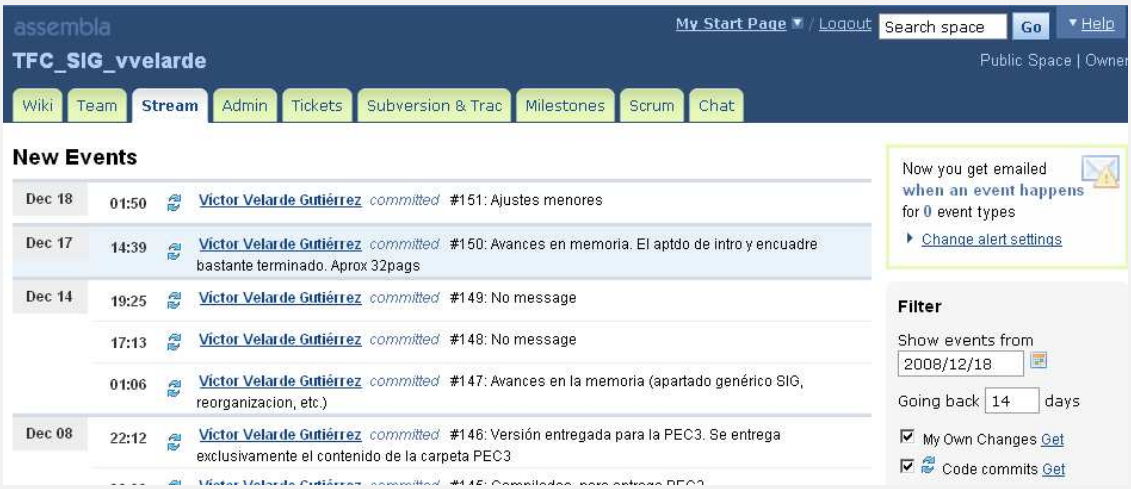

Ilustración 9. Espacio de trabajo Assembla

Como el TFC ha sido un trabajo individual, no se ha usado la parte de grupo, pero sí intensamente el repositorio de Subversion, disponible en la Url:

http://svn.assembla.com/svn/TFC_SIG_vvelarde

3.2. Análisis de requisitos

Una vez configurado el espacio de trabajo, se inició el desarrollo del software, comenzando por la construcción del catálogo de requisitos generales. Los recopilados fueron:

1. El sistema será una extensión integrada en gvSIG (Desktop).

2. El sistema será capaz de consumir servicios Web ofrecidos mediante WPS en su versión 1.0.0.
3. El sistema permitirá aplicar geoprocesos WPS a datos de gvSIG (usar las capas cargadas como parámetros).
4. El sistema permitirá mostrar los resultados de las ejecuciones de procesos en el propio gvSIG (carga de los resultados como capa).
5. La funcionalidad WPS se integrará adecuadamente con otras funcionalidades gvSIG para consumo de servicios OGC.

Para estructurar los requisitos y favorecer su análisis, se eligió la técnica de los **casos de uso**. Como resultado del trabajo en esta línea, se identificaron tres casos de uso (Ilustración 10).

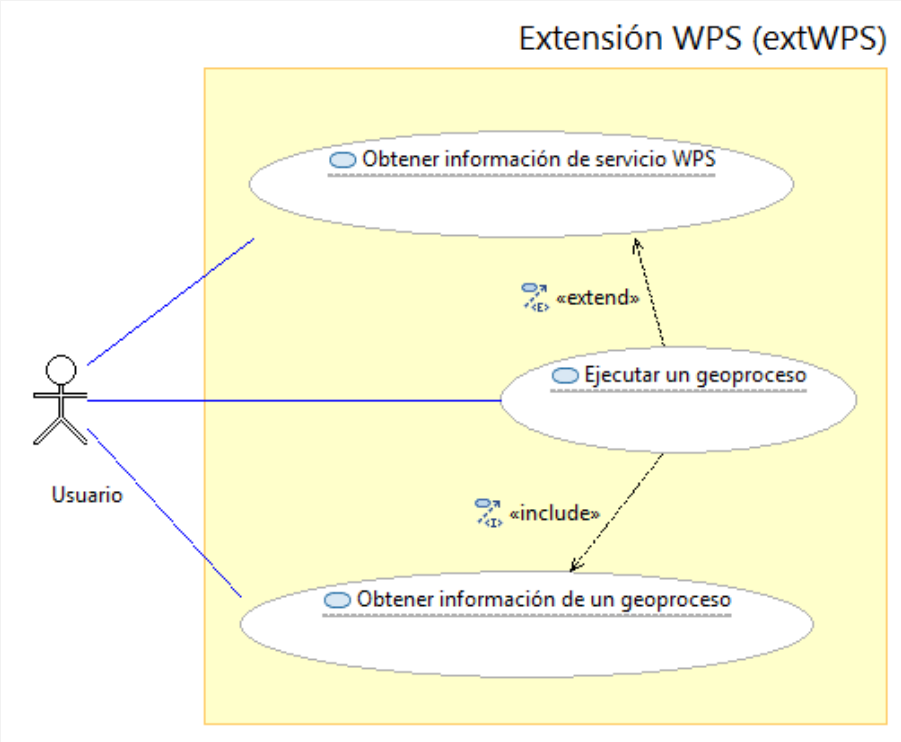

Ilustración 10. Casos de uso de extWPS

Estos casos de uso tienen una correspondencia uno a uno con las **tres operaciones básicas** del interfaz de un servicio WPS. En éste sentido, el disponer

tanto de una especificación preestablecida y estandarizada (OGC) como de ejemplos de implementación (Jump, uDig…) fue una ventaja muy importante.

A continuación incluimos la descripción detallada de los casos de uso (Tablas 4 a 6).

Tabla 4. Caso de uso 01 – Obtener información de un servicio WPS

Objetivos

- Conocer que procesos se pueden ejecutar desde un servicio WPS.
- Saber si un servicio WPS está disponible.

Requisitos previos

- El usuario conoce la url del servicio WPS al que desea conectar.

Pasos

1. El usuario inicia un asistente para la conexión con el servicio WPS.
2. El sistema presenta en la pantalla una caja de texto para indicar la dirección del servicio.
3. El usuario teclea la url del servicio al que desea acceder y pulsa el botón conectar.

    3.1. La url introducida es incorrecta (vacía o no válida formalmente).

    3.2. El sistema notifica el error con un mensaje por pantalla.

    3.3. El usuario la corrige modificando el contenido de la caja de texto.

4. Validado el formato de la url, el sistema realiza una petición de la información sobre el servicio WPS (*GetCapabilities*)

    4.1. La dirección suministrada no responde a la petición de manera correcta (la url no representa un servicio WPS, hay un error de red, *timeout*…)

    4.2. El sistema notifica el error en la petición con un mensaje explicativo por pantalla.

4.3. El usuario corrige si está en su mano el error y vuelve al paso 3.

5. El sistema recibe la información del servicio WPS y la presenta por pantalla, mostrando los datos generales del servicio y una tabla con los geoprocesos disponibles (nombre, descripción y versión).

Tabla 5. Caso de uso 02 – Obtener información de un geoproceso

Objetivos

- Conocer los detalles sobre un geoproceso disponible en un servicio WPS.
- Permitir en un momento posterior proceder a la ejecución del geoproceso.

Requisitos previos

- Se ha ejecutado previamente el caso de uso "01 - Obtener información de un servicio WPS" y se ha establecido la conexión correcta con el servidor.

Pasos

1. Dentro del asistente, el usuario selecciona un geoproceso de los disponibles en el servidor y pulsa siguiente.

2. El sistema realiza una petición al servicio WPS para obtener los detalles del geoproceso (*DescribeProcess*)

   2.1. El servidor no está disponible en ese momento.

   2.2. El sistema notifica el error en la petición con un mensaje explicativo por pantalla.

3. El servicio responde adecuadamente a la petición y se muestran por pantalla los datos concretos del geoproceso, incluyendo su nombre, descripción y parámetros de entrada.

4. En cuanto a los parámetros, serán de tipo simple (datos alfanuméricos, como p.ej. un valor decimal para indicar una distancia) o tipo complejo (capas gráficas) y se mostrarán los controles adecuados para rellenarlos (cajas de texto y listas desplegables, respectivamente).

5. En el caso de parámetros complejos, en los combos aparecerán las capas vectoriales cargadas en ese momento en gvSIG.

Tabla 6. Caso de uso 03 – Ejecutar un geoproceso

Objetivos

- Ejecutar un geoproceso disponible en un servicio WPS.

Requisitos previos

- Se ha ejecutado previamente el caso de uso "02 - Obtener información de un geoproceso".

Pasos

1. El usuario rellena los parámetros de entrada necesarios para la ejecución.
2. El sistema realiza la petición de ejecución al servicio WPS (*Execute*).
3. El sistema muestra por pantalla el resultado de la ejecución del geoproceso.
4. En el caso de resultado correcto, se informará mediante un mensaje en pantalla.
5. Tras finalizar el asistente, se cargará la capa resultado en la vista de mapa actual.

3.3. Diseño

De cara al **diseño del interfaz**, además de tener en cuenta la propia orientación de los casos de uso, se realizó un estudio de dos ámbitos:

- los servicios OWS ya existentes en gvSIG (WMS y WFS).
- otras aplicaciones SIG desktop con clientes WPS (fundamentalmente uDig).

Con ello se buscaba un diseño sencillo, "familiar" para el usuario, y una buena usabilidad, aprendiendo de las experiencias de otros desarrolladores.

El interfaz considerado más adecuado para cumplir con los casos de uso fue una pantalla paso a paso o *wizard* junto a las de otros orígenes de datos OGC (WMS, WFS y WCS). Es el apartado de gvSIG denominado '**Añadir capa**'.

Otra opción barajada posteriormente, pero descartada por falta de tiempo, fue la integración de la extensión en el *Gestor de Geoprocesos*, acercando más la extensión a su esencia, los geoprocesos, que a su forma (servicios OWS). De hecho, ambas podrían convivir y complementarse: en el asistente de añadir capa se explora el servidor, se ven sus geoprocesos y al

ejecutar uno con éxito este se podría añadir a "favoritos" e incorporarse en el Gestor (así las próximas ejecuciones serían más directas). Esta es una línea del TFC que queda abierta.

La mecánica para construir los interfaces de usuario fue:

1) el diseño de bocetos en papel.
2) la generación visual en Eclipse con el plugin Visual Editor y, finalmente,
3) en la fase de implementación, la reconstrucción y ajuste del diseño por código, control a control.

El resultado del diseño fueron dos pantallas (Ilustraciones 11 y 12).

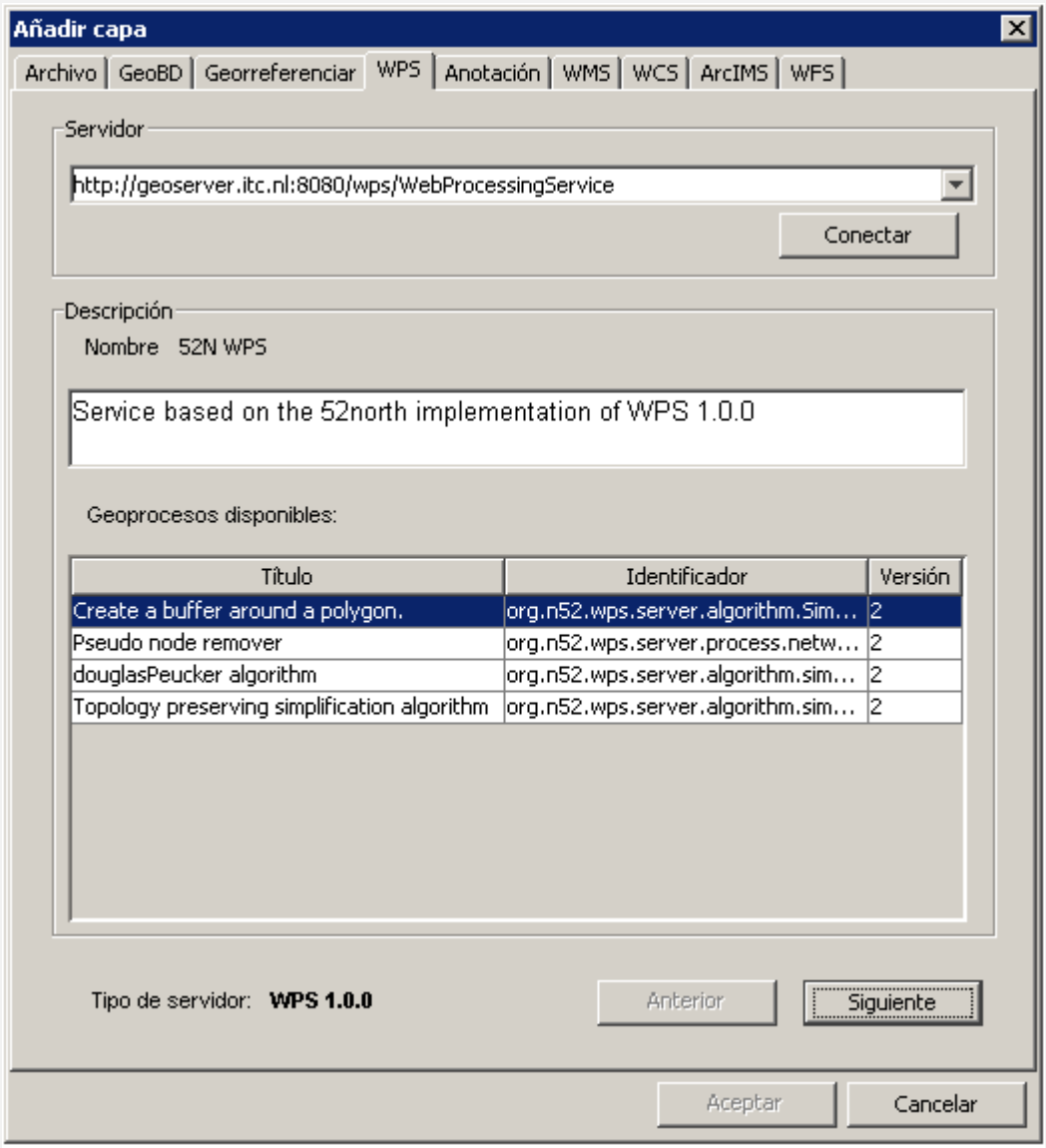

Ilustración 11. Pantalla 1ª del asistente "Añadir Capa" WPS en gvSIG (Servicio)

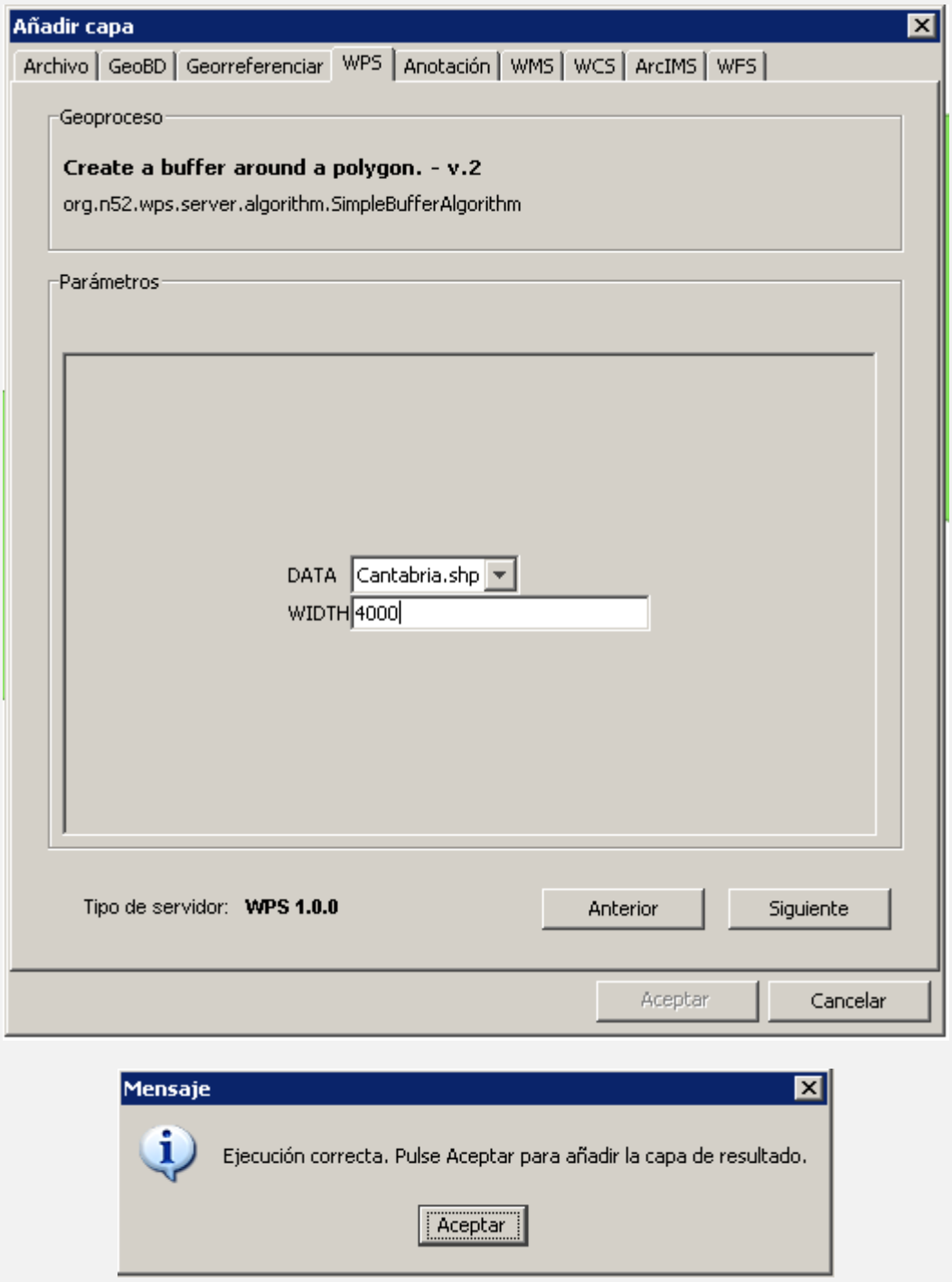

Ilustración 12. Pantalla 2ª del asistente "Añadir Capa" WPS en gvSIG (Geoproceso)

Por otro lado, en cuanto al **diseño de paquetes y clases**, tomando como punto de partida los requisitos funcionales y de interfaz, el patrón MVC y varias iteraciones sobre el código, se llegó al esquema siguiente (simplificado):

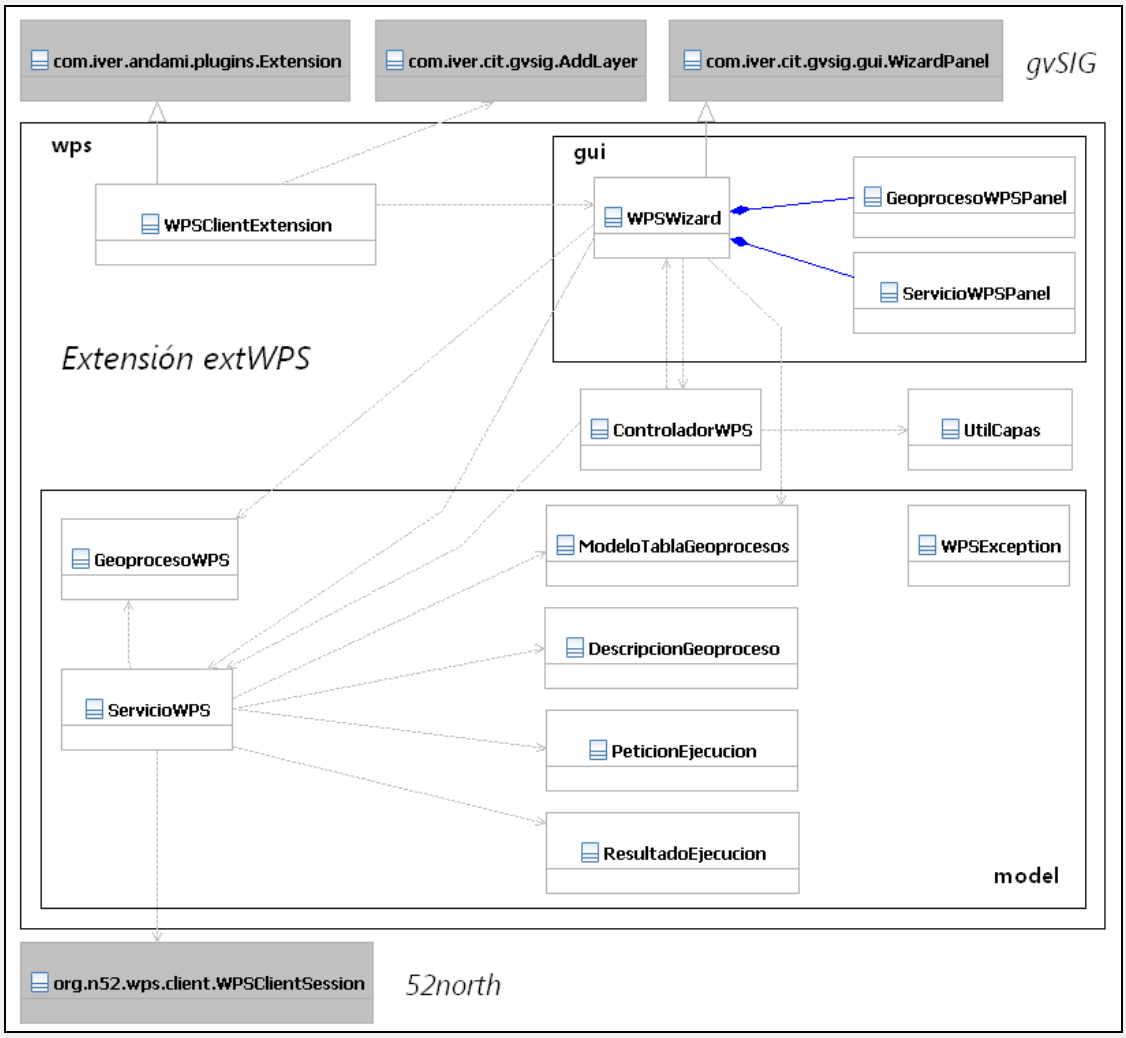

Ilustración 13. Diagrama general de paquetes y clases de extWPS

De manera general, sobre el diseño se puede comentar que refleja claramente cómo la extensión extWPS es fundamentalmente un *trabajo de integración* que permite a gvSIG utilizar las librerías de **52north**, y que está estructurada en 3 paquetes:

a) *edu.uoc.gvsig.wps*: contiene el punto de entrada a la extensión y el controlador (MVC)

b) *edu.uoc.gvsig.gui*: con las clases gráficas.

c) *edu.uoc.gvsig.model*: con las clases de negocio.

La secuencia principal de trabajo dentro de la extensión es la descrita a continuación. ***WPSClientExtension*** es el punto de entrada dentro de gvSIG (heredando de la clase de Andami *com.iver.Andami.plugins.Extension*). En ella se usa la clase *AddLayer* para que se pueda crear la nueva pestaña WPS en el asistente 'Añadir Capa' y conseguir instanciar el interfaz de extWPS.

Con la inicialización de la clase gráfica *WPSWizard* se pone en marcha un mecanismo de intercambio de mensajes con un patrón Modelo-Vista-Controlador, para regular las operaciones definidas en los casos de uso.

Por su parte, el modelo utiliza las librerías cliente de 52north para la realización de sus responsabilidades. La clase ***WPSClientSession*** es el punto de enlace principal de extWPS con las librerías (es una clase *singleton* que actúa como fachada del cliente WPS). En lugar de utilizar directamente las clases de 52north, se recubrieron las más importantes vía composición, con el objetivo de simplificar su interfaz de uso (p.ej. *PeticionEjecucion* utiliza *ExecuteDocument* + *ExecuteRequestBuilder*).

Por simplicidad del diagrama se han obviado otras referencias de la extensión extWPS a elementos de gvSIG, pero cabe destacar que:

- cuando se ejecuta el geoproceso, los resultados se recogen en una capa vectorial de gvSIG (com.iver.cit.gvsig.fmap.layers.FLyrVect)[12].
- se utiliza *Plugin.Services* (servicios generales de Andami) para el log, la traducción de cadenas de texto y la persistencia de la lista de servidores WPS en xml.

En todo caso, consideramos que la mejor manera de comprender el diseño de extWPS es acceder al código fuente, analizarlo y ejecutar los tests. También puede resultar ilustrativo realizar alguna ejecución en *debug* (los fuentes se distribuyen en forma de workspace de Eclipse, así que este proceso es directo).

[12] Si bien la especificación WPS contempla datos en forma raster y vectorial, con 52north el apartado raster aún no está maduro en sus librerías en estos momentos, ni se conocen servicios WPS públicos que acepten y/o devuelvan datos en este formato; por todo ello, la extensión extWPS solo trabaja con datos vectoriales, tanto para entrada como para salida de geoprocesos.

3.4. Codificación

Realizada completamente sobre el IDE Eclipse. El hecho de que gvSIG esté desarrollado en esta plataforma ayuda mucho en el ciclo de codificación, compilación y despliegue. Los fuentes de gvSIG en su versión 1.1.2 se distribuyen de hecho como un *workspace* de Eclipse.

Durante la programación se procuró mantener en todo momento el código lo más claro y ordenado posible, practicando *refactoring*, y se fueron generando las anotaciones para los *Javadocs*.

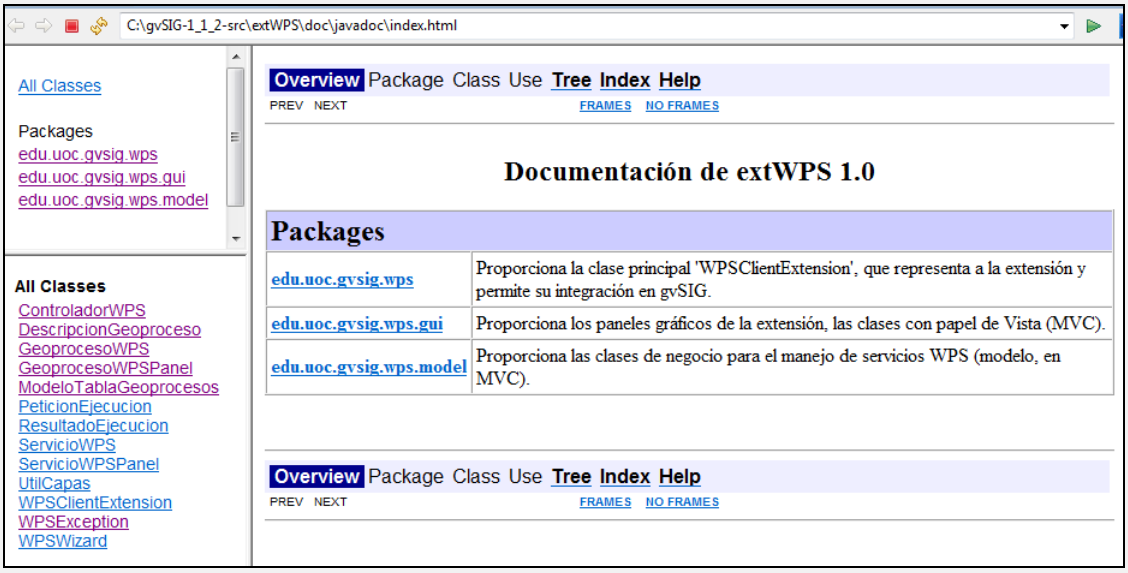

Ilustración 14. Javadocs de extWPS

Además de la codificación en Java del diseño antes indicado, ha habido un tiempo importante para cuestiones complementarias , como ir perfeccionado el fichero de **Ant** *build.xml*, usado para las tareas más usuales de compilación y despliegue, y conocer aspectos como el funcionamiento de Maven.

Gracias al plugin *Subclipse* y el espacio online para Subversion en *Assembla*, la programación se pudo realizar desde distintos equipos de trabajo, de manera fluida y segura, y realizando el etiquetado de fuentes para cada versión (tags).

Incompatibilidad de librerías base entre gvSIG y 52north

Con respecto al uso de 52north es importante señalar que existe una incompatibilidad a nivel de librerías base (*Geotools*) con las versiones utilizadas de ambos componentes: **52north v1.0-rc2** y **gvSIG v1.1.2**, detectada durante la codificación, y que se han tomado una serie de decisiones, en forma de *fork de las librerías de 52north* para resolver este aspecto. A continuación se explica en detalle el problema y la solución adoptada.

52north integra en ese punto las librerías .jar de *Geotools 2.4.2* y *JTS 1.8*, mientras gvSIG utiliza *Geotools 2.1* y *JTS 1.7*. Este es un aspecto importante, puesto que el método para hacer llegar los parámetros gráficos (las geometrías) al geoproceso en el servidor depende de las mismas.

El error se manifiesta al utilizar el método básico de n52 denominado *ExecuteRequestBuilder.addComplexData*, que recibe dos parámetros: una cadena de texto y un *object*. Dicho *Object* es convertido internamente a una ***FeatureCollection*** (fc) de la librería Geotools, como se puede observar en sus fuentes. Luego n52 recoge esa fc y la recorre, para generar a partir de ella el GML que luego hará llegar al geoproceso de manera conveniente.

Si se mantienen las versiones de las librerías indicadas, y se intenta hacer llegar una fc generada en gvSIG a n52, se produce una *InvalidClassCastException* por la incompatibilidad de la 2.1 respecto a la 2.4. Para solucionar esto, se intentó incorporar los jar de la versión 2.4 a la extensión en desarrollo, pero entonces se producía una incompatibilidad general en gvSIG, porque no se pueden mantener dos versiones diferentes de la misma librería.

Después de comentar el problema en los foros de desarrollo de gvSIG, se recibió la recomendación de generar un *fork* igualando las versiones de la librería en ambas partes. Por ello, partiendo de los fuentes de n52 se generó una versión de sus librerías con la librería *Geotools 2.1* (cambiando su referencia en el fichero pom.xml de dependencias de Maven).

```
<dependency>
  <groupId>org.geotools</groupId>
  <artifactId>gt2-main</artifactId>
  <version>2.4.2</version>
</dependency>
```

fue cambiado por:

```
<dependency>
  <groupId>org.geotools</groupId>
  <artifactId>gt2-main</artifactId>
  <version>2.1</version>
</dependency>
```

Al cambiar la referencia, dos clases de la librería *wps-io* se "quejaban" al compilar (*GML2BasicGenerator* y *KMLGenerator*); fue necesario comentar de manera selectiva algunas secciones de las mismas, hasta conseguir la compilación.

De esta manera, se consiguió que gvSIG y n52 se "entendieran" al pasarse los datos en forma de FeatureCollection de Geotools. En este punto se consiguen ejecutar geoprocesos de WPS (v1.0) contra un servidor siempre que los parámetros de entrada del servidor admitan el esquema *gml* siguiente:

http://geoserver.itc.nl:8080/wps/schemas/gml/2.1.2/gmlpacket.xsd o
http://www.opengeospatial.org/gmlpacket.xsd

Geoprocesos que cumplen con esto son p.ej:
http://geoserver.itc.nl:8080/wps/WebProcessingService?SERVICE=WPS&REquest=DescribeProcess&Identifier=org.n52.wps.server.algorithm.SimpleBufferAlgorithm y
http://geoserver.itc.nl:8080/wps/WebProcessingService?SERVICE=WPS&REquest=DescribeProcess&Identifier=org.n52.wps.server.algorithm.simplify.TopologyPreservingSimplificationAlgorithm

En todo caso, para lograr una compatibilidad 100%, sería necesario que gvSIG utilizara una versión más avanzada de *Geotools* o bien que hubiera cambios en el proyecto 52north, y en este sentido también el presente trabajo queda abierto, pendiente de futuras mejoras.

3.4. Pruebas

Las pruebas fueron de dos tipos: programadas en Java y manuales, ejecutadas como usuario de gvSIG sobre la extensión en funcionamiento.

Pruebas programadas. Son repetibles y eficaces para detectar errores en el código; fueron elaboradas mediante **JUnit** (en el entorno integrado de Eclipse –ver Ilustración 15–). Además, en varias ocasiones, especialmente durante la fase de elaboración, se iniciaba el proceso de diseño a través de la programación previa de pruebas, siguiendo un modelo de TDD o "programación guiada por pruebas". Esto permitió ir validando el diseño y avanzar con mayor claridad hacia un API más sencillo y centrado en los objetivos.

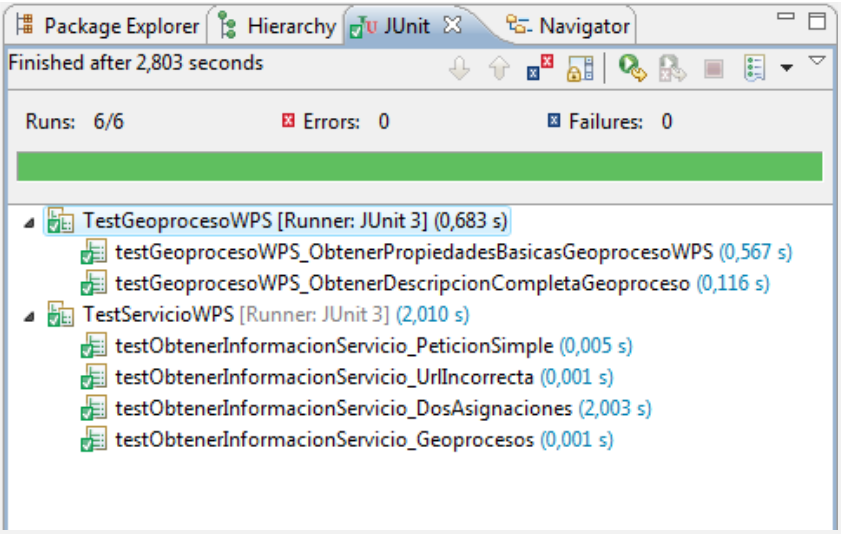

Ilustración 15. Pruebas de extWPS con JUnit

Pruebas manuales. Con la orientación de los casos de uso, se fueron realizando pruebas manuales sobre la extensión. Para realizarlas, se utilizó un juego de datos básico (una capa de puntos, líneas y polígonos)[13].

3.5. Instalador

Conforme la versión fue más estable, se comenzó a elaborar un instalador para la extensión y un manual de instalación. Se realizó una valoración de alternativas disponibles, eligiendo finalmente **IzPack** (que el propio gvSIG utiliza, y posee buena documentación técnica).

Con él se generó un instalador .jar multiplataforma, cuyo cometido fundamental es llevar a los directorios adecuados de gvSIG los binarios de la extensión, para que pueda ejecutarse. De manera complementaria, en el mismo instalador se incluyeron otros paquetes de datos:

- Paquete de **Documentación** (que incluye el manual de usuario).
- Paquete de **Fuentes** (del código y también de los documentos).

[13] En el instalador, bloque *Fuentes*, se incluye el código de los tests para Junit y los datos shapefile usados para las pruebas.

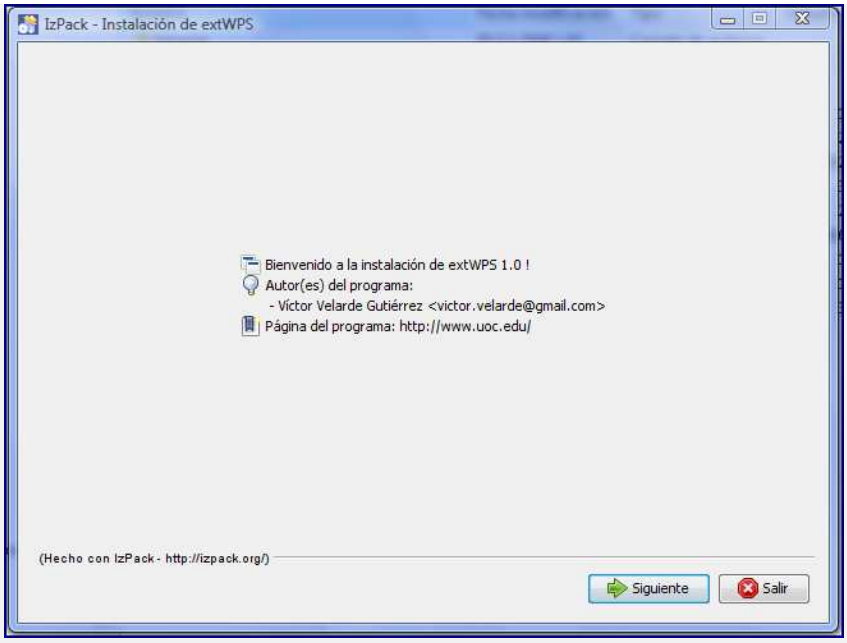

Ilustración 16. Instalador de la extensión extWPS

4. Conclusiones y recomendaciones

Incluiremos en este último apartado algunas reflexiones sobre la experiencia de desarrollo con gvSIG, las contribuciones de extWPS y los principales problemas encontrados. Se incluye finalmente un resumen de las posibles líneas futuras de desarrollo del proyecto.

1. La **integración** de gvSIG con otras soluciones abiertas es factible, aunque laboriosa. Se han podido añadir las capacidades de otro software libre (las librerías de 52north), gracias a cuatro elementos clave:

    a. la propia licencia GPL del software (tanto gvSIG como 52north WPS).

    b. la publicación del código en repositorios abiertos de Subversion.

    c. la extensibilidad de gvSIG.

    d. un trabajo de revisión intensiva de código, elaboración de muchas pruebas y consulta en foros de Internet.

2. Se observa que el ecosistema de **SIG libre en Java** en el que se integra gvSIG es bastante activo y cuenta con numerosas iniciativas[14], aunque esto quizá deriva en una cierta "dispersión de esfuerzos" de desarrollo. Hay piezas básicas de software compartidas en varios proyectos, como las librerías *Geotools* o JTS, pero su utilización no está exenta de problemas si de lo que hablamos es de interoperar con gvSIG.

3. gvSIG utiliza Geotools, lo mismo que 52north, pero debido a usar diferentes versiones y la incapacidad de reconciliarlas, la propia compilación de la extensión ha sido el principal problema técnico del proyecto (en general, la gestión de las dependencias de nuestra extensión sobre librerías externas). Como solución obligada a esta situación hubo que recompilar las librerías de 52north con la misma versión usada por gvSIG y generar un **fork** del proyecto, en lugar de usar sencillamente una versión cerrada y estable publicada en su web. Esto derivó además en que no estuviera disponible el 100% de funcionalidades ofrecidas por 52north en nuestra extensión.

4. De manera paralela, el tener que usar **Geotools** (GT2) como "lenguaje" de comunicación entre ambas piezas de software nos hizo investigar a más bajo nivel del esperado sobre esta librería. La excelente documentación de GT2 (sobre todo los ejemplos breves de su guía de usuario), así como porciones de

[14] En el entorno desktop destacan Jump y uDig, con los que gvSIG tiene bastantes similitudes y también una cierta "lucha" por ocupar el puesto de SIG Java de escritorio.

código similar en Jump, nos ha permitido avanzar en esta línea. Como producto principal se ha generado una clase de utilidad para convertir una capa vectorial de gvSIG a su representación como FeatureCollection de GT2, y a la inversa. Aquí hemos sentido que gvSIG tenía una carencia, ya que creemos que ésta debería ser una característica ya implementada en su núcleo.

5. El estándar **WPS** puede resultar a futuro un modo sencillo y eficaz de publicar y consumir geoprocesos, dentro del contexto general de arquitecturas orientadas a servicio (SOA). No obstante, a día de hoy el número de servidores WPS es realmente exiguo y la tecnología aún no ha cuajado. Durante el proyecto se han buscado sin apenas éxito servidores abiertos para probar e investigar. Los pocos encontrados estaban en versiones antiguas de prueba (WPS 0.4) que NO son compatibles con la nueva versión oficial 1.0 que nosotros utilizamos, p.ej. http://www.cartociudad.es/wps/WebProcessingService y http://www.idee.es/WPS/services. De este modo, habrá que esperar a que estos y otros servicios se actualicen a la nueva versión para que nuestra extensión pueda aportar verdadero valor al usuario base de gvSIG. Si esto va acompañado del avance de Sextante en su iniciativa de publicar sus geoprocesos vía 52north, entonces el resultado será aún mejor.

6. La **formación** específica en gvSIG y el desarrollo de sus extensiones ha sido fundamental para el proyecto. Ha habido un aprovechamiento importante de los materiales online ofrecidos por la comunidad gvSIG (cursos, documentos, foros…). En general, son materiales bastante válidos que permiten que programadores sin experiencia previa en gvSIG como nosotros puedan avanzar de manera autónoma. En el caso concreto de las consultas realizadas en la lista de correo para desarrolladores sobre WPS y gvSIG, la actitud, predisposición y velocidad en la respuesta ha sido buena, aunque quizá al ser el nuestro un tema tan específico (la integración de 52north) la aportación no haya sido determinante. La comunidad de 52 north (*Geoprocessing Community*) ha ofrecido también materiales interesantes, pero su lista de correo es muy poco activa.

7. Vistos otros proyectos como Geotools, se ha echado en falta en gvSIG más profundidad en la **documentación técnica** (p.ej. en forma de pequeños ejemplos de código bien organizados). También hemos comprobado que el código de gvSIG es, por cómo ha sido desarrollado, un código heterogéneo, sin estándares claros (de nomenclatura, identación, comentarios javadoc… falta una "guía de estilo"). Además faltan más tests automáticos en los fuentes de gvSIG para poder aprender de ellos (pruebas tipo JUnit).

8. El manejo adecuado de **herramientas de desarrollo Java** ha facilitado el trabajo y mejorado los resultados, pero exige un tiempo previo de formación si no se trabaja en ese entorno. Hablamos fundamentalmente de Eclipse, como elemento central a manejar con soltura, y de toda una serie de herramientas afines que no conocíamos (Ant, Maven, JUnit, IzPack…).

9. Al igual que con cualquier otro tipo de software, ha sido importante seguir un **proceso de desarrollo**. Aquí se ha aplicado con buenos resultados análisis, diseño y programación orientada a objetos y toda una serie de prácticas en un marco ágil (pruebas programadas, refactoring, automatización, desarrollo iterativo…).

Futuras líneas de trabajo

- Integrar la extensión extWPS dentro del marco general de geoprocesos de gvSIG 1.1.X (ver página 26).
- Llegar a una compatibilidad 100% de librerías entre gvSIG y 52north para poder utilizar correctamente todas sus capacidades. Tanto las ya implementadas sobre geoprocesos (como el manejo de datos en el estándar KML) como las próximas que estén por llegar (especialmente el manejo de capas raster en los geoprocesos).
- Adaptar la extensión extWPS al próximo gvSIG Desktop v2.0.
- Trabajar en otras áreas de WPS no incluidas en la versión 1.0 de la extensión, tales como la validación de parámetros de entrada para los geoprocesos, el cacheo de datos y las peticiones a WPS manejadas de manera asíncrona (generar indicadores de progreso en gvSIG, trabajar con tareas en segundo plano, historial de ejecución, etc.).
- Aumentar el número de pruebas unitarias y mejorar la cobertura del código.
- Analizar la posibilidad de desarrollar un asistente para "publicar un geoproceso local de gvSIG" como servicio WPS en un servidor basado en 52north.
- Perfeccionar, probar y probablemente refactorizar la clase UtilesCapas, en su parte de conversión Geotools hacia/desde capas vectoriales de gvSIG, para poder incluirla en otro apartado más genérico de utilidades de gvSIG.

- Analizar las posibilidades de sinergia con los desarrolladores de Sextante WPS (Víctor Olaya y otros), para unificar enfoques de cara a 52north y reutilizar componentes, de manera que se pudiera maximizar los beneficios del trabajo y no hubiera solapes. En esta línea, se podría conseguir un espacio propio en el área *Incubator* de la Comunidad de Geoprocesos de 52north, del mismo modo que existen para uDig o Jump.

Glosario básico

GML	*Geography Markup Language* (Lenguaje de Marcado Geográfico). Basado en XML, usado para el transporte y almacenamiento de información geográfica.
OGC	*Open Geospatial Consortium*. Organización internacional participada por empresas e instituciones del sector de la información geográfica enfocada a la definición de estándares de interoperabilidad (datos y procesos).
SIG / GIS	Sistemas de Información Geográfica / Geographic Information Systems
WFS	Web Feature Service. Servicio del OGC para acceder y manipular "Fenómenos" (entidades geográficas), que utiliza GML
WMS	Web Map Service. Servicio del OGC para la obtención de imágenes de mapas, habitualmente en formatos como PNG, GIF o JPEG.
WPS	Web Processing Service. Servicio del OGC para la ejecución remota de geoprocesos

ANEXO I. Bibliografía

Bataller A. et al. (2008). Gestión y desarrollo de proyectos, Conceptos y sugerencias. *Trabajo Final de Carrera.* Barcelona : UOC, 2008.

Botella Plana, A. (2008). Sistemas de Información Geográfica. *Nuevas tendencias en SIG* Barcelona : UOC, 2008.

Comas D. y Ruiz E. (1993). Fundamentos de los Sistemas de Información Geográfica. Barcelona, Ariel.

Consejeria de Infraestructuras y Transporte - Generalitat Valenciana. (2008). gvSIG. [En línea] 2008. http://www.gvsig.gva.es/

Foester, T. (2008). Web processing with OGC WPS Specification. [En línea] https://52north.org/twiki/bin/viewfile/Processing/GIdaysWPSworkshop?rev=1;filename=GI-daysWorkshop2008.pdf

Free Software Foundation. (2008). Licencias GNU. [En línea] 2008. http://www.gnu.org/licenses/licenses.es.html

Geotools 2 (2008). Geotools Developer Guide y User Guide [En línea] 2008 http://docs.codehaus.org/display/GEOT/Home y http://docs.codehaus.org/display/GEOTDOC/Home

Goodchild, M.F. et al. (2005). Geographic Information Systems and Science. Chichester, Wiley & Sons, 2005.

Gutiérrez Puebla, J. y Gould, M. (1994). SIG: Sistemas de Información Geográfica. Madrid, Síntesis.

Kareaga Cantero, I. (2008). Diseño y desarrollo de una herramienta de validación topológica en el entorno gvSIG. Proyecto Fin de Carrera. Ingeniería Superior en Informática. UOC. Dirigido por Jesús de Diego Alarcón. En: http//biblioteca.uoc.edu/

Open Geospatial Consortium. (2008). OGC Web Site. [En línea] 2008. http://www.opengeospatial.org/

Peng, Z.R (2003). Internet GIS: distributed geographic information services for the internet and wireless networks. New Jersey: Wiley & Sons.

Worboys, M.F. (2001). GIS: A Computing Perspective. London: Taylor Francis.

ANEXO II. Plan de gestión de riesgos

Riesgos	Medidas previstas
Avería del PC Principal de trabajo	• Se utilizará un repositorio remoto de *Subversion* (en *Assembla*) + copias de seguridad periódicas en soportes externos. • Se dispondrá de un entorno de trabajo configurado para emergencias, en un disco externo de 250 Gb, idéntico al principal y soportado mediante el software *VirtualBox*.
Riesgo tecnológico (desconocimiento de la tecnología)	• Plan de formación inicial intensivo en tecnologías, especialmente gvSIG y WPS • Apoyo en listas de distribución, estudio de manuales, etc.
Desajustes en la planificación	• Control semanal de planificación (y reajuste, si es necesario). • Recuperación de horas cuando sea necesario, ocupando tiempo de otros ámbitos.
Producción de software inestable	• Codificación guiada por requisitos, rediseño frecuente, testeo unitario, "balas trazadoras"…
Retrasos / problemas en las entregas	• Aumentar la frecuencia de las entregas (borradores) y soportar el proceso con herramientas. • Desarrollo, tanto de software como de otra documentación entregable, iterativo e incremental.