

# Diseño y Desarrollo de herramientas de validación topológica en el entorno gvSIG

## **Memoria del proyecto**

Iñaki Kareaga Cantero  
*Ingeniería Informática*  
*Proyecto de Fin de Carrera*  
*Curso 2007/2008-2*  
*Universitat Oberta de Catalunya*

## Resumen

Este documento describe los trabajos realizados en el ámbito del Proyecto Fin de Carrera del alumno. Se detallan las tareas ejecutadas y la documentación recogida a lo largo del proceso y su seguimiento. Primero, se analizan los Sistemas de Información Geográfica en general y, a continuación, la plataforma gvSIG y tecnologías relacionadas. gvSIG es una plataforma de software libre, fácilmente extensible, que aporta al usuario un interfase sencillo para el manejo de Bases de Datos geográficas desde diferentes fuentes y formatos. La solución software desarrollada en el ámbito del proyecto es una extensión para gvSIG. Esta solución permite realizar validaciones topológicas para *redes de agua* en esa plataforma. Además, el trabajo realizado sienta las bases para el desarrollo de herramientas de validación topológica de carácter más general.

## Abstract

This document describes the work done within the Master Thesis. It details the tasks performed and the documentation gathered throughout the process and its follow-up. First, Geographic Information Systems will be analyzed, and then the gvSIG platform and other related technologies. gvSIG is a free software platform, easily extensible, which provides a simple user interface for managing geographic databases from various sources and formats. The software solution developed within the project is an extension for gvSIG. This solution enables topological validations for water networks in that platform. The work also lays groundwork for developing more general tools for topological validation.

## Índice

Glosario.....	4
Introducción.....	6
1. Plan de Proyecto.....	7
2. Resumen de conceptos GIS y funciones del entorno gvSIG aplicables al caso en estudio.....	14
2.1. Conceptos generales sobre los Sistemas de Información Geográfica (SIG).....	14
2.1.1. Definición y funciones de los SIG.....	14
2.1.2. Modelos de datos en los SIG.....	14
2.1.3. Estructuras de datos en el modelo vectorial 2D.....	15
2.2. Plataforma gvSIG, características principales.....	16
2.2.1. gvSIG Desktop.....	17
2.2.2. PostGIS.....	18
2.2.3. Sextante.....	18
2.2.4. Piloto de redes.....	19
2.2.5. Versión en desarrollo FMAP->TOPOLOGIA.....	19
2.2.6. JTS.....	20
2.3. Caso en estudio.....	20
3. Catálogo de requisitos.....	23
3.1. Punto de partida.....	23
3.2. Catálogo de requisitos.....	23
4. Análisis.....	25
4.1. Casos de Uso.....	25
4.2. Modelo de datos.....	26
5. Diseño.....	29
5.1. Introducción.....	29
5.2. Caso de Uso: Lanzar geoprocso.....	29
5.3. Caso de uso: Realizar Validación.....	30
5.4. Casos de uso: Configurar parámetros de validación y Configurar simbología de resultado.....	31
5.5. Diagrama de clases de diseño.....	31
6. Implementación y pruebas.....	33
6.1. Introducción.....	33
6.2. Pruebas de unidad.....	33
6.2.1. Caso de uso "Lanzar geoprocso".....	33
6.2.2. Caso de uso "Realizar Validación".....	35
6.3. Pruebas de integración.....	40
6.3.1. Pruebas de integración usando los mismos datos de pruebas de las pruebas de unidad.....	40
6.3.2. Pruebas de Integración sobre datos aportados por el consultor.....	41
Conclusiones y recomendaciones.....	47
Agradecimientos.....	49
Lista de referencias.....	50
ANEXO I. Manual de usuario de la extensión WaterNetwork para gvSIG.....	51
1. Requisitos de la solución software.....	51
2. Instalación de la extensión WaterNetwork para gvSIG.....	51
3. Selección de capas a validar y ejecución del geoprocso.....	52
4. Interpretación de los resultados de la validación.....	55
5. Modificación de los parámetros de validación y simbología de los resultados.....	56
ANEXO II. Procedimiento seguido para la instalación del entorno de desarrollo.....	59
ANEXO III. Referencias al paquete de instalación, código fuente y a los datos de prueba.....	60
ANEXO IV. Bibliografía.....	61

## Glosario

- Accesibilidad a actividades de una red: Aplicando un modelo de oferta y demanda de actividades en los nodos de una red, existen diferentes medidas que permiten calcular la facilidad de acceso que un nodo tiene (demanda) a algún tipo de actividad distribuida en otros nodos de la red (oferta).
- Accesibilidad topológica de un nodo: La mayor o menor cantidad de aristas y nudos que es necesario recorrer para llegar a ese nodo desde el nodo más distante topológicamente a él.
- ANDAMI: Gestor del interfase de usuario utilizado en la plataforma gvSIG extensible mediante plugins
- Árbol: véase definición comparada en la entrada de glosario *Línea / Árbol / Red*
- Arco: Elementos lineales de las redes, formados por uno o más segmentos rectos. El atributo más importante que se les asigna es la longitud o coste de recorrerlas. En gvSIG se denominan polilíneas.
- Arista: véase *Arco*.
- Camino óptimo: Camino de menor coste de los posibles a través de la red entre dos nodos.
- Camino: Recorrido de la red desde un nodo origen a un nodo destino.
- Cohesión topológica: Grado de interconexión en una red. Existen medidas del grado de cohesión topológica como, por ejemplo, el número de aristas en relación al número de nudos.
- FMAP: Núcleo con las clases para gestión de GIS en la plataforma gvSIG
- GIS (Geographic Information System): Sistema de Base de Datos que contiene información geográfica o georeferenciada
- GPL (General Public License): Es una licencia orientada principalmente a proteger la libre distribución, modificación y uso de software
- gvSIG: Es un sistema de información geográfica programado en JAVA y basado en diferentes bibliotecas y programas de software libre.
- JAVA: Lenguaje de programación orientado a objetos. La plataforma gvSIG se ha desarrollado en este lenguaje.
- JTS: Librería de código abierto para el lenguaje JAVA que pone a disposición del programador objetos con herramientas de topología 2D para tareas de análisis espacial
- Línea / Árbol / Red: Son tres tipos de objetos lineales, según la topología: líneas (sin uniones entre ellas), árbol (segmentos con intersecciones pero sin formar bucles), red (líneas unidas y con posibilidad de formar bucles).

Desde el punto de vista de los SIG, una red representa un sistema interconectado lineal, que forma una estructura espacial por la que pueden pasar flujos de algún tipo (red de aguas, red de carreteras, etc.)

- Modelo de localización de una red: Modelos matemáticos que intentan establecer la mejor localización de un nodo, de manera que se maximice la accesibilidad a actividades de los mismos para el resto de la red.
- Modelo topológico: El conjunto de las relaciones que deben cumplir las geometrías de las entidades espaciales; consigo mismas y con otras geometrías, explicitadas a modo de restricciones o reglas.
- Nodo: Elementos puntuales (de tipo punto) de la red que se sitúan sobre las aristas. Pueden identificar orígenes y destinos de los flujos que circulan por la red.
- Nudos: Interconexión entre aristas.
- OGC (Open Geospatial Consortium): Es una organización internacional cuyo fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica
- Polilínea: véase *Arco*.
- POSTGIS: Se trata de una extensión para el SGBD relacional PostgreSQL que aporta soporte para objetos geográficos al mismo
- Red: véase definición comparada en la entrada de glosario *Línea / Árbol / Red*
- SEXTANTE: Es un software de escritorio basado en SAGA, a partir del cual se han desarrollado un conjunto de más de 200 extensiones GPL para la plataforma gvSIG.
- SIG (Sistema de Información Geográfica): véase GIS.
- Sinuosidad / Razón de sinuosidad: La relación entre la longitud real de la línea y su longitud si fuese recta. Sirve para comparar la intensidad de la simplificación realizada al representar el objeto geográfico real mediante una arista formada por segmentos rectos.
- Topología: véase *Modelo topológico*.
- Validación topológica: Estudio de una o varias capas de manera que cumplan las restricciones o reglas necesarias para adaptarse a un modelo topológico definido.

## Introducción

Este Proyecto fin de carrera trata de realizar un trabajo de síntesis de los conocimientos adquiridos en otras asignaturas de la carrera y ponerlos en práctica conjuntamente en el marco de un trabajo concreto. Se podrá observar que en este proyecto se ponen en práctica los conocimientos de Bases de Datos y Programación Orientada a Objetos adquiridos a lo largo de los estudios, entre otros. Además de éstos, la memoria del proyecto introducirá al lector en el mundo de los SIG (*Sistemas de Información Geográfica*), o GIS (*Geographic Information Systems*) en inglés.

Los GIS son Sistemas de Base de Datos que gestionan información geográfica o georeferenciada. Son de gran utilidad para analizar información geoespacial y resolver diversos problemas de gestión y planificación. Aportan herramientas que permiten acceder a la información de manera gráfica y extraer de manera sencilla información asociada a los objetos geográficos representados.

Los objetivos del proyecto son:

- conocer qué es un GIS
- manejar las tareas más habituales de análisis
- familiarizarse con el entorno gvSIG.

Por otro lado, gvSIG es una plataforma abierta de GIS orientada al usuario y distribuida con licencia GPL (*General Public License*). Debido al carácter abierto de esta plataforma de software libre, además de conocerla a nivel usuario, será posible también diseñar y desarrollar herramientas que se integren en este entorno. Estas nuevas herramientas desarrolladas constituirán la solución software del proyecto.

- El desarrollo de esta solución software forma también parte de los objetivos del proyecto.

Será necesario, por tanto, configurar el entorno donde se va a desarrollar la solución y aprender las peculiaridades de la programación en GIS. Se deberán analizar los diferentes predicados y funciones espaciales en 2D que la plataforma gvSIG aporta al programador, así como las diferentes librerías/proyectos Software Libre (SL) que les dan soporte.

La solución a desarrollar estará formada por un conjunto de herramientas que, mediante el uso de funciones espaciales, permitan la validación topológica de una *red de distribución de agua*.

El alcance del proyecto se divide en dos apartados principales:

- Estudio de la tecnología GIS y el entorno gvSIG y sus extensiones. Dentro de la tecnología GIS, realizar un estudio más detallado de la aplicación de esta tecnología en los problemas relacionados con las redes y, más concretamente, la validación topológica de una red de distribución de aguas mediante herramientas GIS.
- Desarrollar un software en el entorno gvSIG que permita la validación topológica de una red de distribución de agua.

Primero se redactará un Plan de Proyecto que resuma el alcance del mismo y divida el trabajo mediante una tabla de tareas. Se realizará también una planificación de las mismas. Antes de la redacción final de la memoria del proyecto (junto a la que se entregarán también la solución software, datos de prueba, código fuente, etc.) se realizarán entregas parciales del trabajo realizado.

Por último se realizará una presentación virtual que consistirá en un vídeo explicativo del proyecto y de la solución software desarrollada.

# 1. Plan de Proyecto

Este apartado representa la primera entrega del Proyecto de Fin de Carrera. Mediante el mismo, se planificará el trabajo a realizar durante el semestre.

Primero, se analizarán los **objetivos** planteados en el enunciado y se intentará definir, a grandes rasgos el **alcance** del mismo. A continuación, se enumerarán los **entregables** del proyecto, los diferentes productos que se irán obteniendo como consecuencia del trabajo para ser evaluados por el Tribunal de evaluación.

La planificación temporal de los trabajos se muestra a continuación. Primero se descompone el alcance del proyecto en un conjunto de **tareas y subtareas**. Después se propone un **calendario de trabajo** y la **planificación temporal** correspondiente de las tareas previstas teniendo en cuenta los hitos del proyecto.

El documento concluye con una **previsión de incidencias, riesgos y un plan de contingencia**, la enumeración del **material** que se deberá utilizar y la **bibliografía** que se ha utilizado para realizar este documento.

## Alcance y objetivos del proyecto

Los objetivos que aparecen en el enunciado son los siguientes:

- Conocer qué es un SIG y manejar las tareas más habituales de análisis.
- Familiarizarse con el entorno gvSIG/SEXTANTE. Conocer y manejar el software gvSIG y sus extensiones, principalmente SEXTANTE. Configurar el entorno para el desarrollo de nuevas funcionalidades sobre el entorno gvSIG (junto a POSTGRESQL + POSTGIS/Java Topology Suite + SEXTANTE ) en el IDE Eclipse.
- Aprender las peculiaridades de la programación en SIG. Analizar los diferentes predicados y funciones espaciales en 2D, así como en las diferentes librerías/proyectos Software Libre (SL) que les dan soporte.
- Comprender los problemas y las características de los entornos de redes.
- Diseñar y desarrollar un conjunto de herramientas que, mediante el uso de estas funciones espaciales, permitan la validación topológica de una red de distribución de agua.

Analizados los mismos, podríamos descomponer el proyecto en dos partes principales:

- Estudio de la tecnología GIS y el entorno gvSIG y sus extensiones. Dentro de la tecnología GIS, realizar un estudio más detallado de la aplicación de esta tecnología en los problemas relacionados con las redes y, a modo de ejemplo, la validación topológica de una red de distribución de aguas mediante herramientas GIS.
- Desarrollar un software en el entorno gvSIG que permita la validación topológica de una red de distribución de aguas.

## Entregables del proyecto

Los **entregables** del proyecto son los diferentes productos que se obtienen como consecuencia del trabajo, y que son entregados al cliente como resultados del proyecto. En este caso serán evaluados por el Tribunal de evaluación.

### 1 Memoria del proyecto

Entregable principal del proyecto, a entregar al final del mismo. Se intentarán respetar los aspectos formales expresados en el documento del tablón de la asignatura e incluir, en el cuerpo del mismo, el resumen de todas las tareas realizadas, tanto de estudio del entorno GIS como del desarrollo de software.

A parte de estos elementos, el cuerpo de la memoria, incluirá los siguientes apartados, que conforman la documentación de las tareas del proyecto expresadas más adelante. Estos apartados que se nombran a continuación servirán a su vez como entregables independientes en las entregas parciales del proyecto (PEC2 y PEC3):

#### 1.1. Resumen de conceptos GIS y funciones del entorno gvSIG aplicables al caso en estudio

Este documento será el resultado del estudio planteado la primera parte del Alcance del proyecto. Debería de concluirse antes de cualquier tarea relacionada con el desarrollo del software y servir como base para realizar el Análisis de este.

El documento partirá de una visión general y destacará elementos teóricos sobre los que, posteriormente, se construirá el caso práctico que pide el enunciado, en base a la documentación recogida en la bibliografía recomendada y otros recursos de Internet.

#### 1.2. Catálogo de requisitos

Texto que recogerá los requisitos de la solución software.

#### 1.3. Documentación del Análisis

Recogerá los casos de uso que se expresarán mediante texto y gráficamente. Contendrá también las clases de entidad identificadas y las relaciones entre estas, todo ello expresado mediante el diagrama de clases correspondiente.

#### 1.4. Documentación del Diseño

Contendrá las secuencias de operaciones para cada uno de los casos de uso e incluirá el diagrama de clases ampliado (operaciones y clases/atributos aparecidos en el diseño de los casos de uso). También detallará el diseño de la interfaz de usuario

#### 1.5. Manual de usuario

Contendrá el manual de uso de las funcionalidades implementadas y también las instrucciones para la instalación y configuración del Producto instalable.

#### 1.6. Presentación virtual

Presentación de diapositivas que resumirá la memoria. A entregar al final del proyecto.

### 2 Plan de proyecto e Informes de seguimiento

El plan de proyecto es este mismo documento y marca el inicio de la planificación.

Los Informes de seguimiento serán documentos breves que mostrarán el grado de cumplimiento de la planificación en los hitos intermedios (PEC2 y PEC3) y de los problemas e incidencias acontecidos.

### 3 Código fuente

Resultado de la implementación, debidamente estructurado y comentado.

### 4 Producto instalable

Incluirá todos los elementos necesarios para ejecutar la solución

## Tareas y subtareas

### 1 Estudio preliminar

- Estudio de la bibliografía recomendada en el enunciado
- Comprensión de los conceptos más importantes sobre GIS y su aplicación en las redes
- Búsqueda de información en la web sobre la validación topológica de redes.
- Búsqueda de información en las web de gvSIG y SEXTANTE. Búsqueda de información en la web del resto de tecnologías planteadas: JTS y PostGIS

Todo esto se deberá documentar convenientemente. Se redactará, por tanto, un documento con la parte teórica necesaria para el proyecto. Se deberá poner especial atención en la elaboración de un glosario de términos sobre GIS y gvSIG aplicado al caso en estudio y a la recopilación de referencias y bibliografía.

Se estima una carga de trabajo de 20 horas, realizadas simultáneamente con la tarea 2.

### 2 Instalación y configuración del entorno de desarrollo

- Búsqueda de información al respecto en Internet
- Instalación del entorno con la ayuda de tutoriales o documentación
- Ejecución del caso práctico de gvSIG disponible en la web oficial
- Tomar nota de los trabajos realizados y las referencias utilizadas

Se estima una carga de trabajo de 20 horas, realizadas simultáneamente con la tarea 1.

### 3 Especificación de los requisitos de la solución software

Se iniciará cuando concluya la tarea 1.

- Estudio del enunciado y los detalles del estudio preliminar para elaborar una lista de requisitos de la solución software
- Redacción del Catálogo de requisitos

Se estima una carga de trabajo de 10 horas.

### 4 Análisis

Se iniciará cuando concluya la tarea 3.

- Partiendo de la lista de requisitos, se agruparán estos en casos de uso y se identificarán los actores principales
- Se elaborará un diagrama de casos de uso
- Una vez determinados los casos de uso, se identificarán las clases entidad que deberán

usarse en el modelo orientado a objetos y sus atributos

- Se elaborará un diagrama de clases
- Se redactará el documento de Análisis

Se estima una carga de trabajo de 20 horas.

## 5 Diseño

Se iniciará cuando concluya la tarea 4.

- Se realizará una lista de operaciones que formarán la ejecución de cada caso de uso
- Se identificarán las funciones de la API del entorno gvSIG o de sus extensiones que deberán ser invocadas y el punto de la ejecución donde se invocarán
- Se ampliará el diagrama de clases elaborado en la tarea 4 con las operaciones identificadas y las clases/atributos adicionales identificadas en el diseño.
- Se redactará el documento de Diseño

Se estima una carga de trabajo de 10 horas.

## 6 Implementación

Se iniciará cuando concluya la tarea 5.

- Implementación en JAVA del diseño realizado, con los debidos comentarios.
- Generación del ficheros JAVADOC

Se estima una carga de trabajo de 30 horas.

## 7 Pruebas

Durante el proceso de pruebas se documentarán los resultados para ser incluidos en la consolidación de la memoria.

### 7.1. Pruebas unidad

A realizar en paralelo a la implementación, para validar el funcionamiento interno de cada clase implementada. Conjuntamente con la tarea 6.

### 7.2. Pruebas de integración

Se iniciará cuando concluya la tarea 6.

- Pruebas del sistema en su conjunto.
- Se realizarán las correcciones pertinentes en las anteriores fases del desarrollo de la solución debidas a posibles fallos de integración.

Se estima una carga de trabajo de 5 horas.

## 8 Elaboración del producto instalable

Se iniciará cuando concluya la tarea 7.

- Se empaquetarán las clases compiladas en un fichero JAR con las librerías correspondientes de manera que la solución pueda ejecutarse mediante un sólo comando.

Se estima una carga de trabajo de 5 horas

## 9 Redacción del resto de la documentación del proyecto

### 9.1. Manual de Usuario

Una vez hayan concluido las pruebas de integración (tarea 7.2) y la solución funcione correctamente se redactará el Manual de Usuario. También deberá estar concluida la elaboración del producto instalable (tarea 8).

Se estima una carga de trabajo de 5 horas

### 9.2. Consolidación del cuerpo de la memoria

Se iniciará cuando concluya la tarea 9.1.

- Se integrará en la memoria la documentación redactada hasta la fecha
- también de los resultados de las pruebas
- información de los informes de seguimiento entregados en las PECs

Se estima una carga de trabajo de 10 horas.

### 9.3. Redacción de los aspectos formales de la Memoria

Se iniciará cuando concluya la tarea 9.2.

Se estima una carga de trabajo de 10 horas.

### 9.4. Preparación de la presentación virtual

Se iniciará cuando concluya la tarea 9.3.

Se estima una carga de trabajo de 5 horas.

## 10 Tareas de seguimiento del proyecto

Una vez entregado el Plan de Proyecto, las tareas de gestión del mismo se materializarán mediante la generación de los informes de seguimiento para las entregas correspondientes a la PEC2 y a la PEC3.

Se estima una carga de trabajo de 1 hora para la redacción de cada uno de los dos informes de seguimiento.

## Calendario de trabajo

Desde la entrega del plan de trabajo y hasta la fecha límite de entrega de la memoria y la presentación virtual, se dispone de 13 semanas. Estas 13 semanas marcan, por tanto, la duración del proyecto.

El calendario de trabajo se centrará en trabajar los fines de semana, y se dedicará aproximadamente 10 horas semanales al proyecto, divididas en 2 jornadas, sábado y domingo, de 5 horas cada una. Para que la comunicación con el consultor pueda ser más fluida, fuera de estas jornadas, se revisarán también los mensajes en el aula del campus virtual cada dos días aproximadamente.

Los hitos externos del proyecto a tener en cuenta son los siguiente:

- 15/04/2008: Entrega de la PEC2
- 20/05/2008: Entrega de la PEC3
- 09/06/2008: Entrega de la Memoria y Presentación virtual

Se intentará enviar siempre un borrador al buzón del consultor a lo largo de la semana anterior a cada entrega.

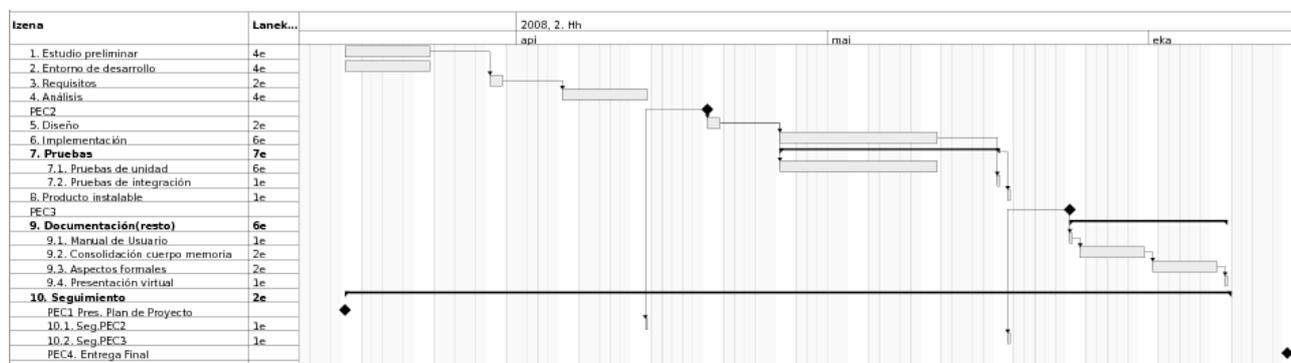
## Planificación

Combinando las tareas con el calendario de trabajo propuesto, se obtiene la siguiente planificación:

Tarea	Inicio	Fin
<b>PEC1. Presentación Plan de Proyecto</b>	<i>Mar 11</i>	
1. Estudio preliminar	Mar 15	Mar 23
2. Entorno de desarrollo	Mar 15	Mar 23
3. Requisitos	Mar 29	Mar 30
4. Análisis	Abr 5	Abr 13
<b>PEC2</b>	<i>Abr 15</i>	
5. Diseño	Abr 19	Abr 20
6. Implementación	Abr 26	May 11
<b>7. Pruebas</b>	Abr 26	May 17
7.1. Pruebas de unidad	Abr 26	May 11
7.2. Pruebas de integración	May 17	May 17
8. Producto instalable	May 18	May 18
<b>PEC3</b>	<i>May 20</i>	
<b>9. Documentación(resto)</b>	May 24	Jun 8
9.1. Manual de Usuario	May 24	May 24
9.2. Consolidación cuerpo memoria	May 25	May 31
9.3. Aspectos formales	Jun 1	Jun 7
9.4. Presentación virtual	Jun 8	Jun 8
<b>PEC4. Entrega Final</b>	<i>Jun 9</i>	

<b>10. Seguimiento</b>	
10.1. Redacción inf. Seguimiento PEC2	Abr 13
10.2. Redacción inf. Seguimiento PEC3	May 18

Esto mismo, expresado mediante un diagrama de Gantt:



Se puede observar que los entregables que estarían listos para cada entrega parcial (PEC2 y PEC3)

serían los siguientes:

- PEC2: Resumen de teoría, Catálogo de requisitos, Análisis y el Informe de Seguimiento correspondiente.
- PEC3: Diseño, Código fuente, Producto instalable y el Informe de seguimiento correspondiente.

Para las últimas tres semanas, por lo tanto, dejaría las sólo tareas de documentación pendientes.

### **Incidencias, riesgos y plan de contingencia**

En mi caso, no hay más asignaturas que puedan influir en las entregas, porque está es la única asignatura que estoy cursando este semestre. Sin embargo, no dispongo de tiempo para trabajar en el proyecto entre semana por motivos laborales y familiares. Intentaré centrar el trabajo los fines de semana, pero puede que necesite más tiempo para poder cumplir la planificación. En caso de necesidad clara, podría conseguir ayuda puntual de la familia.

Para evitar retrasos por pérdida o destrucción accidental de datos, se utilizarán los procedimientos de seguridad normales en estos casos (copias de seguridad, etc.).

Por lo demás, no se prevén otras incidencias que puedan influir en la planificación.

## **2. Resumen de conceptos GIS y funciones del entorno gvSIG aplicables al caso en estudio**

Este apartado describe el trabajo realizado en el Estudio preliminar a las tareas de desarrollo de la solución. Entre otros trabajos, se ha realizado la lectura de diferentes fuentes de información, se ha realizado un tutorial para comprender mejor el manejo del entorno gvSIG y se ha realizado una primera aproximación a la solución a desarrollar.

### **2.1. Conceptos generales sobre los Sistemas de Información Geográfica (SIG)**

Para la elaboración de este apartado se ha realizado una lectura del texto recomendado en el enunciado del proyecto (Bosque Sendra, Joaquín. Sistemas de Información Geográfica. RIALP, 1997). El texto aporta una visión general de la tecnología GIS ó SIG (Geographic Information Systems / Sistemas de Información Geográfica) y todos los aspectos asociados. Además de los conceptos principales, se han recogido numerosos términos relacionados con el análisis de redes, en el capítulo XI. Estos términos se han recogido a modo de resumen en el apartado "Glosario" de este documento.

#### **2.1.1. Definición y funciones de los SIG**

Un SIG se define como un sistema de Base de Datos que contiene información geográfica o georeferenciada. A estos sistemas se les asocian diferentes funciones:

- Entrada o digitalización de datos geográficos y preparación de los mismos para su gestión en el SIG. Estos datos tienen dos componentes. Por un lado el componente espacial que define objetos geográficos y, por otro, atributos temáticos asociados a los mismos que se expresan mediante variables temáticas.

Las variables temáticas se relacionan con los objetos geográficos representados en mapas mediante diferentes dimensiones geométricas (puntos, líneas, áreas o volúmenes) para representar información temática de manera gráfica. Existen diferentes niveles de medida para estas variables: nominales, ordinales y cuantitativas, que dan lugar a diferentes tipologías de mapas para la representación de los mismos. Para los mapas de puntos, por ejemplo, las tipologías respectivas son: mapas de puntos uniformes, mapas de símbolos y mapas de símbolos graduados.

- Gestión espacial de los datos que incluye búsquedas geográficas
- Análisis de los datos, mediante técnicas especiales para datos geográficos.
- Generación de resultados cartográficos

#### **2.1.2. Modelos de datos en los SIG**

La representación digital de los datos geográficos se puede realizar en los siguientes formatos:

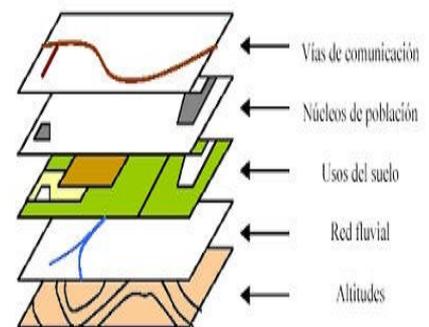
- Geocodificación de objetos geográficos partiendo del sistema de georeferenciación

tradicional. Se trata, básicamente, de capturar en una BBDD las coordenadas en base a un sistema de coordenadas estandarizado. La translación entre diferentes escalas y sistemas de coordenadas es una operación importante en este caso. También se puede realizar indirectamente recogiendo en la BBDD un nombre o número, que referencia indirectamente un lugar geográfico.

- Representación vectorial de la información espacial. Según la dimensión topológica de los objetos geográficos (puntuales, lineales, poligonales), se representan los límites o perímetro que los separa del entorno; mediante el almacenamiento de las coordenadas de los vértices que marcan el inicio y fin de los segmentos rectos que forman esos límites. Existen diferentes técnicas de estructuración de estos datos geográficos. Estas técnicas se estudiarán con más detalle en el siguiente punto.

Los datos vectoriales se suelen dividir según su extensión geográfica en hojas (a modo de los mapas tradicionales) y en capas, que forman los diferentes estratos temáticos de la representación vectorial.

La digitalización de los datos vectoriales se suele realizar mediante tabletas digitalizadoras, que son capaces de capturar las coordenadas de los vértices que se indiquen sobre un mapa tradicional. Por cada mapa digitalizado se deberán indicar, al menos, las coordenadas absolutas (referencia a un sistema de coordenadas universal) de 3 puntos no lineales para que se pueda realizar la traslación de coordenadas locales del mapa a puntos georeferenciados.



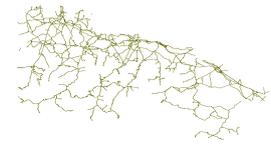
- Representación RASTER: Mapas de bits obtenidos a través de la digitalización como imagen de mapas tradicionales. Partiendo de una representación RASTER se puede obtener una vectorial por medio de la *vectorización*, que trata de identificar los objetos topológicos partiendo de mapas de bits, de modo automático o semiautomático
- Quadtree: Representación jerárquica/recursiva. Subdivisión de los objetos geográficos por niveles
- Modelado digital del terreno, MDT: Para la representación de la topografía, que incluye la información sobre la altura (tercera dimensión espacial) sobre el plano.

### 2.1.3. Estructuras de datos en el modelo vectorial 2D

Como ya se ha mencionado, el modelo vectorial en dos dimensiones espaciales representa mediante objetos geométricos el límite entre las entidades geográficas y su entorno, al estilo de los mapas tradicionales. Se trata por tanto de una representación simbólica que modeliza la ubicación de los lugares geográficos. Se usan los siguientes tipos de objetos:

- Puntuales (dimensión topológica 0): Se codifican mediante un par de coordenadas X e Y. Representan la ubicación espacial simplificada de un lugar geográfico. Por ejemplo, una población en un mapa con escala suficiente sin aportar representación de los objetos de menor jerarquía como calles, edificios, etc.

- Lineales (dimensión topológica 1): Los hechos geográficos que se ajustan a esta representación lineal (límites y fronteras, redes de transporte, hidrografía, etc...) se representan mediante una sucesión de segmentos rectos cuyos vértices se codifican con un par de coordenadas X e Y. Por tanto, se pueden codificar como una sucesión de pares de coordenadas.
- Polígonos (dimensión topológica 2): Se codifican como en el caso anterior aunque lo que representan es el área que encierran en vez de la línea que sirve de límite de los mismos. Sirven, por ejemplo, para representar divisiones administrativas en un mapa político.



Se observa que el elemento fundamental de referencia es el segmento lineal, delimitado por dos vértices (en los objetos puntuales ambos vértices tienen las mismas coordenadas y el segmento tiene longitud 0). A partir de este modelo de datos, se pueden usar diferentes estructuras para almacenarlos en memoria:

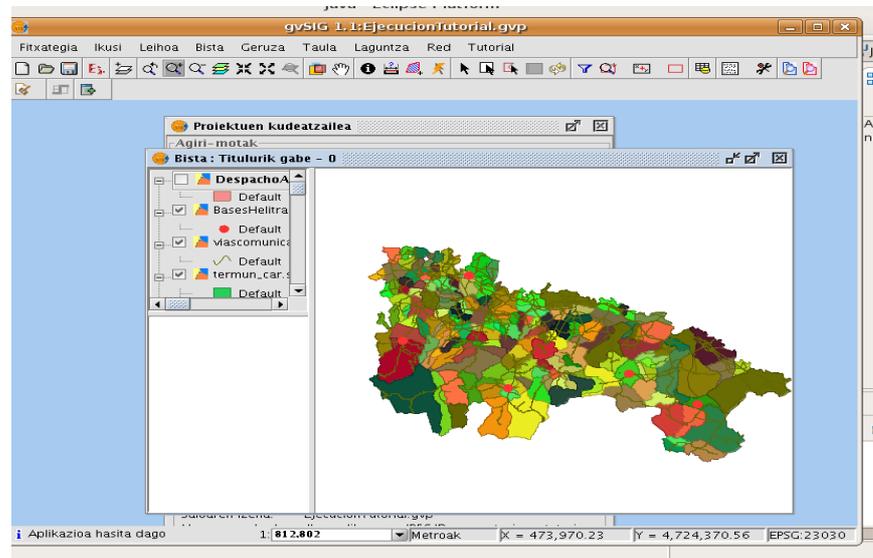
- Listas de coordenadas de los vértices de los polígonos: Para cada objeto geográfico se registra su nombre, el número de vértices y una lista de coordenadas de los vértices. Se repiten, por tanto, las coordenadas de los vértices comunes entre diferentes objetos.
- Diccionario de vértices: Se almacenan las coordenadas de todos los vértices (de todos los objetos) y listas de incidencia de los objetos (puntos, líneas o polígonos) con los vértices que los forman.
- DIME: Estructura que recoge la topología completa. Indicada para mapas de polígonos (mapas basados en divisiones artificiales del terreno, por ejemplo, divisiones administrativas). Se almacenan las coordenadas de los vértices de los segmentos. Para estos últimos se indica qué polígonos separa cada uno. De esta manera, resulta sencillo saber, por ejemplo, qué polígonos son contiguos. El segmento es la base de la codificación.
- Arco/Nodo: Un arco es una sucesión de segmentos rectos y un nodo es el vértice final de un arco. Estructura indicada para la representación de hechos geográficos de carácter natural (suelos, manchas de vegetación, redes, ...). La base de la codificación es el arco, para el cual se almacenan los nodos (origen y final) y los vértices intermedios entre los segmentos rectos. Los polígonos se representan en base a los arcos que los delimitan.

## 2.2. Plataforma gvSIG, características principales

Partiendo de la información de la página web oficial de gvSIG (<http://www.gvsig.gva.es/>) se ha realizado una evaluación de las posibilidades que aporta la herramienta, plataforma tecnológica y capacidades de personalización. Se han tenido en cuenta, además, otros proyectos como: SEXTANTE, JTS y POSTGIS.

## 2.2.1. gvSIG Desktop

En cuanto a gvSIG, se trata de un sistema de información geográfica programado en JAVA y basado en diferentes librerías y programas de software libre. La empresa *IVER Tecnologías de la Información* es la responsable del desarrollo pero forma parte de un proyecto más amplio (gvPONTIS) promovido por la *Generalitat Valenciana*, mediante el cual se pretende liberar (distribución bajo licencia GPL) el software desarrollado para la *Conselleria d'Infraestructures i Transport*.



Aporta un interfase de usuario sencillo para el manejo de BBDD geográficas desde diferentes fuentes y formatos, orientado al usuario final y muy extensible. Es a su vez software libre.

En cuanto a la arquitectura de la aplicación, se divide en 3 módulos principales:

- ANDAMI: Gestor del interfase de usuario extensible mediante plugins.
- gvSIG GUI: Extensión que contiene los elementos de interacción con el usuario de la aplicación base (menús, cuadros de diálogo, etc.)
- FMAP: Núcleo con las clases para gestión de GIS. Contiene entidades JTS y JAVA2D modificadas para los objetivos del programa, drivers para leer y escribir los diferentes formatos soportados y operaciones de bajo nivel con datos geográficos.

A partir de esa aplicación base, se programan extensiones basadas en estas librerías u otras (GeoTools, etc.) y se pueden integrar en la GUI. Se distribuye también una versión reducida para dispositivos móviles que permite interactuar con sistemas GPS.

Desde la web oficial (<http://www.gvsig.gva.es/>) se puede descargar una versión ya compilada e instalable con todos los componentes necesarios. Por lo tanto, el único requisito es tener la versión adecuada del JRE de JAVA preinstalada. De todas maneras se puede descargar una versión del instalador que incluye también el JRE.

En cuanto al funcionamiento de gvSIG sus características principales son:

- Vista: Permite añadir capas al proyecto y gestionarlas. En estas capas es donde se almacenarán y gestionarán los objetos GIS. Estos se cargan desde ficheros en diferentes formatos: formatos de datos GIS, como por ejemplo *Shapefile* (SHP), mapas en formato CAD o diversos formatos de imagen raster. También se pueden obtener estos mediante conexión a servicios de distribución de información GIS (WMS, WFS, WCS).
- Tabla: Gestiona la parte temática de la información que se carga mediante conexión a un SGBD, ficheros de texto (CSV), etc. El formato SHP incorpora los datos en tablas con formato dBASE (DBF).
- Mapa: Genera salidas de los resultados del programa combinando datos de diferentes capas

y tablas con otros elementos de los mapas como las leyendas, etc.

Para comprender mejor el funcionamiento del programa se ha realizado el tutorial para gvSIG disponible para su descarga en la web.

[http://www.gvsig.gva.es/fileadmin/conselleria/images/Documentacion/cursos/Tutorial-gvSIG-1\\_0.pdf](http://www.gvsig.gva.es/fileadmin/conselleria/images/Documentacion/cursos/Tutorial-gvSIG-1_0.pdf)

Mediante este tutorial, se han podido realizar las siguientes tareas y ver, de esta manera, cuales son las funcionalidades principales que aporta la versión de sobremesa de gvSIG y cómo se usan:

- Carga desde ficheros SHP de información GIS y su gestión dentro de una vista a modo de capas: activación, cambio de la simbología de los objetos vectoriales, ...
- Selección de diferentes conjuntos de objetos geográficos mediante consultas a la información temática, mediante filtros, selección manual, inversión de selección, ...
- Creación de ficheros SHP (capa) desde cero, con los objetos geográficos y datos temáticos correspondientes y edición de estos datos temáticos.
- Creación de nuevas capas a partir de un *geoproceso* sobre los datos de una o varias capas distintas (ejemplo: generación de polígonos que delimitan área de influencia de 25 km de varios puntos, como bases de extinción de incendios)
- Creación de un mapa como salida de los datos elaborados en el que se han insertado vistas, leyendas, escala e indicación del norte, texto libre e imágenes JPG externas. También se ha exportado ese mapa a formato PDF.

La ejecución del tutorial ha sido satisfactoria y sin problemas relevantes.

### 2.2.2. PostGIS

En cuanto a POSTGIS, se trata de una extensión para el SGBD relacional PostgreSQL desarrollado por la empresa *Refractions* y que aporta soporte para objetos geográficos al mismo. Aporta también la posibilidad de interactuar con la BBDD mediante una extensión del lenguaje SQL adaptada para GIS (Simple Feature Access: SQL Option). Esta extensión es un estándar establecido por el OGC (Open Geospatial Consortium).

*Simple Feature Access: SQL Option* incluye también funciones que calculan relaciones espaciales entre diferentes objetos geográficos, como la intersección, distancia, superposición, etc. Mediante estas sentencias SQL adaptadas se podría, por tanto, definir reglas topológicas.

### 2.2.3. Sextante

Por último, SEXTANTE es un software de escritorio basado en SAGA, a partir del cual se han desarrollado un conjunto de más de 200 extensiones GPL para gvSIG desarrolladas por la *Universidad de Extremadura* y financiadas por la *Junta de Extremadura*, que se centra principalmente en el modelado y análisis de la información mediante imágenes raster. La instalación de las extensiones se realiza mediante un instalador que se encarga de incluir la extensiones en gvSIG (indicando al instalador la carpeta donde está instalado el programa) de manera muy sencilla. Una vez instalado, las extensiones de SEXTANTE son accesibles en gvSIG.

Incluye también una herramienta denominada *Modelizador gráfico*, a través de la cual es posible definir nuevas extensiones sin tener que realizar tareas de programación, de forma gráfica. La definición de esos modelos, se realiza mediante la inclusión de entradas y procesos (extensiones o

modelos ya implementados) que se relacionan en un grafo. En principio, podría ser una plataforma útil para la definición de reglas topológicas.

#### 2.2.4. Piloto de redes

Extensión de gvSIG desarrollada por IVER que aporta a gvGIS la capacidad de gestionar topologías de redes (sobre capas lineales) y la de crear una leyenda de densidades de puntos (sobre capas de polígonos).

En cuanto a las topologías de redes, una vez generada asignando los costes de flujo a los arcos, permite el cálculo de rutas sobre esta topología. Las rutas son caminos realizados a través de una red a lo largo de diferentes "paradas" (objetos tipo punto, que se pueden guardar/cargar en una/desde una capa independiente). El piloto de redes calcula el coste total de una ruta y, además, es capaz de calcular la ruta de coste mínimo entre dos paradas.

La generación de leyendas de densidades de puntos, permite que a los objetos de tipo polígono se les pueda añadir una simbología de densidad de puntos en función del valor de una variable, relleno más denso para valores más altos de la variable y más puntos más dispersos para valores bajos.

Este piloto es una extensión, por lo tanto opcional en gvSIG. De todas maneras, en un futuro estas funcionalidades serán incluidas en la versión principal de gvSIG.

#### 2.2.5. Versión en desarrollo FMAP->TOPOLOGIA

Esta extensión aún está en desarrollo aunque se puede encontrar información sobre la misma en la siguiente dirección:

<https://gvSIG.org/web/docdev/docs/desarrollo/FMap/topologia>

De esta web, se ha recogido la siguiente cita, que explica qué es una topología y una validación topológica:

*En el mundo GIS se conoce como topología a las relaciones que deben cumplir las geometrías de las entidades espaciales; consigo mismas y con otras geometrías. Puesto que estas relaciones se explicitan en forma de restricciones, y puesto que cada modelo de datos de información vectorial puede forzar el cumplimiento de distintas restricciones (normalmente en función del uso que se vaya a dar a la información almacenada según este modelo) al conjunto de restricciones topológicas asociadas a un determinado modelo de datos vectorial lo conoceremos como modelo topológico, o simplemente, topología.*

(...)

*En caso de que no se cumplan las restricciones impuestas por estos modelos (modelo topológico implícito de OGC, modelo topológico de redes) los componentes de gvSIG encargados de la realización de estos análisis (geoprocesamiento, análisis de redes, etc.) lanzarán errores en tiempo de ejecución, o producirán resultados incorrectos.*

*Esto motiva la necesidad de dotar a gvSIG de funcionalidades de gestión topológica que permitan:*

- *Definir un conjunto de restricciones topológicas que han de regir las relaciones de los elementos geométricos consigo mismos, con otros elementos geométricos de una misma capa y con los elementos geométricos de otras capas (proceso conocido como "creación de topología").*
- *Validar el cumplimiento de las restricciones topológicas definidas en la topología por las diferentes geometrías puestas en juego en el modelo topológico.*
- *Visualizar aquellos casos que no pasaron el proceso de validación originando errores topológicos, tanto de forma gráfica como alfanumérica, proporcionando información sobre el tipo de error producido y los elementos vectoriales que intervinieron.*

- *Aplicar correcciones topológicas a las geometrías que causaron un error topológico, bien de forma automatizada mediante la ejecución de geoprocesos dedicados bien de forma interactiva haciendo uso de las herramientas de CAD de gvSIG.*

Por lo tanto, una validación topológica es, en la práctica, un estudio de una o varias capas de manera que cumplan las restricciones necesarias para adaptarse a un modelo topológico definido. A través de este módulo en desarrollo se dotará a gvSIG de las funcionalidades detalladas en la cita (creación de la topología, validación topológica y visualización/corrección de errores topológicos)

En la web que se ha mencionado se dispone de información sobre la arquitectura del módulo en desarrollo, las funcionalidades y el diseño. Desafortunadamente el desarrollo está en su etapa inicial y no se dispone aún de código para ser probado o adaptado.

## 2.2.6. JTS

JTS es una librería de código abierto desarrollada por la empresa *Vivid Solutions* para el lenguaje JAVA que pone a disposición del programador objetos con herramientas de topología 2D para tareas de análisis espacial. gvSIG hace uso de esta librería para tareas de geoprocesamiento que requieran de alguna operación básica del álgebra vectorial (unión, diferencia, intersección, diferencia simétrica): por ejemplo, la creación del área de influencia que se ha realizado en la ejecución del tutorial.

Conviene indicar que gvSIG no trabaja de forma nativa con esta librería para dibujar los objetos geográficos y que realiza una conversión de los mismos, como fase previa a las tareas de geoprocesamiento.

A través de la web de JTS (<http://www.vividsolutions.com/jts/jtshome.htm>) se puede descargar la documentación de referencia de esta librería.

## 2.3. Caso en estudio

El caso en estudio forma parte del ámbito del análisis de redes en el que se pide el desarrollo de una aplicación GIS que de soporte a dos tipos de objetos geográficos; según el enunciado:

- *Arcos. Constituyen los elementos lineales principales de la red.*
- *Nodos. Elementos que conectan los arcos. Se incluyen las uniones o elementos puntuales que se sitúan sobre los arcos sin partirlos. Estas últimas modelizan tanto elementos puntuales sobre los arcos como conexiones entre arcos del tipo fin de arco/vértice de arco.*

En cuanto a los arcos, son objetos lineales para los que gvSIG tiene soporte nativo. Para una red de aguas, cada arco modeliza una *tubería* (<http://es.wikipedia.org/wiki/Tuber%C3%ADa>) o una *acometida* (<http://es.wikipedia.org/wiki/Acometida>). La primera es un conducto de agua general de la red mientras que la segunda permite conectar la red particular de agua de cada edificio a la red general de aguas que se está representando. La diferencia en la modelización de ambas es que las *tuberías* están unidas a otras *tuberías* en sus extremos (a través de elementos de unión), mientras que las *acometidas* se unen a las *tuberías* a lo largo de estas últimas, sin partirlas. El concepto de longitud/coste de arco no resulta de interés en el caso en estudio, porque no se va a estudiar el control de flujo a través de la red de aguas.

En cuanto a los nodos, son objetos puntuales para los que gvSIG también ofrece soporte, pero el enunciado realiza una particularización de los mismos. En la definición genérica de la bibliografía se habla de elementos puntuales de una red, en general, sin más limitaciones. Sin embargo, el enunciado diferencia entre dos tipos de nodos:

- Uniones que marcan el inicio/final de un arco. Al contrario de en otros tipos de redes (como una red de carreteras, por ejemplo), en una red de aguas conviene modelizar el elemento que une las *tuberías*, porque una tubería no se une a otra directamente, sino a través de un elemento o accesorio de unión de tuberías. De manera simple, se hará referencia a estos elementos como *válvulas* (<http://es.wikipedia.org/wiki/V%C3%A1lvula>) de la red de aguas. Estos nodos de tipo *válvula* pueden almacenar atributos interesantes para gestión de flujo y control de la red a través de SIG (sin están abiertas o cerradas, la cardinalidad de arcos que pueden conectar como máximo según el tipo de válvula, ...). Todas las *tuberías* deberán interceptar, por tanto, una *válvula* en cada extremo.
- Elementos puntuales que no parten el arco sobre el que se sitúan. Representan *válvulas* donde se pueden conectar otras *tuberías* en un punto a lo largo de una *tubería* pero sin partirla en dos. También podrían representar otros elementos como *caudalímetros*, *hidrantes*, *dispositivos de purga*, etc...

Por lo tanto, a partir de una representación vectorial de líneas y puntos; se deberá crear una topología o modelo topológico para una red de aguas. Según el planteamiento realizado, la topología estará formada por las siguientes reglas topológicas:

- que cada *tubería* (arco) intercepte una *válvula* (nodo) en cada extremo y sólo una.
- que cada *acometida* (arco que intercepta otro arco en un punto a lo largo del mismo) no intercepte o cruce ningún otro elemento más, tampoco en su otro extremo.
- que las *válvulas puntuales* (nodos no iniciales ni finales de un arco) intercepten una *tubería* (que no sean puntos aislados, sino que estén sobre algún arco de la red actuando como nodos de la misma)
- que no haya *tuberías* ni *acometidas* que estén aisladas de la red, esto es, que cada arco intercepte, por lo menos, otro arco en alguno de sus extremos.
- ... todo esto, independientemente de las capas en las que se encuentren los objetos, porque en el caso de la validación topológica de redes de aguas, resulta de gran interés que la topología pueda formarse con objetos de diferentes capas. De esta manera, el usuario obtendría ventajas como: poder guardar una capa sólo con las *acometidas*, dibujar diferentes planos correspondientes a instalaciones que se superponen y compararlos, etc.

Por otro lado, el enunciado pide que la solución permita validar la red de aguas de manera que localice la aparición de los siguientes hechos topológicos:

- a. Autocruce (intersección) de arcos.
- b. Nodo sobre inicio/fin de arco.
- c. Nodo sobre vértice de arco (Unión).
- d. Punto inicial/final de arco sobre vértice de arco (Unión).
- e. Cruces entre arcos sin nodo.
- f. Cardinalidad de nodo/arcos.

La solución deberá de ser capaz de indicar cada una de las validaciones anteriores y de mostrar los errores topológicos. En gvSIG, se puede realizar un Geoproceso que dibuje en nuevas capas los objetos puntuales o lineales resultantes de la identificación de hechos topológicos.

Tras estudiar el piloto de redes, se observa que está orientado al cálculo de costes y no a la validación topológica de redes. Además sólo recoge información de una sola capa lineal y los nodos

no pueden ser recogidos desde una capa de puntos. De todas maneras el código de esta extensión se podrá tomar como referencia para la solución del caso en estudio.

En cuanto a la validación topológica, la versión en desarrollo de Topología para gvSIG podría ser de utilidad para hacer uso de la misma en el caso en estudio, pues permitirá definir una topología personalizada (red de aguas) y realizar la detección de errores topológicos en base a la misma. Pero, aún no está disponible ese desarrollo.

Las opciones restantes para el tratamiento de funciones topológicas en el caso en estudio son JTS, POSTGIS y el Modelizador gráfico de Sextante. Teniendo en cuenta que en el caso de estudio sólo se van a estudiar relaciones topológicas de intersección, no se prevé la necesidad de un entorno complejo. Tras estudiar la documentación de las tres opciones, se ha optado por JTS, que es la más simple y la que mejor se ajusta al caso en estudio.

Por lo tanto, se creará una extensión de gvSIG que sea capaz de ejecutar un *geoproceso* que valide las reglas topológicas y dibuje en unas nuevas capas de resultados (mediante nuevos objetos puntuales o lineales) los errores topológicos y los hechos topológicos encontrados en las capas estudiadas. Además de las librerías básicas de gvSIG, se usará, por tanto, la librería JTS para el análisis topológico.

## 3. Catálogo de requisitos

### 3.1. Punto de partida

El objeto de este apartado es elaborar una lista detallada de los requisitos funcionales de la solución que se va a desarrollar. Para determinar cuales serán estos requisitos se partirá del Estudio Preliminar realizado y del enunciado de la práctica.

Tras una lectura de los documentos mencionados podríamos resumir los objetivos de la solución mediante el siguiente texto:

El objetivo principal es poder validar una topología de redes para el caso concreto de una red de aguas en gvSIG. Se extenderán las funcionalidades de gvSIG, para poder realizar una validación topológica, en base a unas reglas topológicas concretas (correspondientes a una red de aguas) a modo de Geoproceso. Este geoproceso partirá de un conjunto de capas seleccionadas por el usuario, las analizará y generará el resultado de la validación. Este resultado podrá ser diferente según las opciones especificadas en el interfase de ese Geoproceso, pero tendrá como producto final la creación de varias capas resultado que contendrán objetos vectoriales para representar este resultado.

El siguiente guión describe la interacción del usuario con la aplicación:

Una vez el usuario haya cargado los objetos vectoriales en las diferentes capas y seleccione las que quiere que formen parte de la red de aguas, ejecutará un Geoproceso denominado *Validación de redes de aguas*. El sistema le pedirá qué tolerancia se aplicará al crear una topología basada en las capas seleccionadas. A continuación el sistema validará la topología en función de las reglas de validación formuladas para topologías de redes de aguas, teniendo en cuenta la tolerancia indicada por el usuario. En caso de existir algún error, el sistema informará al usuario de que se han encontrado errores topológicos y mostrará el resultado de la validación. Si la topología se ha creado correctamente, el programa mostrará al usuario un menú para elegir uno o varios hechos topológicos a buscar en la topología (los 6 del enunciado). El resultado de la ejecución serán nuevas capas donde se muestren esos resultados en forma de objetos puntuales o lineales con atributos asociados. Estos atributos aportarán información sobre el resultado: identificación del error encontrado o el hecho topológico identificado, para que el usuario pueda personalizar esa presentación configurando diferente simbología según esos atributos.

De este texto se deduce que la solución a desarrollar deberá ser capaz de cumplir una serie de requisitos que se detallan a continuación.

### 3.2. Catálogo de requisitos

Los requisitos se organizarán en dos grupos. Por un lado, se describen los requisitos básicos de la solución; los que se consideran necesarios para obtener las funcionalidades solicitadas. Por otro lado, se describen los requisitos adicionales que aportarán unas funcionalidades no necesarias pero sí de gran utilidad. En la implementación, se priorizarán los *requisitos básicos* frente a los *requisitos adicionales*.

#### *Requisitos básicos*

- Generar una topología de red a partir de los objetos lineales y puntuales de varias capas seleccionadas por el usuario.
- Sobre la topología generada, realizar una validación topológica en base a unas reglas topológicas concretas, almacenando en memoria los errores o hechos topológicos encontrados.

- Tener en cuenta en la validación, la tolerancia que se usará al calcular las distancias para considerar si los objetos se interceptan o no.
- Crear nuevas capas de resultados y dibujar en estas capas copias de los objetos originales que simbolicen el resultado de la validación topológica realizada, adjuntado los atributos necesarios para interpretarla.

### ***Requisitos adicionales***

- Mostrar al usuario las diferentes opciones de la validación topológica para que elija las que desea realizar.
- Posibilitar que el usuario modifique el parámetro de tolerancia en las distancias
- Posibilitar al usuario la configuración de la simbología del resultado

## 4. Análisis

### 4.1. Casos de Uso

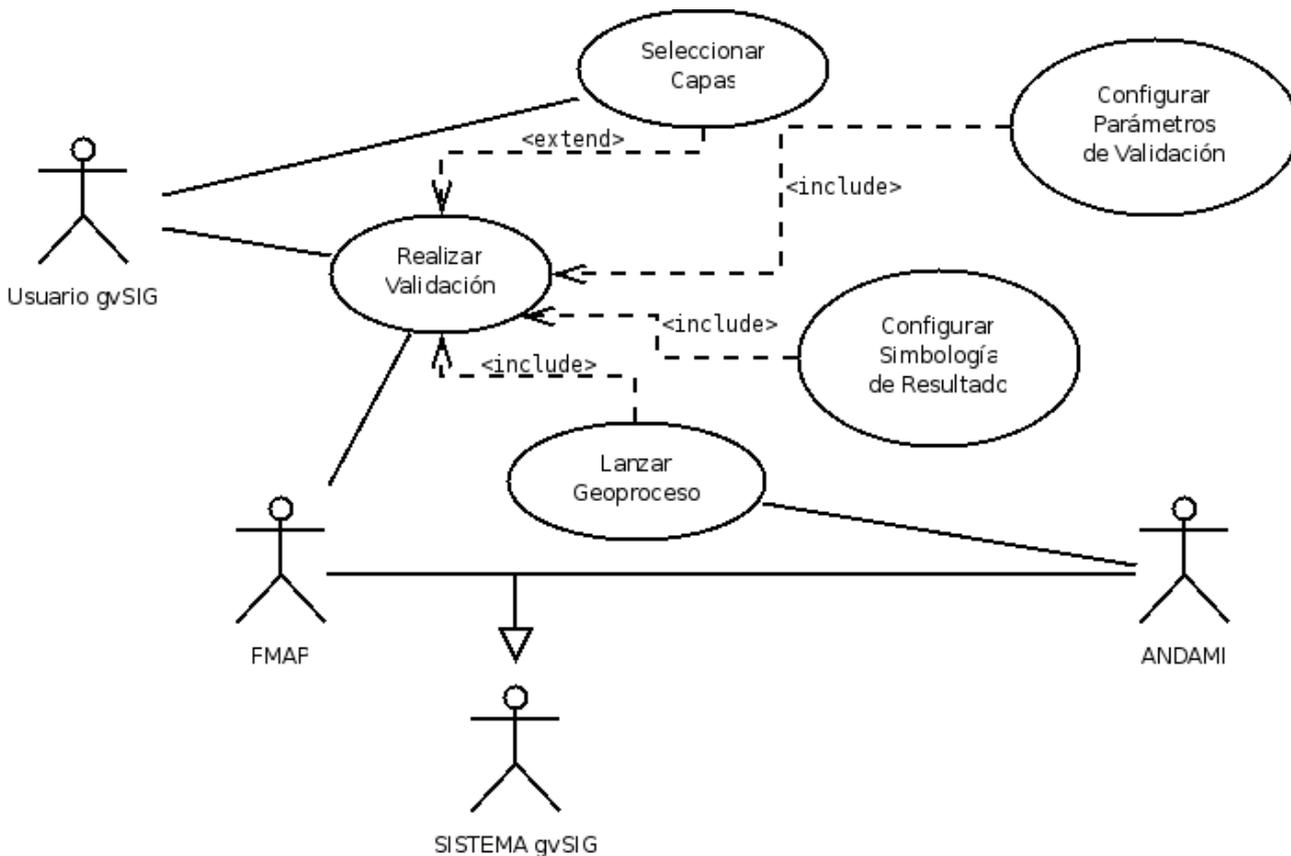
Los requisitos planteados indican que la solución no necesitará gran complejidad en la interacción con el usuario porque, desde el punto de vista de éste, el uso de la misma no será otro que *Realizar la validación topológica*.

En este caso, además, la solución se integrará en la herramienta *gvSIG Desktop*, por lo que gran parte de los casos de uso necesarios como requisitos para la validación topológica ya están implementados. Por otro lado el propio sistema gvSIG interactuará con los casos de uso de la solución aportando gran parte de las funciones necesarias y encargándose de la gestión de los datos. Por eso, no se deberán implementar. Todo ello a través de dos librerías: FMAP (núcleo de objetos geográficos) y ANDAMI (gestor del interfase de usuario). Además del Usuario de gvSIG, estas librerías también serán, por tanto, actores desde el punto de vista de los casos de uso de la solución.

En cuanto a los casos de uso, se han identificado los siguientes:

- **Seleccionar capas:** Contiene las operaciones necesarias para que, a través del interfase de usuario, pueda indicarse al sistema desde qué capas se tomarán los objetos geográficos para la Validación topológica. En realidad, es un caso de uso ya implementado en gvSIG para otros objetivos diferentes. Por tanto, no se deberá implementar. Se menciona aquí a nivel informativo, porque su ejecución forma parte del proceso necesario para la *Validación topológica*.
- **Lanzar Geoproceso:** Consiste en poner en marcha el proceso de Validación topológica a través del interfase de usuario de gvSIG. Interactúa con la librería ANDAMI además de con el usuario.
- **Configurar parámetros de validación:** Consiste en indicar al sistema qué reglas de validación van a ser aplicadas en vez de todas a la vez (comportamiento por defecto) y de la distancia de tolerancia que se establecerá en los cálculos de la misma. Estas son funcionalidades adicionales del catálogo de requisitos.
- **Configurar simbología de resultado:** El usuario podrá elegir, a través de este caso de uso, entre varias simbologías predefinidas para las nuevas capas de resultado. Esta también es una funcionalidad adicional del catálogo de requisitos.
- **Realizar Validación:** El caso de uso principal de la solución. Se encargará de Generar la topología de Red, Realizar la validación topológica y de mostrar el resultado mediante la Creación de nuevas capas con los atributos necesarios.

El diagrama de casos de uso resultante es el siguiente:



## 4.2. Modelo de datos

En cuanto a los datos, gran parte de los que se usarán en la solución son gestionados por la librería FMAP de gvSIG de forma nativa (objetos geográficos / *Shapes*, capas / *Layers*, ...), por lo que no se deberá analizar el modelo de datos para estas entidades que manejará la solución. El análisis se centrará por tanto en el modelo de datos derivado de las nuevas funcionalidades que se deben añadir: las correspondientes a la validación topológica.

Por un lado se ha identificado la entidad *Topology* que, cuando la topología se cree, agrupará una colección de referencias a objetos geográficos (*Geometry*) y a las capas (*FLayer*) que los contienen. Esta clase encapsulará las operaciones necesarias para realizar la validación topológica como tal en base a una colección de reglas (*Rule*) concretas.

La estructura de datos de estas reglas es uno de los puntos más importantes del modelo de datos propuesto. La validación se centrará en localizar si se cumple la intersección entre arcos y nodos y con qué cardinalidad sucede. Cada regla estará formada por condiciones (*Condition*) agrupadas mediante operadores AND y OR, teniendo en cuenta la precedencia de operadores en el orden indicado. Cada regla se validará partiendo de un objeto concreto y estudiando su intersección con todos los objetos de la topología del tipo indicado en el atributo *InterceptsA*, excepto él mismo.

En cuanto a las condiciones que componen una regla, se expresarán de la siguiente manera:

operator	objectType	interceptsA	minCardinality	maxCardinality	intersPointBase	intersPointSecondary
----------	------------	-------------	----------------	----------------	-----------------	----------------------

Cada atributo de los expresados admite estos valores:

- **operator:** Operador AND/OR a aplicar en el cálculo lógico con respecto a las anteriores condiciones de una misma regla. Se aplica la precedencia de operadores en el orden indicado.
- **objectType:** Tipo de objetos a los que se aplica la regla: ARC (arco), NODE (nodo) o ANY (cualquiera de ambos)
- **interceptsA:** Tipo de objeto respecto al que se estudia la intersección con respecto del indicado en el atributo anterior.
- **min/maxCardinality:** 0 indica que el objeto que se esta validando no se debe interceptar con objetos de la clase indicada en *Intercepta con* para que la condición se cumpla, un numero positivo indica la cardinalidad máxima o mínima en la intersección para validar la condición y un número negativo indica que no hay límite de cardinalidad.
- **intersPointBase:** Punto de intersección en el objeto base. ANY indica que se valida la intersección en cualquier punto, BEGIN únicamente en el vértice extremo inicial, PATH indica la intersección en un vértice a lo largo del objeto (y no en los extremos), END únicamente en el vértice extremo final y LAST indica el último vértice de intersección validado por una condición anterior en la misma regla.
- **intersPointSecondary:** Punto de intersección en el objeto interceptado. La posición del punto se expresará como en el atributo anterior.

Por ejemplo, la regla "*que cada tubería (arco) debe interceptar una válvula (nodo) en cada extremo y sólo una*" se expresará de la siguiente manera:

<i>Rule1</i>	ARC	NODE	0	0	BEGIN	ANY
OR	ARC	NODE	0	0	END	ANY
OR	ARC	NODE	2	-1	BEGIN	ANY
OR	ARC	NODE	2	-1	END	ANY

*Esta regla se validará para todos los arcos que no intercepten un nodo en sus dos extremos o que intercepten más de uno. Se validará, por tanto, para todos los arcos erróneos según el texto.*

Otro ejemplo "*que cada acometida (arco que intercepta otro arco en un punto a lo largo del mismo) no debe interceptar una válvula en su unión con la tubería y tampoco debe interceptar o cruzar ningún otro elemento más, ni en su otro extremo.*"

<i>Rule2</i>	ARC	ARC	1	1	BEGIN	PATH
AND	ARC	NODE	1	-1	BEGIN	ANY
OR	ARC	ARC	1	1	BEGIN	PATH
AND	ARC	ANY	1	-1	PATH	ANY
OR	ARC	ARC	1	1	BEGIN	PATH
AND	ARC	ANY	1	-1	END	ANY
OR	ARC	ARC	1	1	END	PATH
AND	ARC	NODE	1	-1	END	ANY

OR	ARC	ARC	1	1	END	PATH
AND	ARC	ANY	1	-1	PATH	ANY
OR	ARC	ARC	1	1	END	PATH
AND	ARC	ANY	1	-1	BEGIN	ANY

El conjunto de condiciones se validará para todos los arcos erróneos según el texto, esto es, para todas las acometidas (arcos) que presentan alguna válvula (nodo) en la unión con la tubería (arco interceptado) o que intercepten algún otro elemento en el otro extremo a lo largo del arco.

Dos último ejemplos, "Cruces entre arcos sin nodo" y "nodos aislados de la red"

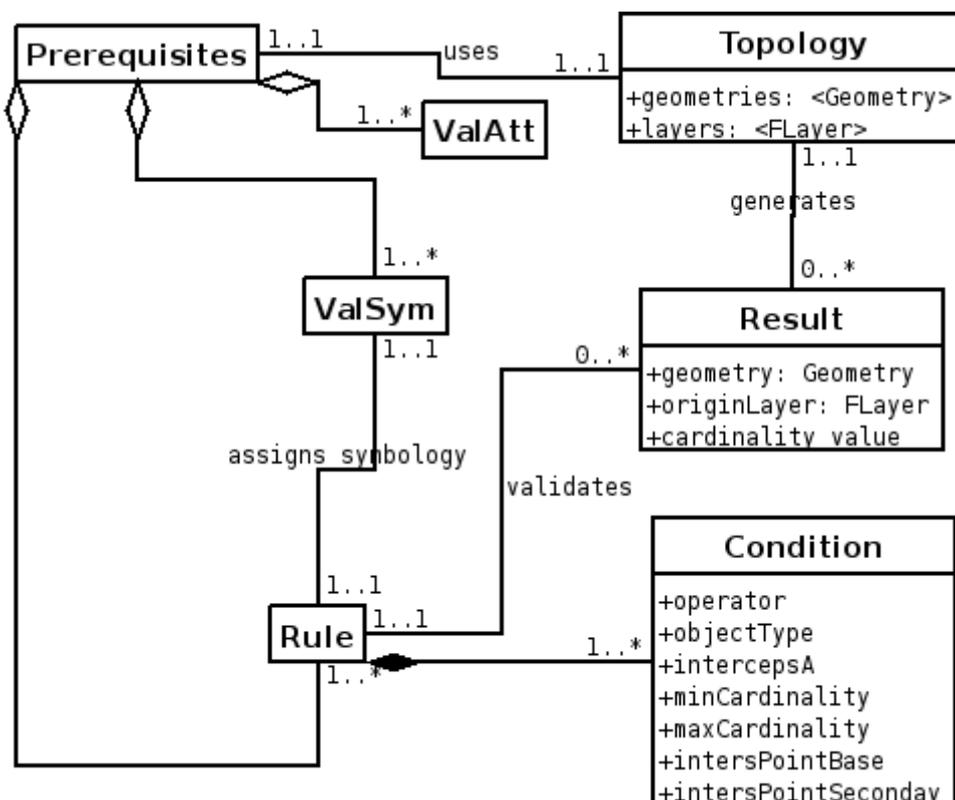
Rule3	ARC	ARC	1	-1	PATH	PATH
AND	ARC	NODE	0	0	LAST	ANY

Rule4	NODE	ARC	0	0	ANY	ANY
-------	------	-----	---	---	-----	-----

Los diferentes atributos de la validación (*ValAtt*) que el usuario podrá determinar así como las simbologías (*ValSym*) del resultado deberán también tenerse en cuenta en el modelo. Esta información se accederá a través de una instancia de la clase de prerequisites (*Prerequisites*).

Por último, los nuevos objetos geográficos resultantes de la validación (copiados de los objetos originales de la topología) se guardarán en instancias de la clase *Result*. Además del nuevo objeto geográfico (*Geometry*) correspondiente, también contendrá referencias a la capa (*FLayer*) original y un valor numérico que refleje el resultado de la validación en los casos necesarios (cálculos de cardinalidad).

El diagrama de clases resultante es el siguiente:



## 5. Diseño

### 5.1. Introducción

En este documento se describe el diseño de la solución software del proyecto. Partiendo del análisis realizado anteriormente, se tratará ahora de añadir la información necesaria para implementar las funcionalidades analizadas en el entorno gvSIG y de definir en detalle las operaciones que se deberán realizar en cada caso de uso.

A nivel de diseño, la solución consistirá en desarrollar un plugin para gvSIG con una extensión de tipo *geoproceso*.

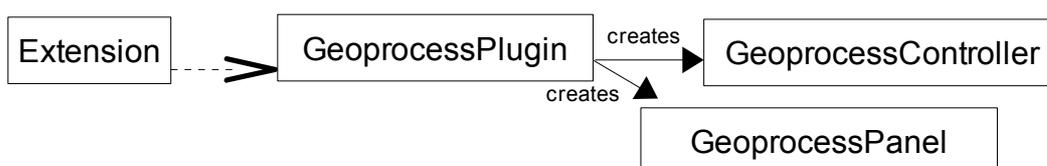
Primeramente, se describirán las operaciones y el diseño del interfase de usuario para cada caso de uso y, finalmente, se resumirán todas las clases, atributos y operaciones identificadas en el diagrama de clases de diseño.

### 5.2. Caso de Uso: Lanzar geoproceso

Las operaciones que se deberán realizar en este caso de uso son las siguientes:

1. Informar al usuario del proceso que se va a realizar y preguntar si quiere continuar con el mismo. El usuario dará su conformidad por medio de un botón tipo "Abrir geoproceso" y, en caso contrario, anulará el proceso por medio de un botón tipo "Cerrar". Esta operación no deberá ser implementada, porque gvSIG la realiza de forma nativa para todos los geoprocesos.
2. Recoger las capas seleccionadas por el usuario. En caso de que no haya seleccionado ninguna o alguna de las capas no sea del formato adecuado, informar al usuario y abortar el proceso.
3. Cargar en memoria los prerequisites de la validación (*Prerequisites*). Primero con valores fijos (por defecto) y, luego, dando opción al usuario de modificarlos a través de la invocación de los casos de uso *Configurar parámetros de validación* y *Configurar simbología de resultado* (estos últimos se implementarán en último lugar, si se dispone de tiempo suficiente, por no considerarse requisitos básicos de la solución).
4. Crear una instancia de la clase *Topology* pasándole como parámetro la lista de referencia a las capas sobre la que se va a efectuar la validación y a los prerequisites.

El geoproceso debe implementarse respetando el paradigma que gvSIG sigue para la implementación de las extensiones de geoprocesos. Se deben de implementar varias clases, que se estructuran según el diagrama adjunto.



Según este paradigma,

- La clase *extensión* se encarga de informar a ANDAMI que incluya la extensión y de qué tipo de extensión se trata, pasándole una referencia a la clase del Plugin correspondiente. En el fichero *config.xml* de configuración y en otros, se da información suplementaria para que el gestor de interfase de usuario pueda mostrarla en pantalla (descripción del geoproceto), sin tener que escribir código para esta labor.
- El *plugin del geoproceto* es la clase cuya instancia creará el sistema cuando se cargue la extensión en gvSIG y se encarga de interactuar con el gestor del interfase de usuario de gvSIG. También se encarga de crear la instancia del controlador del geoproceto y el Panel del geoproceto
- El *controlador del geoproceto* es el que ejecuta las operaciones del mismo. El método *launchGeoprocess()* es invocado cuando el usuario decide lanzar el geoproceto. Por tanto esta clase será la encargada de ejecutar las operaciones del geoproceto como tal.
- El *Panel del geoproceto* es el encargado en interactuar con el usuario. Intercambia mensajes con el controlador del geoproceto.

Los detalles del interfase de usuario vienen dados ya por gvSIG. Solamente será necesario escribir un fichero de texto con la descripción en los diferentes idiomas de lo que va a realizar el geoproceto e incluir una imagen que ayude a ilustrar ese texto.

### 5.3. Caso de uso: Realizar Validación

Es el caso de uso principal de la solución que incluye otros 4 casos de uso (*Seleccionar capas*, *Configurar parámetros de validación*, *Configurar simbología de resultado* y *Lanzar geoproceto*). Las operaciones que aquí se describen, son las operaciones que pertenecen al caso de uso *Realizar Validación* de manera exclusiva:

1. Generar la topología de red con todos los objetos que aparecen en las capas seleccionadas
2. Realizar la validación topológica para las reglas (*Rule*) correspondientes a **errores topológicos** almacenada en la instancia de la clase *Prerequisites*. Crear objetos de la clase *Result* en base a la validación de esas reglas.
3. En caso de que se hayan encontrado **errores topológicos**, informar al usuario mediante un aviso y pasar directamente al paso 5
4. Realizar la validación topológica para las reglas (*Rule*) correspondientes a **hechos topológicos** y crear los resultados igual que en paso 2.
5. Mostrar los **errores topológicos** o **hechos topológicos** encontrados, mediante la creación de nuevas capas resultado y añadir en estas copias de los objetos originales que han validado alguna de las reglas añadiendo los atributos necesarios para identificar la regla o las reglas que ha validado cada objeto.

Todas estas operaciones, se deberán añadir a modo de métodos en las clases identificadas en el análisis, de la siguiente manera:

- Clase *Topology*
  - **buildTopology ()**  
Se encargará de extraer referencias a todos los objetos de las capas y prepararlos para su posterior validación topológica

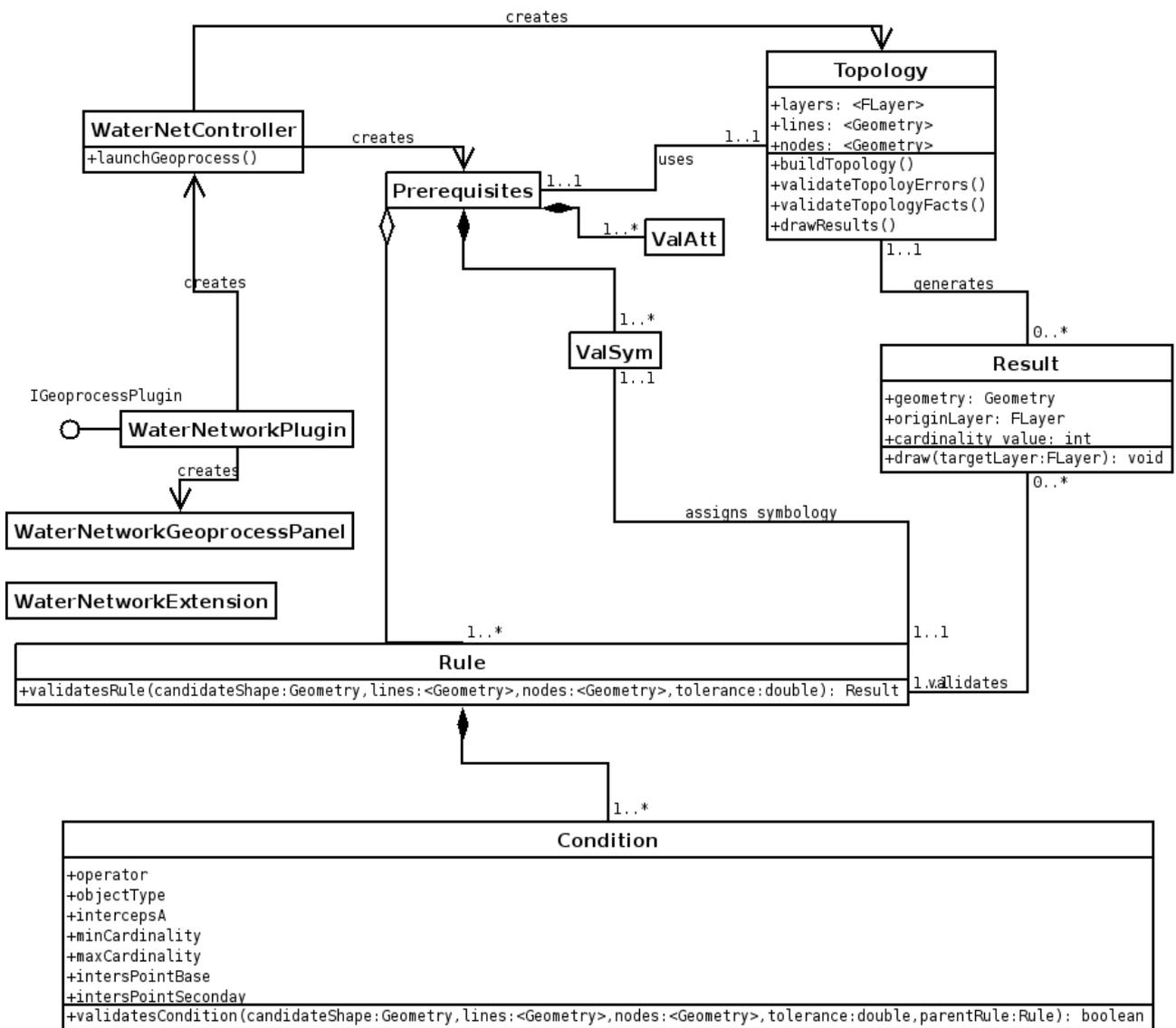
- **validateTopologyErrors ()**  
Se encargará de recorrer la lista de objetos y ejecutar la validación para todas las reglas correspondientes a **errores topológicos**. En caso de validaciones positivas creará instancias de los objetos resultado correspondientes que contendrán una.
- **validateTopologyFacts ()**  
Igual que la anterior pero para **hechos topológicos**.
- **drawResults ()**  
Se encargará de crear las nuevas capas necesarias, establecer los atributos que se usarán en cada una, asignares la simbología adecuada y de dibujar los objetos resultado en ellas.
- Clase Rule
  - Result **validatesRule** (*Geometry candidateShape*, <Geometry> *lines*, <Geometry> *nodes*, double *tolerance*)  
Ofrecerá un resultado de tipo *Result* (incluyendo todos los detalles del resultado de la validación) indicando si la regla se cumple o no para esa geometría tomando como referencia todas las demás. Mediante el parámetro *tolerance*, se indicará al método que tolerancia en la distancia entre los objetos para considerar que se interceptan y así validar la regla. Se encargará solamente de validar las diferentes condiciones que componen la regla y calcular sus resultados en conjunto con el operador OR o AND.
- Clase Condition
  - boolean **validatesCondition** (*Geometry candidateShape*, <Geometry> *lines*, <Geometry> *nodes*, double *tolerance*, *Rule parentRule*)  
Igual que la anterior pero sólo para validar una condición no toda la regla. Se ofrece una referencia a la regla a la que pertenece esa condición para poder acceder a consultar o modificar información necesaria para la validación entre unas condiciones y otras (último punto de intersección validado y cardinalidad).
- Clase Result
  - void **draw**(*FLayer targetLayer*)  
Utilizando las funciones de dibujo, se encargará de dibujar el objeto resultado en la capa indicada y de incluir los valores necesarios en las tablas de atributos temáticos para ese objeto dibujado.

Una vez terminado el caso de uso *Lanzar Geoproceso*, el sistema ya dispondrá de la información necesaria del usuario, por lo que no realizará más peticiones al mismo. Ahora, el sistema se centrará en las operaciones topológicas. Por lo tanto no se deberá de diseñar un interfase de usuario para este caso de uso.

#### **5.4. Casos de uso: Configurar parámetros de validación y Configurar simbología de resultado**

Estos casos de uso agrupan las funcionalidades adicionales. Estas se encuentran aún por diseñar.

#### **5.5. Diagrama de clases de diseño**



## 6. Implementación y pruebas

### 6.1. Introducción

En base al diseño expuesto anteriormente, se ha procedido a la implementación del mismo en lenguaje JAVA. Se ha usado el entorno de desarrollo cuyas instrucciones de configuración se adjuntan en el ANEXO II.

Además del código JAVA y junto a este, se han creado ficheros de texto adicionales y se han adaptado los ficheros de configuración necesarios para poder compilar e integrar correctamente la solución a modo de extensión de la plataforma gvSIG. Se puede encontrar la referencia a todos estos ficheros en el ANEXO III.

Se ha realizado también la internacionalización de los literales de interfase de usuario en inglés (por defecto), castellano y euskera.

Los comentarios sobre detalles de programación se han incluido junto al código, a modo de comentarios en el lenguaje JAVA. También se han añadido comentarios en formato JAVADOC que describen cada clase pública y sus métodos, para facilitar la reutilización del código a otros programadores. La documentación JAVADOC (en formato HTML) se ha generado junto al código. Las referencias a estos elementos se describen en el ANEXO III.

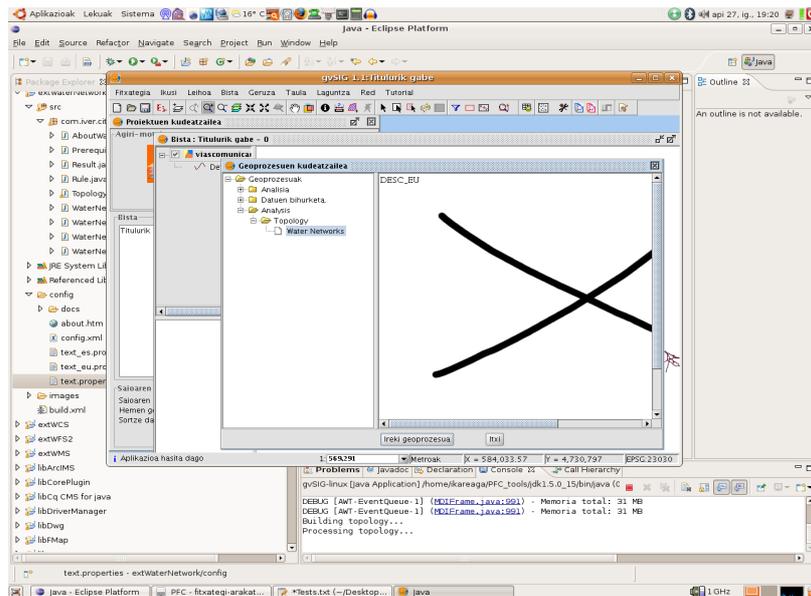
### 6.2. Pruebas de unidad

Se han realizado con el objeto de estudiar funcionalidades individuales. Debido al tamaño de la solución desarrollada, en vez de orientar las pruebas a nivel de clase, se han orientado a nivel de caso de uso. En el caso de uso principal "Realizar Validación" se ha optado por probar cada una de las reglas independientemente con la ayuda de datos preparados ad hoc para esta función.

#### 6.2.1. Caso de uso "Lanzar geoprocso"

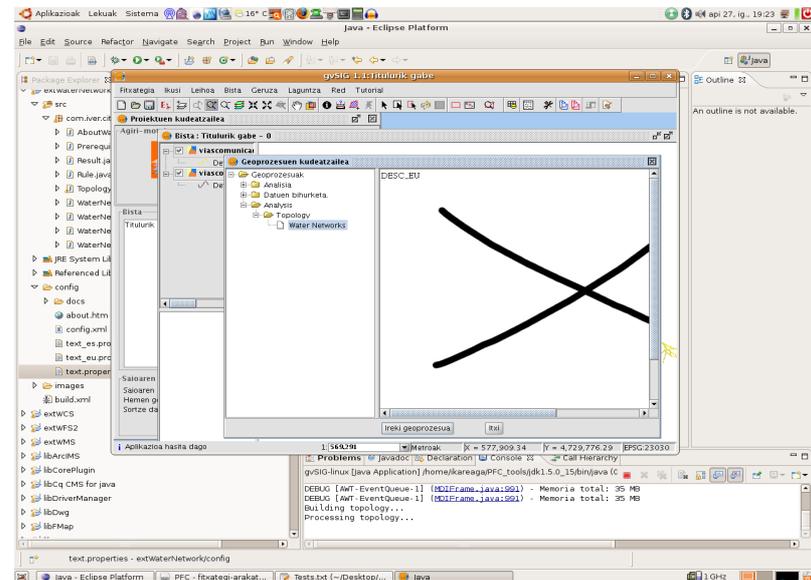
Se ha estudiado el comportamiento de este caso de uso en función de las capas que ha seleccionado el usuario y su tipología.

101. Ejecución normal con 1 capa correcta seleccionada

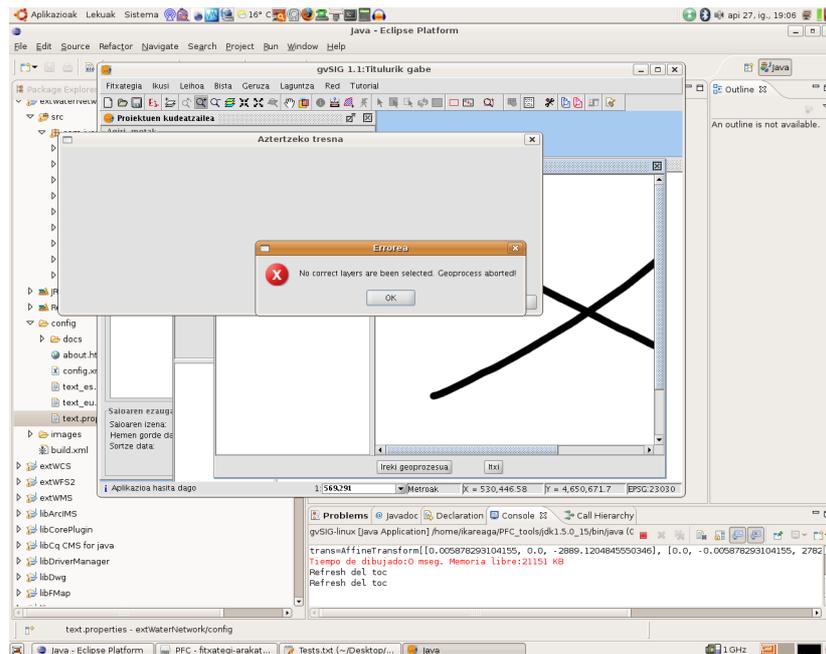


### 102. Cancelación del geoproceso

### 103. Ejecución normal con 2 capas correctas seleccionadas



### 104. Ejecución errónea, con ninguna capa seleccionada

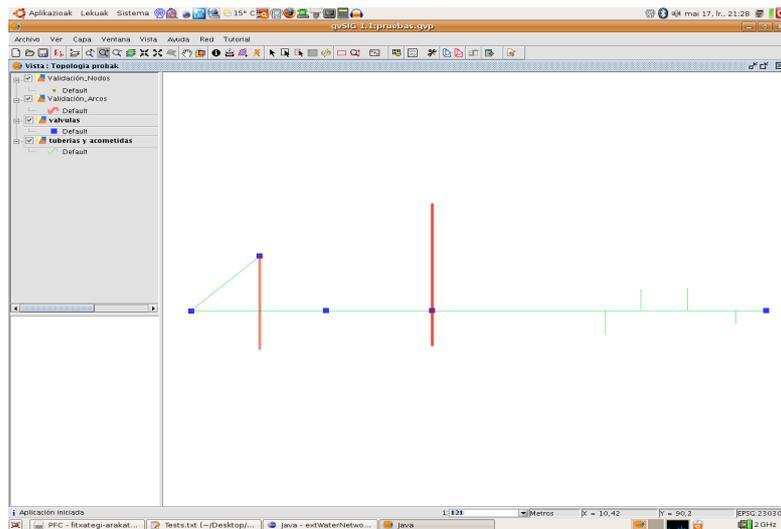


105. Ejecución errónea, con ninguna capa seleccionada

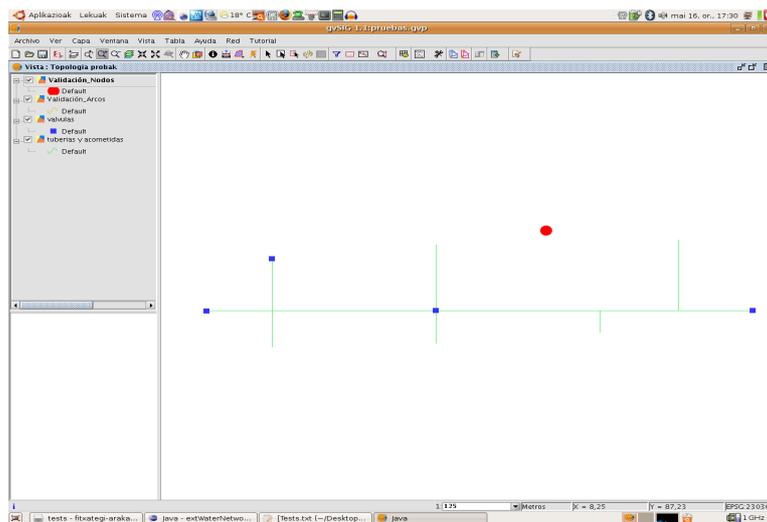
## 6.2.2. Caso de uso "Realizar Validación"

Se ha estudiado la ejecución de cada regla de validación con los datos de ejemplo necesarios con el objetivo de observar si la regla se cumple en todos los casos necesarios y asegurar que no ofrece "falsos positivos".

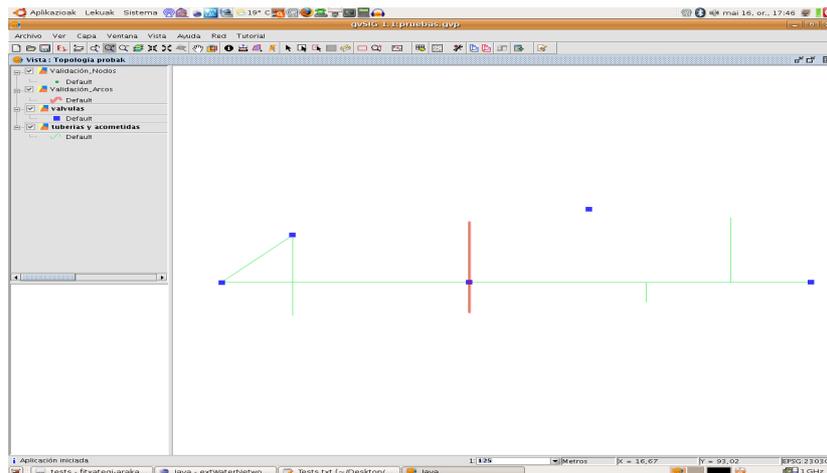
201. Regla "Not correctly valve-limited pipes on edges, except intakes". Valida que cada tubería intercepte un nodo en cada extremo y solo uno. En el ejemplo se puede ver que los dos arcos verticales cumplen la regla, uno de ellos porque no tiene ningún nodo en sus extremos y el otro porque sólo tiene un nodo en su extremo superior. La tubería horizontal no cumple la regla porque es correcta topológicamente (está limitada por nodos en ambos extremos), al igual que la tubería a 45°. Además, para una de las tuberías verticales, existe un nodo en la intersección de dos arcos (y no en los extremos de los arcos) que no provoca un falso positivo. Las acometidas (arcos que se conectan a lo largo de las tuberías) están exentas de cumplir la regla. En el ejemplo se observa que efectivamente, aunque no interceptan ningún nodo, no ofrecen un falso positivo.



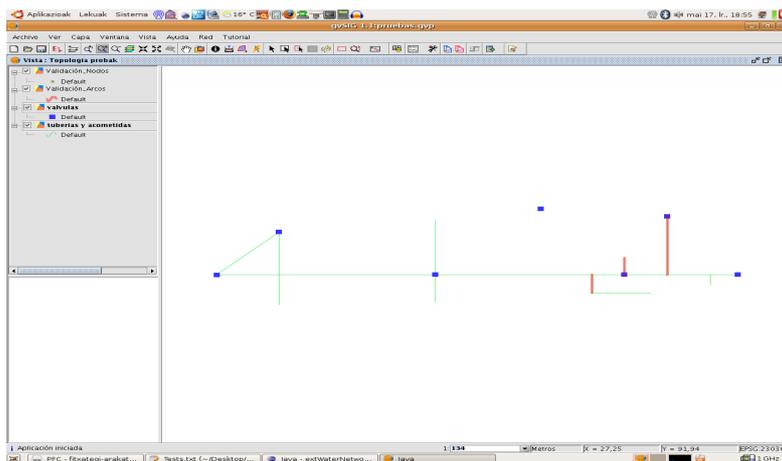
202. Regla "Isolated nodes". Localiza los nodos aislados de la red. En el ejemplo se observa que el nodo marcado en rojo es el que valida la regla y, por tanto, erróneo topológicamente.



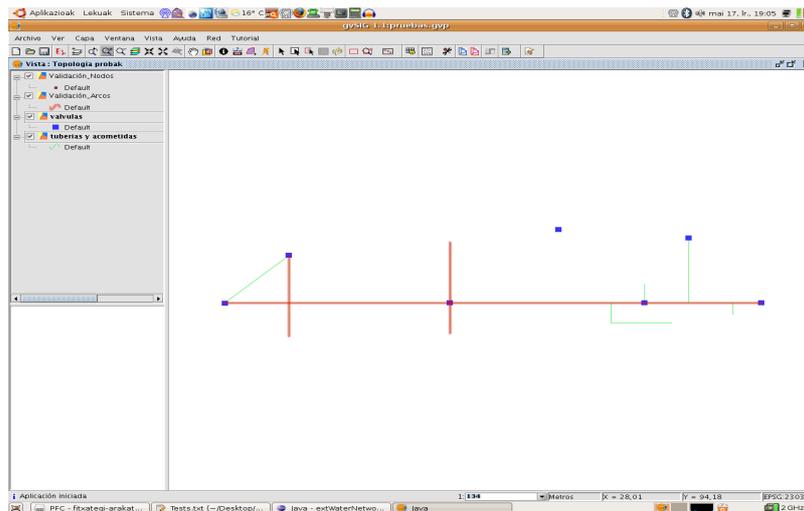
203. Regla "Not edge connected arcs". Localiza los arcos que no estén conectados con otros arcos en sus extremos, representarían tuberías o acometidas aisladas de la red, consideradas errores topológicos. En el ejemplo, se observa que hay una tubería que cruza a otra tubería (además hay un nodo en la unión entre ambas), pero como no está conectada con otro arco en ninguno de sus extremos, valida esta regla.



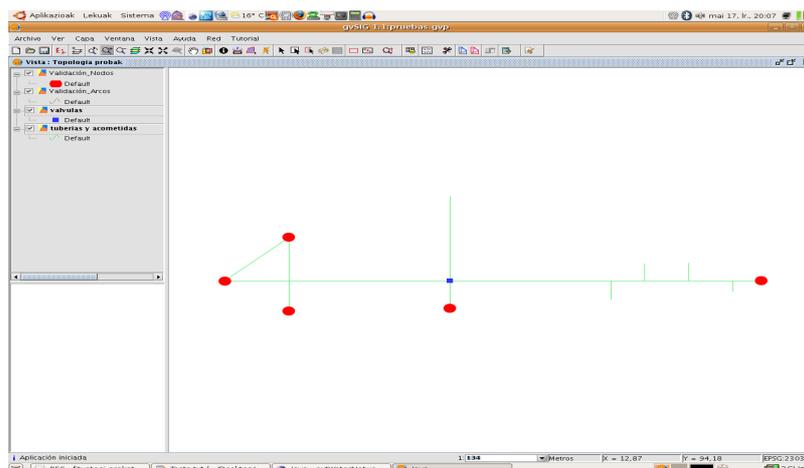
204. Regla "Not correct intakes". Localiza las acometidas erróneas. Las que interceptan algún nodo o algún otro arco aparte de la tubería a la que están unidas. En el ejemplo se observa que de las 4 acometidas dibujadas, solo una no cumple la regla (es correcta). Las otras 3, interceptan algún otro elemento además de la tubería a la que están unidas.



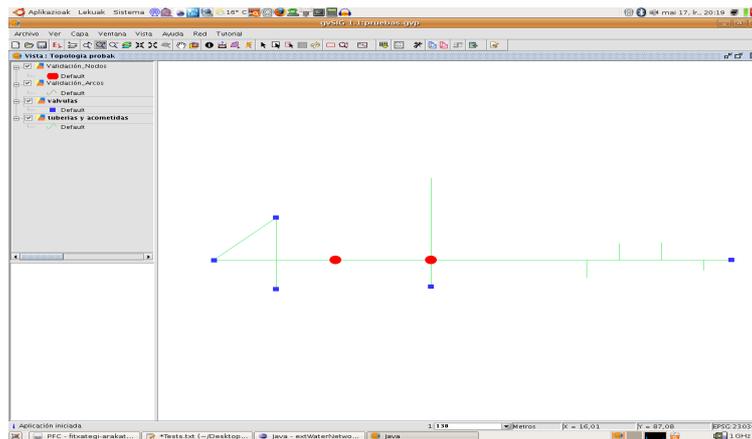
205. Regla "Crossings arcs" o, según el enunciado del proyecto, "Autocruce (intersección) de arcos". Localiza los cruces de arcos (intersección que no se da en los extremos), independientemente de que en el punto de cruce haya o no un nodo. El resultado correcto de la ejecución se muestra en el ejemplo en el que se marcan los arcos que intersectan con otro arco o más.



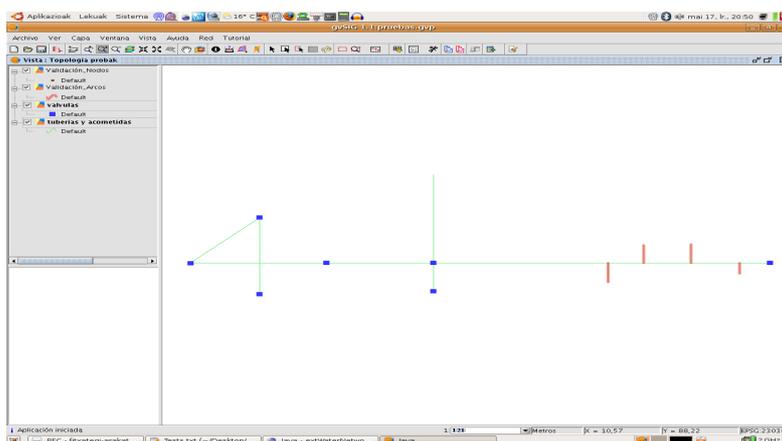
206. Regla "Limit nodes" o, según el enunciado del proyecto, "Nodo sobre inicio/fin de arco". Son los nodos que limitan cada tubería en sus extremos. En el ejemplo, todos excepto el nodo que se encuentra en la intersección entre las dos tuberías más largas, porque esta intersección es a lo largo de ambas tuberías y no en los extremos.



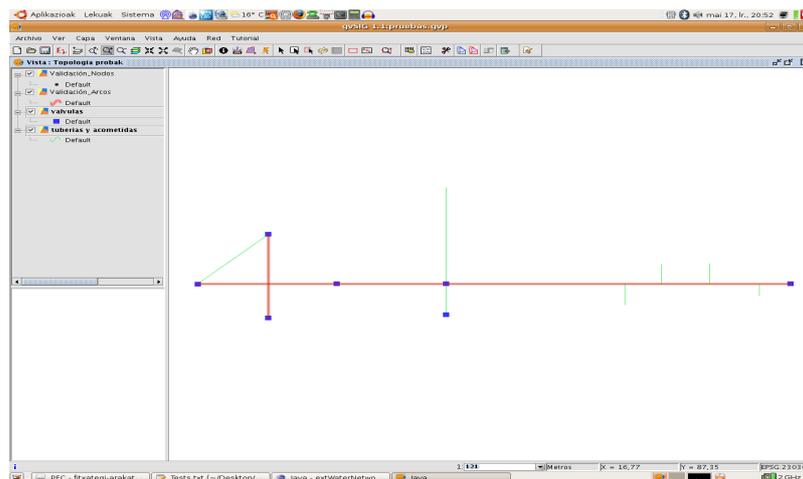
207. Regla "Link nodes" o, según el enunciado del proyecto, "Nodo sobre vértice de arco (Unión)". Son los nodos que sirven para unir dos o más tuberías y se encuentran a lo largo de las mismas y no en los extremos. Se observa en la figura, que sólo cumplen esta regla el nodo que se encuentra en la intersección de las dos tuberías más largas y el nodo que se encuentra a lo largo (sin unir ninguna otra tubería) de la tubería horizontal más larga.



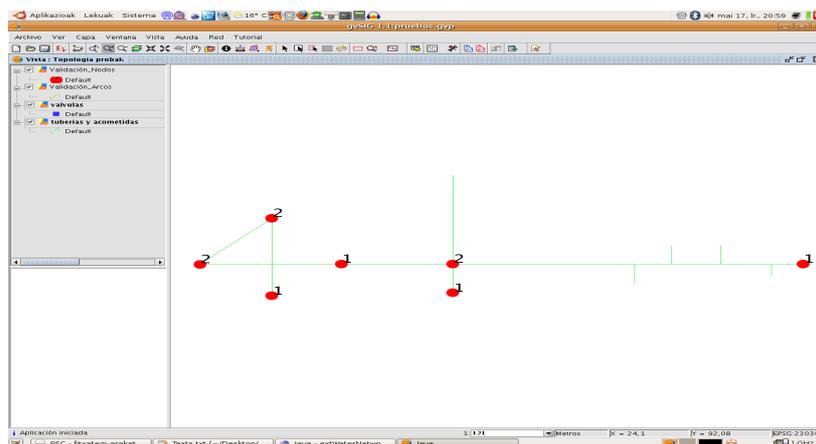
208. Regla "Intakes" o, según el enunciado del proyecto, "Punto inicial/final de arco sobre vértice de arco". Corresponden a las acometidas, que se unen a las tuberías a lo largo de las mismas por uno de sus extremos. Se observa en la figura que todas las acometidas dibujadas cumplen esta regla.



209. Regla "Crossing arcs without node" o, según el enunciado del proyecto, "Cruces entre arcos sin nodo". Lo cumplen los arcos que se cruzan a lo largo de los mismos (no en los extremos) y no hay un nodo en la intersección. En el ejemplo, se puede observar un sólo caso, porque en el otro cruce existente hay un nodo en la intersección.



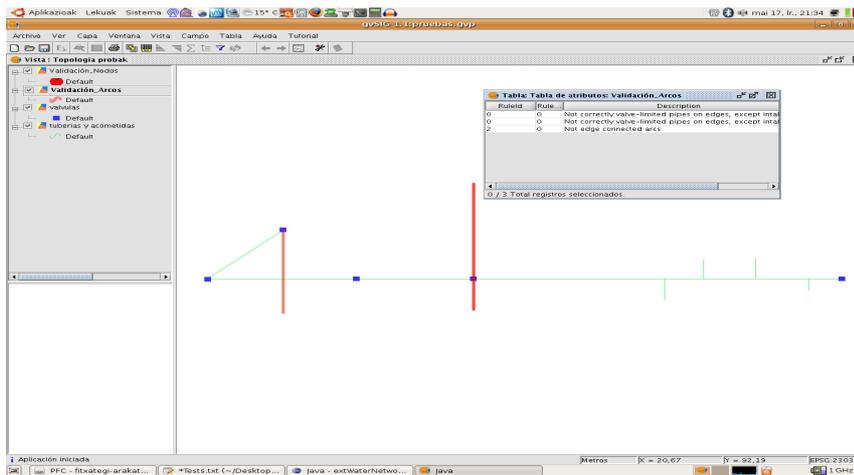
210. Regla "Node cardinality" o, según el enunciado del proyecto, "Cardinalidad de nodo/arcos". Esta regla se valida para cualquier nodo que intersecte o no uno o más arcos. El resultado de la regla ofrece la cardinalidad de esas intersecciones, como se puede observar en el ejemplo.



### 6.3. Pruebas de integración

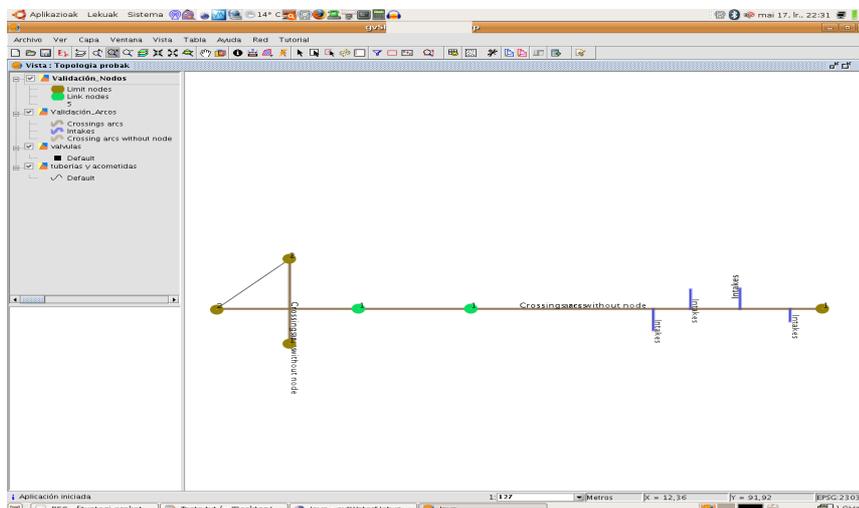
#### 6.3.1. Pruebas de integración usando los mismos datos de pruebas de las pruebas de unidad.

301. Al aplicar todas las reglas a la vez, el programa da prioridad a las definidas como Error topológico. Si se validan Errores topológicos se interrumpe la validación. En la figura, se observa que se han validado las reglas "Not correctly valve-limited pipes on edges, except intakes" y "Not edge connected arcs".



A continuación, se completa el dibujo, añadiendo el nodo que falta en la tubería vertical izquierda y eliminando la tubería vertical derecha que validaba los dos errores. Al repetir el proceso, ya no hay errores de validación por lo que se pasa a estudiar las reglas para hechos topológicos.

302. El resultado de la validación de los hechos topológicos se muestra en la siguiente figura. Se ha añadido una leyenda de colores (según el campo RuleId) que permite identificar más fácilmente qué regla se ha validado para cada elemento.

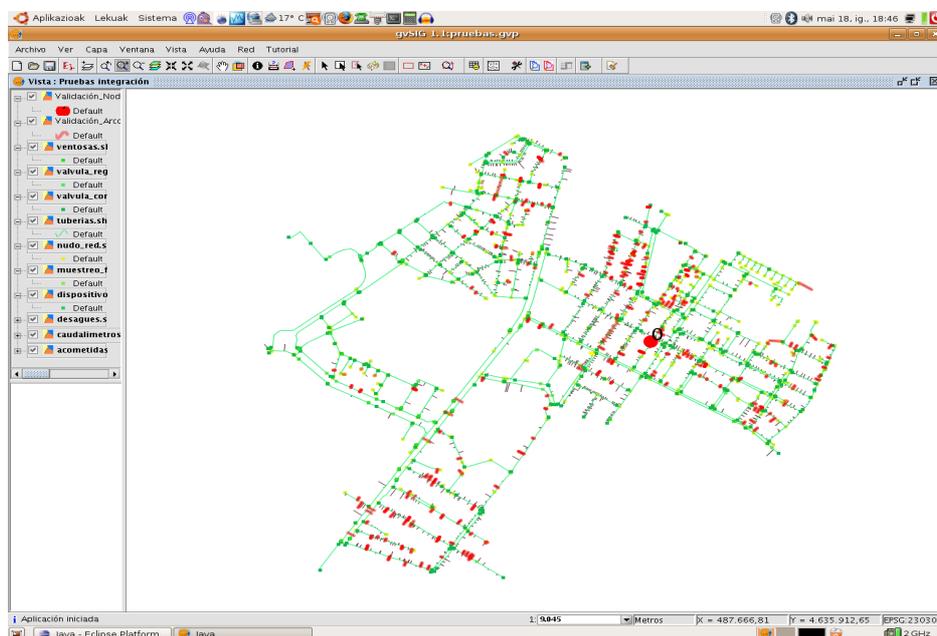


### 6.3.2. Pruebas de Integración sobre datos aportados por el consultor

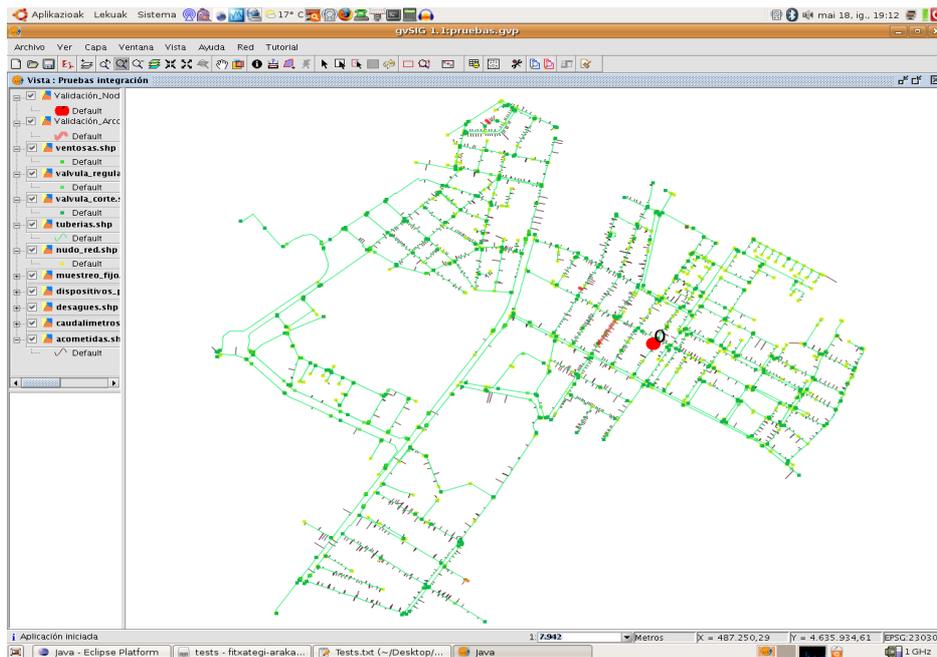
401. Problemas de rendimiento. Debido a la gran cantidad de geometrías (más de 4000) suministradas en los datos, se ha comprobado que la solución tiene graves problemas de rendimiento. Se ha estudiado el código, para poder optimizarlo, y se ha comprobado que el "cuello de botella" se encuentra en el método "calculateInterPoint" de la Subclase "Condition" en la clase "Rule". Se ha optimizado este método para que construya sólo las estructuras de memoria

estrictamente necesarias. Y también se ha optimizado el cálculo de las geometrías de intersección tipo PATH. Antes se usaban funciones de JTS. Se ha optimizado este cálculo usando funciones trigonométricas generales de la librería *java.Math*, para este caso concreto.

402. Una vez solucionado este tema se ejecuta de nuevo la validación y aparecen numerosos errores topológicos, como se puede observar en la figura (errores en rojo). La mayoría de ellos son debidos a que muchas de las acometidas del plano cumplen con la regla ERROR nº 3 (véase prueba unidad 204). Tras examinar varios de los errores se observa que, al dibujar el plano, muchas acometidas se han solapado (por estar dibujadas varios arcos uno sobre otro o con cruces entre los mismos). Esta regla se cumple cuando una acometida intercepta cualquier otro elemento (nodo o arco) a parte de la tubería a la que está conectada, por lo tanto es lógico que esto ocurra.

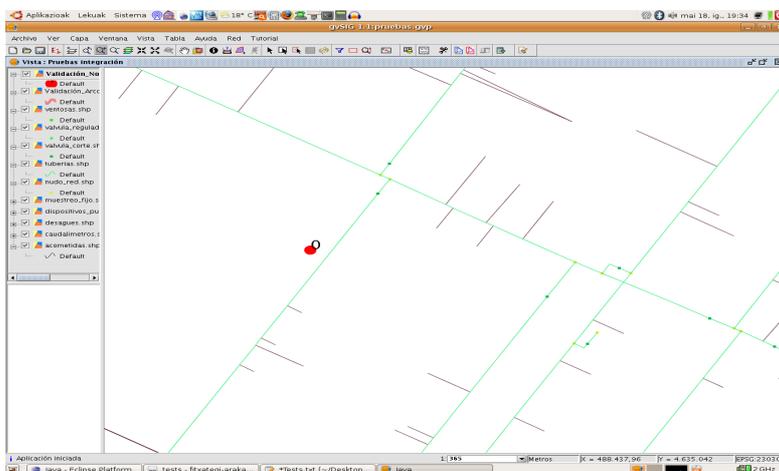


403. Se repite la ejecución, eliminando del testeo la regla de ERROR nº3, que daba problemas con las acometidas. Por tanto, en la fase de ERRORES topológicos, ya no se testeará que las acometidas sean estrictamente correctas. Se adjunta una captura de pantalla del resultado.



Los errores topológicos que aparecen son ahora los siguientes:

- Se cumple la regla "Isolated node" en un caso (ver figura). Es realmente un error topológico

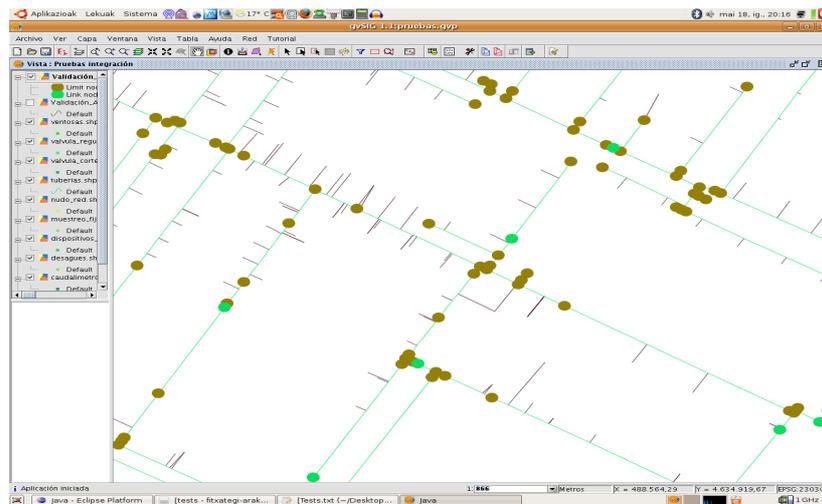


- Se cumple la regla "Not edge connected arcs" en dos casos (ver figura), debidos a acometidas mal dibujadas, porque terminan más allá de la intersección con la tubería y no son identificadas como acometidas sino como tuberías que cruzan la tubería principal.

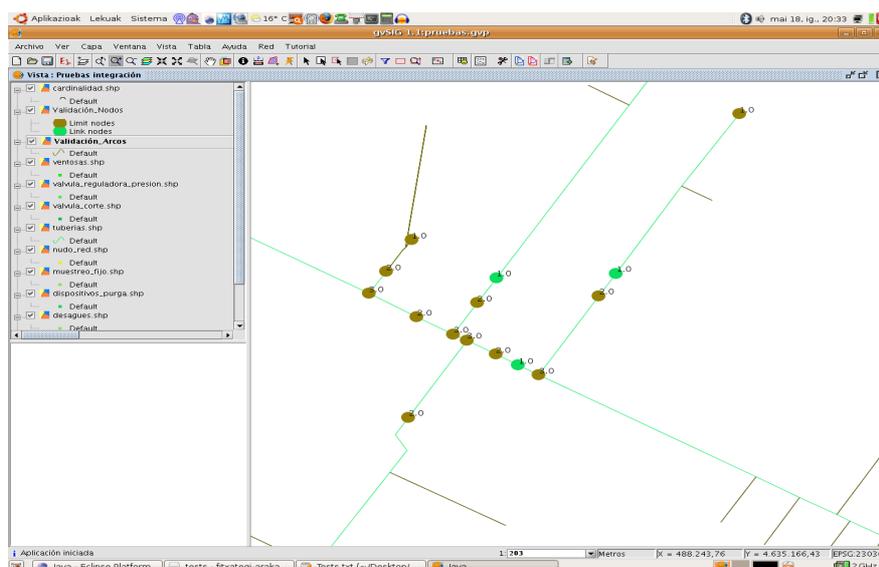


404. Se han corregido manualmente todos estos errores y se ha vuelto a ejecutar la validación. Ahora ya no aparecen ERRORES topológicos y el programa muestra como resultado las apariciones de los 6 HECHOS topológicos. Se ha añadido una simbología de colores para mostrar los resultados de manera más visual.

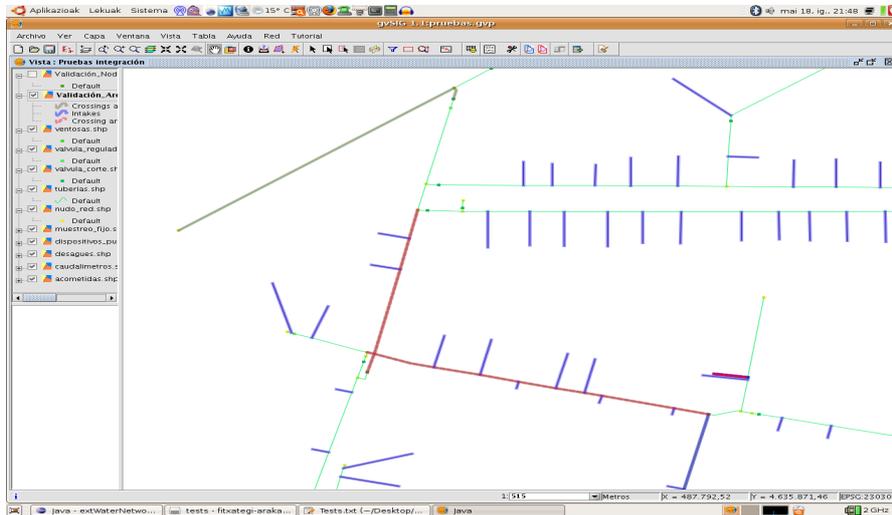
- En esta primera figura se destacan los hechos topológicos sobre objetos tipo nodo detectados "Limit nodes" y "Link nodes". Los primeros son nodos que unen diferentes tuberías por sus extremos (en marrón), mientras que los segundos, son nodos a lo largo de las tuberías (en verde).



- En la siguiente figura se muestra la cardinalidad nodo/arco calculada por la última regla. Para no mezclar en la imagen las diferentes cardinalidades (de diferentes reglas), se han exportado los resultados de la regla n°5 a un nuevo SHP(cardinalidad.shp), y se ha cargado en una capa con simbología especial para mostrar ese resultado.



- En la siguiente figura se indican, mediante un código de colores, las acometidas (en azul) y los arcos que se cruzan (en rojo y gris). Los cruces de arcos aparecen en diferente color, rojo o gris, cuando en la intersección hay un nodo (porque los cruces sin nodo validan una regla adicional). Para minimizar los falsos positivos al considerar que los nodos se cruzan (para evitar falsos positivos con algunas acometidas) se ha aumentado la tolerancia de 0.1 a 0.5.



## Conclusiones y recomendaciones

El proyecto ha cumplido sus objetivos, porque ha permitido realizar un análisis de requisitos sólido, basado en un estudio preliminar y plasmar ese análisis en una solución software, a modo de extensión para gvSIG.

### Incidencias y soluciones aportadas

En primer lugar, los trabajos de documentación de la Tarea 1 supusieron una carga de trabajo algo superior a la esperada, lo que provocó un pequeño retraso. El hecho de que haya que desarrollar la solución sobre una plataforma ya existente (gvSIG) ha requerido adquirir cierto nivel de conocimiento de la misma antes de poder definir bien las funcionalidades de la solución a desarrollar en el proyecto. De todas maneras, este trabajo de documentación aportó la ventaja de que las funcionalidades a desarrollar se concretaran más fácilmente y ayudó a recuperar ese tiempo. Se simplificaron los trabajos posteriores, fundamentalmente durante la tarea 4 (Análisis).

Por otro lado, en un principio se decidió posponer la realización de la tarea 8 (*Elaboración del producto instalable*) a la siguiente fase del proyecto. Por un lado, por problemas de plazos para terminar correctamente todas las pruebas (tarea 7), pero por otro lado por no encontrarse documentación ni ejemplos para realizar esta tarea, al contrario de lo que ocurrió con otras tecnologías que se han usado durante la implementación (extensiones de tipo geoproceto, JTS, etc...). Este hecho obligó a realizar un estudio más detallado de esa tecnología del que estaba previsto en un principio. De todas maneras, al realizar este estudio posteriormente, con la ayuda del consultor, el producto instalable estuvo listo con sólo unos pocos días de retraso.

Durante las pruebas de integración sobre los datos aportados por el consultor se detectó un grave problema de rendimiento en la aplicación, debido a la gran cantidad de geometrías (más de 4000) suministradas en los datos. Se estudió el código, para poder optimizarlo, y se comprobó que el "cuello de botella" se encontraba en el método "calculateInterPoint" de la Subclase "Condition" en la clase "Rule". Se optimizó este método para que construyera solamente las estructuras de memoria estrictamente necesarias. Y también se optimizó el cálculo de las geometrías de intersección tipo PATH. Antes se usaban funciones de JTS. Se optimizó este cálculo dejando a un lado la librería JTS y usando funciones trigonométricas generales de la librería *java.Math*, para este caso concreto.

Por último, cabe destacar que el método para repartir el trabajo durante el fin de semana y no durante la semana, no ha podido llevarse a cabo de la manera prevista. Ha resultado muy complicado dedicar los fines de semana íntegramente al estudio, por razones familiares. Esto me ha obligado a trabajar casi diariamente de manera irregular aprovechando los momentos libres y pedir ayuda a la familia en algunos momentos.

### Contribución al proyecto gvSIG

En cuanto a las herramientas de validación topológica en el entorno gvSIG, este proyecto aporta algunos puntos a tener en consideración aunque no es, obviamente, la herramienta que gvSIG necesita de manera definitiva. Esta es una solución orientada a las redes de agua.

La solución software desarrollada se concentra en las topologías tipo red, y en las relaciones de *intersección* entre objetos geográficos. Una extensión para validación topológica para gvSIG, debería aportar también soporte para objetos de tipo *área* (no solo objetos *lineales* y *puntuales*). Por otro lado, existen relaciones topológicas que también habría que tener en cuenta en una validación topológica general, como la *inclusión* o la *distancia*.

La principal contribución de este proyecto a la plataforma gvSIG es el modelo de datos propuesto

en la representación de reglas topológicas. Es un modelo de datos bastante extensible y que podría dar también soporte otras funcionalidades mencionadas anteriormente (objetos tipo área, relaciones topológicas de inclusión, ...) con unos pequeños cambios.

### **Recomendaciones para futuros desarrollos**

Supongamos que a partir del modelo de datos propuesto en el análisis, se añade a la estructura de las condiciones de las reglas otro nuevo campo que representase la relación topológica a estudiar: *intersección*, *distancia*, *inclusión*, etc. y además se añaden también las *áreas* como tipos de objeto soportados.

Así, por ejemplo, se podrían definir reglas de hechos topológicos para diferenciar las regiones de los estados en un mapa político, suponiendo que las *áreas* que representan a estados deban **contener** una o más regiones y las *áreas* que representan a las regiones no deban **contener** ningún otro área para considerarse como tales. También podría definirse una regla que validase si los *objetos puntuales* de un mapa urbano, que representan bocas de riego, se encuentran a una *distancia* mínima dada, etc.

Por otro lado, el proyecto ha dejado sin diseñar e implementar los casos de uso *Configurar Parámetros de Validación* y *Configurar Simbología de Resultado*.

El caso de uso *Configurar Parámetros de Validación* podría ser de gran utilidad para evitar al usuario tener que editar el fichero *text.properties* mediante un editor de texto independiente, sin ninguna ayuda. Sería de gran ayuda, principalmente, para posibilitar al usuario las reglas que desea validar en cada momento. Por otro lado, el hecho de que las reglas se almacenen en ficheros externos resulta práctico, aunque la solución debería facilitar la carga y grabación de estos ficheros durante la ejecución del programa. Además, debería de aportar un "editor de reglas" que permitiera la edición de las mismas, en pantalla, de manera asistida. El usuario podría, así, cargar unas reglas topológicas por defecto, personalizarlas en pantalla según sus necesidades y grabarlas para poder ser usadas en otra ocasión, además de realizar la validación topológica como tal a modo de geoproceto.

En cuanto al caso de uso *Configurar Simbología de Resultado*, gvSIG ya tiene casi resuelta esta necesidad a través de la herramienta para *cargar/grabar Leyendas* de capas. El problema es que la carga/grabación de leyendas es manual y no se encuentra integrada en el geoproceto. Este caso de uso debería guiar al usuario en esta tarea desde el propio geoproceto.

## **Agradecimientos**

En este apartado de agradecimientos quisiera hacer mención, principalmente, a mi familia; que tanto me ha apoyado a lo largo del proyecto y también en otros momentos de la carrera. La dedicación a los estudios supone un sacrificio, no solo para el estudiante, sino también para su familia, especialmente cuando ha de dedicarse al estudio el poco tiempo libre que deja la vida laboral.

Por otro lado, quisiera destacar también la labor de mi consultor en la UOC, Jesús De Diego. Sus consejos han sido de gran ayuda. También ha sido muy importante que me haya ofrecido la posibilidad de contrastar su opinión sobre los materiales a entregar antes de cada entrega definitiva.

## Lista de referencias

A continuación se resumen las referencias a otros textos que se han realizado 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/fileadmin/conselleria/images/Documentacion/descargas/ficheros/11/gvSIG-1\\_1-src.zip](http://www.gvsig.gva.es/fileadmin/conselleria/images/Documentacion/descargas/ficheros/11/gvSIG-1_1-src.zip)
- Tutorial de gvSIG 1.0:  
[http://www.gvsig.gva.es/fileadmin/conselleria/images/Documentacion/cursos/Tutorial-gvSIG-1\\_0.pdf](http://www.gvsig.gva.es/fileadmin/conselleria/images/Documentacion/cursos/Tutorial-gvSIG-1_0.pdf)
- Información sobre la versión en desarrollo "FMAP->TOPOLOGIA" para gvSIG  
<https://gvsig.org/web/docdev/docs/desarrollo/FMap/topologia>

VIVID SOLUTIONS. Java Topology Suite

- Información general: <http://www.vividsolutions.com/jts/jtshome.htm>

Junta de Extremadura. Consejería de Industria, Energía y Medio ambiente. Web de SEXTANTE

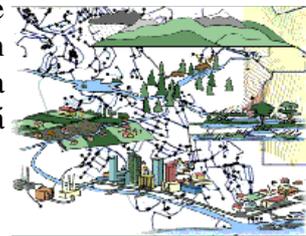
- Información general: <http://www.sextantegis.com/>

Referencias Wikipedia

- Definición de tubería: <http://es.wikipedia.org/wiki/Tuber%C3%ADa>
- Definición de acometida: <http://es.wikipedia.org/wiki/Acometida>
- Definición de válvula: <http://es.wikipedia.org/wiki/V%C3%A1lvula>

## ANEXO I. Manual de usuario de la extensión *WaterNetwork* para gvSIG

La extensión *WaterNetwork* para gvSIG es una herramienta que añade nuevas funcionalidades a esta plataforma de Sistemas de Información Geográfica, con el objeto de realizar validaciones en redes de aguas. Una vez instalada, la herramienta de validación topológica se encontrará integrada en la plataforma a modo de **Geoproceso**.



### 1. Requisitos de la solución software

La solución es una extensión sobre la plataforma *gvSIG Desktop*. Esta extensión no incluye nuevos requisitos de hardware o software adicionales a los indicados para la plataforma mencionada. La versión de gvSIG que deberá instalarse previamente es:

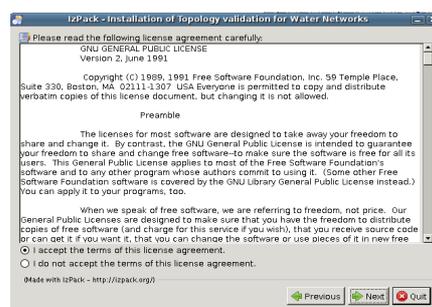
- **gvSIG 1.1** (<http://www.gvsig.gva.es/index.php?id=gvsig11&L=0&K=1%2C>)

### 2. Instalación de la extensión *WaterNetwork* para gvSIG

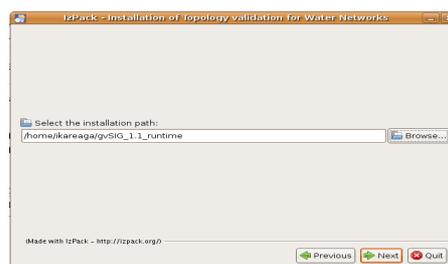
Una vez instalada la plataforma gvSIG, ya se puede instalar la extensión *WaterNetwork*. El usuario deberá ejecutar el programa de instalación multiplataforma **WaterNetworkInstall.jar** mediante el siguiente comando:

```
$ java -jar WaterNetworkInstall.jar
```

El programa de instalación mostrará un asistente que guiará al usuario a través del proceso de instalación. Primero mostrará una pantalla de presentación, a continuación mostrará en pantalla las notas de la versión y después deberá de aceptarse el acuerdo de licencia



Una vez aceptada la licencia, deberá introducirse el directorio base en el cual se instalará la solución. Es necesario que este directorio base coincida con el directorio donde está instalada la plataforma gvSIG.



El instalador advertirá que esa carpeta ya existe (ya que gvSIG está instalado previamente). A continuación, se podrá elegir entre los componentes que se quieren instalar:

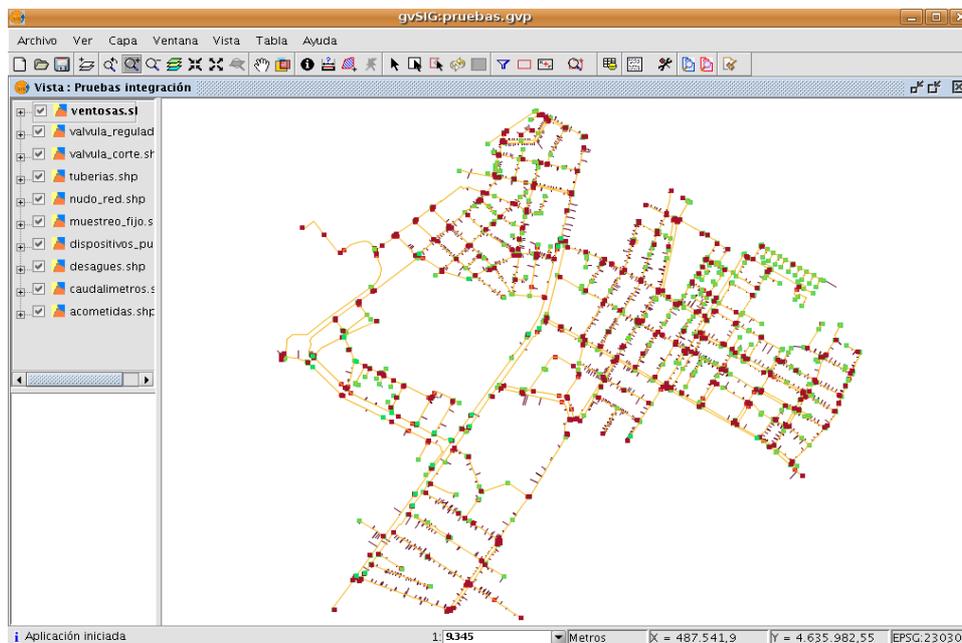
- Base: Los ejecutables de la solución
- Docs: Documentación del software
- Sources: Código fuente de la solución software

Una vez elegidos los elementos a instalar el programa de instalación procederá a copiar los elementos necesarios y la instalación concluirá.

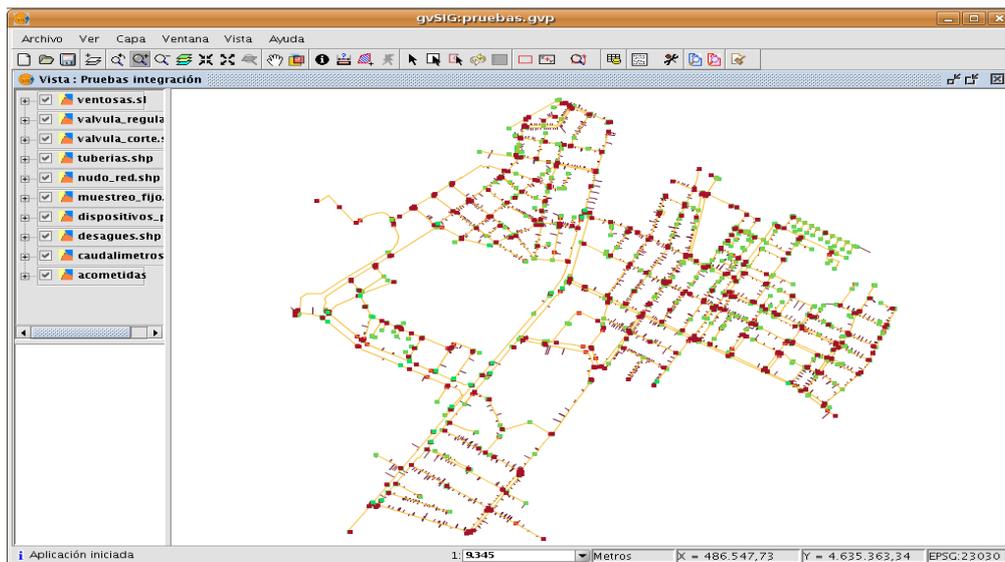
### 3. Selección de capas a validar y ejecución del geoproceso

Una vez instalada la extensión para gvSIG, al acceder a la aplicación, el usuario podrá encontrar un nuevo Geoproceso. La ejecución del mismo se deberá realizar siguiendo los siguientes pasos:

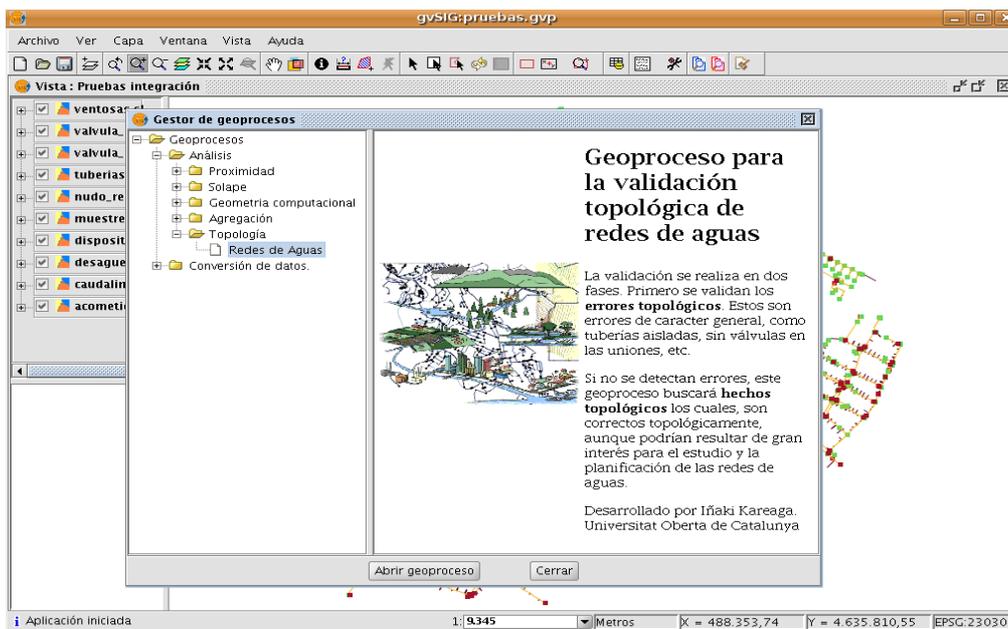
1. Cargar en la vista deseada las capas vectoriales sobre las que se quiere realizar la validación topológica. Consultar el manual de gvSIG para más información sobre la carga de capas vectoriales en una vista.



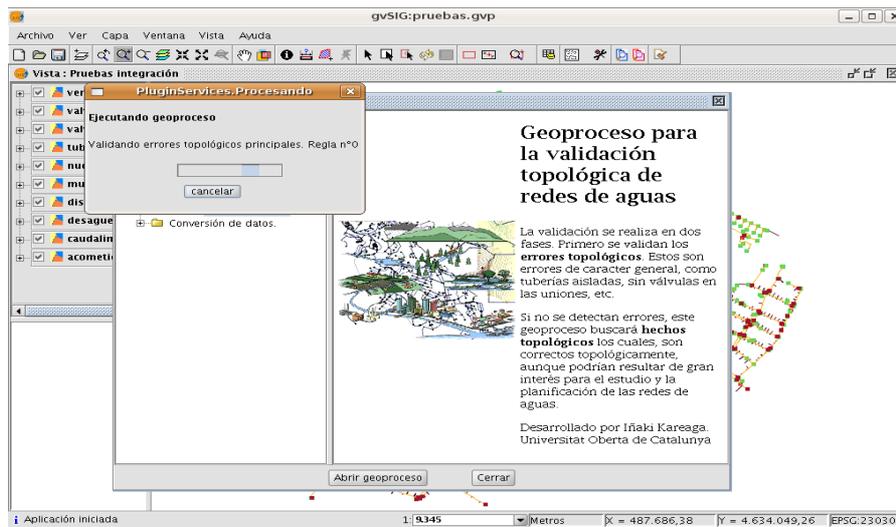
2. Seleccionar las capas a validar (haciendo CTRL+clic sobre las capas deseadas en lista de capas). Deben seleccionarse una o más capas válidas. Las capas válidas son las que contienen objetos vectoriales puntuales o lineales (no son válidas, por tanto, las capas de polígonos).



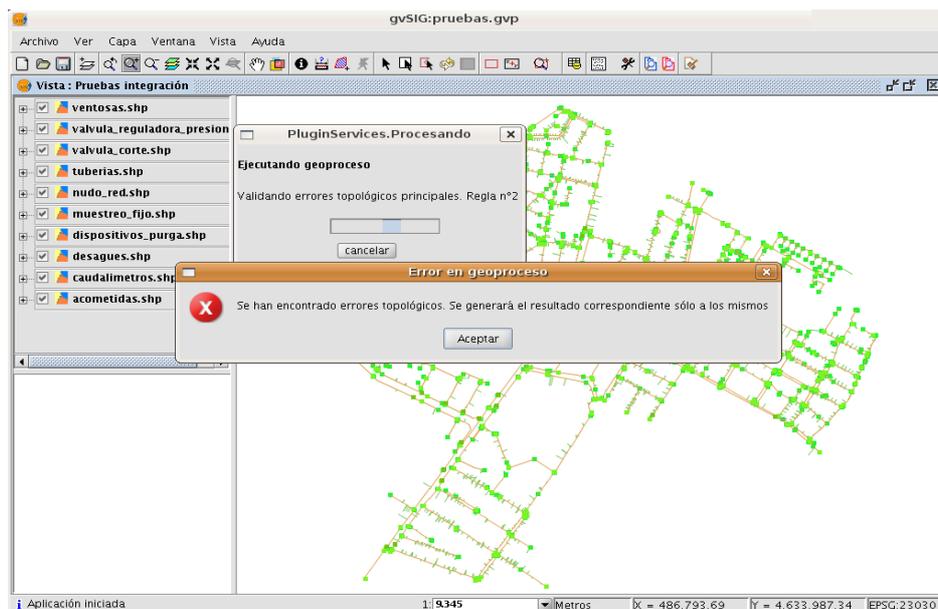
### 3. Ejecutar el Geoproceso "Análisis->Topología->Redes de Aguas"



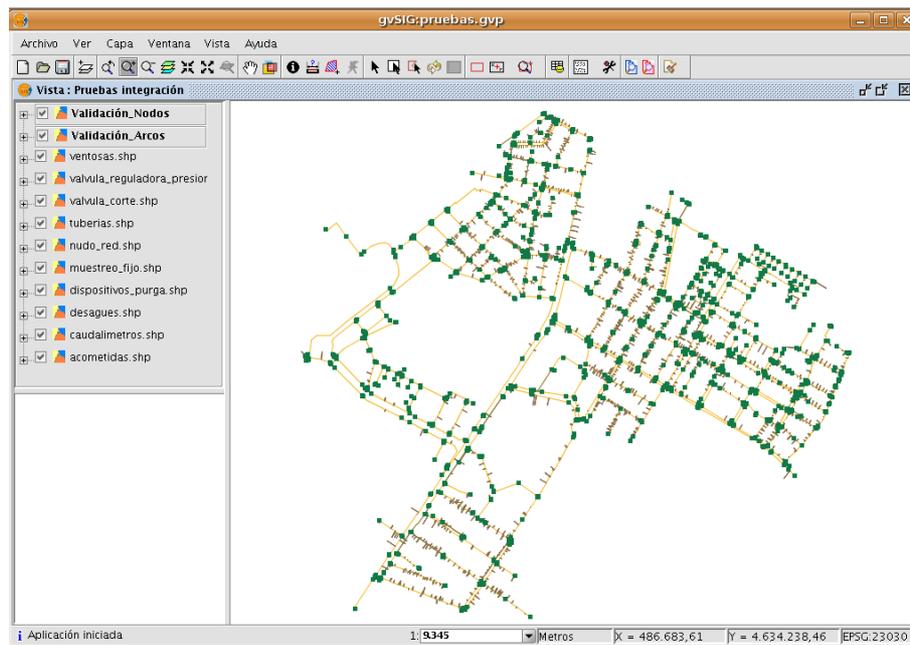
4. "Abrir el geoproceso" y ejecutarlo confirmando la opción sobre el botón "Aceptar". Si se ha seleccionado una o más capas válidas, el programa comenzará las tareas de validación informando de la regla que está validando en cada momento. Primero se evaluarán las reglas topológicas correspondientes a errores topológicos.



5. En caso de que alguna regla topológica correspondiente a errores topológicos se cumpla, el programa informa de este hecho al usuario y detiene la validación después de la última regla de este tipo.



6. En caso de que no hubiera ningún error topológico, el programa pasará a validar las reglas relativas a hechos topológicos. Como resultado, se obtendrán dos nuevas capas: "Validación\_Arcos" y "Validación\_Nodos".



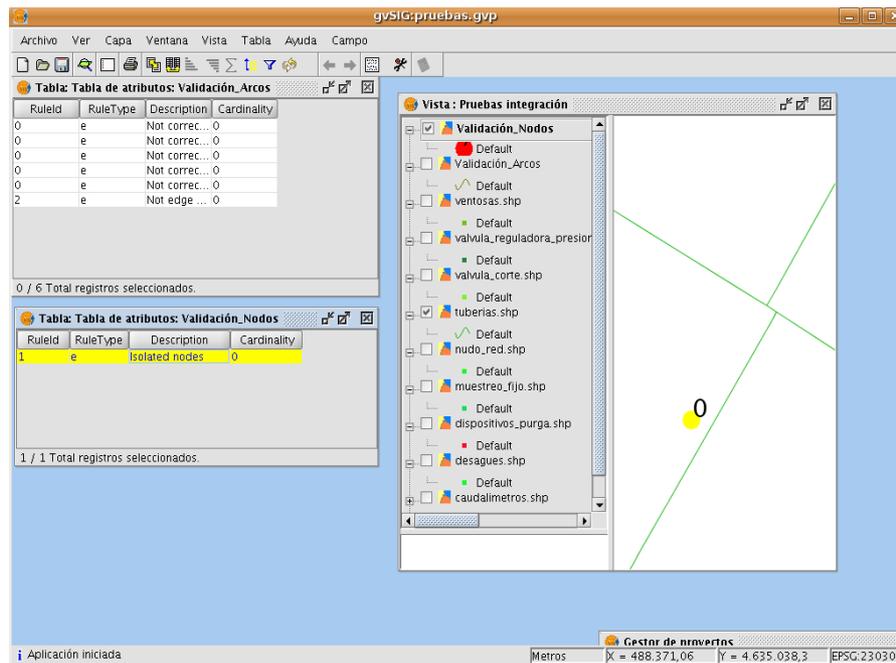
#### 4. Interpretación de los resultados de la validación

Las dos capas generadas por el geoproceso contienen copias de los objetos originales que han validado alguna regla. La capa *Validación\_nodos* agrupa los objetos puntuales y la capa *Validación\_arcos*, los objetos lineales.

Los objetos creados tendrán asociada una tabla de valores con los siguientes campos:

- RuleId: Nº de regla que ha validado el objeto clonado en la capa de resultados
- RuleType: Tipo de regla. "e" para Errores topológicos y "f" para Hechos topológicos.
- Description: Descripción textual de la regla
- Cardinality: Cardinalidad de intersecciones ocurridas en la validación de esa regla

En el ejemplo de la siguiente figura se puede observar que el nodo marcado ha validado la regla nº1 de tipo "error topológico" (e). Esta regla se valida para los nodos que no interceptan ningún arco. La cardinalidad de intersecciones para la validación de la regla ha sido, por tanto, 0 (lo que ha provocado que la regla se valide).



## 5. Modificación de los parámetros de validación y simbología de los resultados

En esta versión de la solución no se ha implementado la posibilidad de modificar los parámetros de validación a través del interfase de usuario. Sin embargo, estos parámetros pueden modificarse a través de la edición del fichero de parámetros de la extensión.

```
[gvSIG_home]/bin/gvSIG/extensiones/com.iver.cit.gvsig.waterNetwork/text.properties
```

Este fichero se podrá editar mediante un editor de texto estándar, y a continuación reiniciar gvSIG para que los cambios tengan efecto. A continuación, se describen los parámetros principales del fichero y la manera de modificarlos:

### Parámetros generales (*param*):

```
param.tolerance=0.5
param.rules_error=3
param.rules_fact=6
```

El parámetro *tolerance* indica la distancia máxima a la que se deberán encontrar dos objetos para que la validación considere que se interceptan (0,5 unidades en el ejemplo). Los parámetros *rules\_error* y *rules\_fact* indican al programa el número de reglas que se han descrito para cada tipo; Errores topológicos y Hechos topológicos respectivamente (3 reglas de errores topológicos y 6 reglas de hechos topológicos en el ejemplo).

### Definición de reglas topológicas (*rule*):

Cada regla topológica está formada por una o más condiciones, que se asocian entre ellas mediante operadores AND o OR (con precedencia de operadores según ese orden). La definición de las reglas sigue la siguiente sintaxis:

```
rule_[tipo de regla][nº de regla]=[Descripción de la regla];[condición1];[condición2];...
```

también puede expresarse en varias líneas (indicando \ al final de cada una)

```
rule_[tipo de regla][n° de regla]=[Descripción de la regla];\  
[condición1];\  
[condición2];\  
...
```

El tipo de regla podrá ser "e" ó "f" (Error o Hecho topológico, respectivamente) y cada condición se expresa a través de 7 atributos:

```
[operador],[TipoObj],[interceptaUn],[minCard],[maxCard],[ptoIntersBase],[ptoIntersSec];
```

- **operador:** Operador AND/OR a aplicar en el cálculo lógico con respecto a las anteriores condiciones de una misma regla. Se aplica la precedencia de operadores en el orden indicado.
- **TipoObj:** Tipo de objetos a los que se aplica la regla: ARC (arco), NODE (nodo) o ANY (cualquiera de ambos)
- **interceptaUn:** Tipo de objeto respecto al que se estudia la intersección con respecto del indicado en el atributo anterior.
- **min/maxCard:** 0 indica que el objeto que se esta validando no debe interceptar objetos de la clase indicada en *interceptaUn* para que la condición se cumpla. Un numero positivo indica la cardinalidad máxima o mínima en la intersección para validar la condición. Por último, un número negativo indica que no hay límite de cardinalidad mínima o máxima.
- **ptoIntersBase:** Punto de intersección en el objeto base. ANY indica que se valida la intersección en cualquier punto, BEGIN únicamente en el vértice extremo inicial, PATH valida la intersección en un vértice a lo largo del objeto (y no en los extremos), END únicamente en el vértice extremo final y LAST indica el último vértice de intersección validado por una condición anterior en la misma regla.
- **ptoIntersSec:** Punto de intersección en el objeto interceptado. La posición del punto se expresará como en el atributo anterior.

Por ejemplo, si tomamos la siguiente regla:

```
rule.e1=Isolated nodes;\  
OR,NODE,ARC,0,0,ANY,ANY;
```

Estaríamos ante la regla nº1 de las reglas de tipo "Error topológico" que se describe con el texto "Isolated nodes". Esta regla trata de validar todos los objetos de tipo nodo (*TipoObj=NODE*) que intercepten 0 (*minCard=maxCard=0*) arcos (*interceptaUn=ARC*). El punto de intersección no debiera tenerse en cuenta ni en el nodo que se está evaluando ni en los posibles arcos que interceptase (*ptoIntersBase=ANY, ptoIntersSec=ANY*).

### Simbología de resultados:

En cuanto a la simbología de resultados, estos pueden modificarse manualmente a través de la opción *Propiedades* de la capa, que aporta gvSIG. Sin embargo, junto a la documentación de la solución se han incluido varias "Leyendas" de simbología de resultados con el objeto de facilitar al usuario la definición de los mismos. Las leyendas aportadas son las siguientes:

- LeyendaNodosERRORES.gvl: Optimizada para mostrar errores topológicos en la capa Validación\_nodos
- LeyendaNodosHECHOS.gvl: Optimizada para mostrar hechos topológicos en la capa Validación\_nodos
- LeyendaArcosCARDINALIDAD.gvl: Optimizada para mostrar hechos topológicos en los que la cardinalidad de intersecciones validada por la regla deba ser impresa a modo de

etiqueta del objeto geográfico, en la capa Validación\_nodos

- LeyendaArcosERRORES.gvl: Optimizada para mostrar errores topológicos en la capa Validación\_arcos
- LeyendaArcosHECHOS.gvl: Optimizada para mostrar hechos topológicos en la capa Validación\_arcos

Consultar el manual de gvSIG para más información sobre el uso de Leyendas de simbología de capas.

## ANEXO II. Procedimiento seguido para la instalación del entorno de desarrollo

Primeramente se deben descargar los fuentes correspondientes a la versión de 1.1 de gvSIG y también los correspondientes al Piloto de redes (que puede ser interesante tener instalado como ejemplo).

Tras descomprimir las fuentes de gvSIG, se han seguido los pasos del fichero *Leeme.txt* de la carpeta raíz. Siguiendo estos pasos, se han descargado también la versión recomendada del JDK y el entorno de desarrollo Eclipse. En este último caso no se recomienda una versión específica. Se ha optado por la última versión estable (Eclipse 3.3.2) adaptada para desarrolladores Java (con los plugins recomendados para esta tarea). En todos los casos se han descargado los builds correspondientes a Linux i568 que es el sistema operativo donde se realizarán los trabajos de desarrollo del proyecto (sobre la distribución Ubuntu 7.10). Se han instalado estas dos aplicaciones en una carpeta con acceso de lectura y escritura para el usuario actual en el sistema.

En el fichero *Leeme.txt* mencionado se detallan los pasos para realizar la primera compilación automática en Eclipse y la ejecución del programa. Se han seguido los pasos y todo a funcionado sin problemas.

En cuanto al Piloto de Redes, incorpora también un fichero *Leeme* con los pasos a seguir para importar el proyecto junto a los fuentes de la versión base gvSIG 1.1. Se han seguido los pasos y todo a funcionado sin problemas.

Antes de dar por concluida esta tarea, se ha realizado el tutorial recomendado en el enunciado, mediante el cual, se explica cómo se debe desarrollar una extensión para gvSIG desde 0. El tutorial se ha concluido con éxito.

Los principales aspectos aprendidos mediante este tutorial, son los siguientes:

- Los elementos y librerías necesarios para añadir un proyecto de Eclipse al desarrollo de gvSIG, también los comandos para *ant* que realizan la compilación y copia de las clases a las carpetas adecuadas.
- La edición del fichero *config.xml* donde se indican al gestor de ventanas (*andami*) cuales son las extensiones a incluir y la manera en la que serán accesibles desde el programa principal (menú, toolbar, ...).
- La internacionalización del programa
- Dibujar objetos geográficos y asignarles valores a sus atributos asociados usando la librería *Fmap*.
- A través de la clase *MapControl* recoger la interacción del usuario con el mapa, programar una clase *listener* y asociarla al mapa de manera que se ejecuten acciones en base a esta interacción.
- Selección de objetos geográficos de una capa en base a criterios de posicionamiento de los mismos.
- Obtener atributos temáticos a partir de la selección anterior y visualizarlos a través de un *JPanel*.

## ANEXO III. Referencias al paquete de instalación, código fuente y a los datos de prueba

Los elementos que se enumeran a continuación se han empaquetado en un fichero denominado *ikareaga\_Datos.zip*:

- **leeme.txt**: Lista de contenidos del fichero *ikareaga\_Datos.zip*.
- **WaterNetworkInstall.jar**: Programa de instalación del software (incluye la opción de instalar también el código fuente y documentación)
- **ManualUsuario.pdf**: Manual de usuario con las instrucciones de instalación y manejo del software.
- **extWaterNetwork/**: Código fuente del software que incluye documentación en formato Javadoc . Contiene los descriptores necesarios para integrarse a modo de proyecto en la plataforma de desarrollo *Eclipse*. Los contenidos del directorio son los siguientes:
  - **extWaterNetwork/build.xml**: Fichero de comandos ANT para compilar crear la extensión para gvSIG a partir del código fuente
  - **extWaterNetwork/bin/**: Clases JAVA compiladas
  - **extWaterNetwork/config/about.html**: Contiene información general sobre la extensión desarrollada, para ser mostrada en el menú de ayuda de gvSIG.
  - **extWaterNetwork/config/config.xml**: Fichero con parámetros para ANDAMI, que describe la integración de la solución software en la plataforma gvSIG.
  - **extWaterNetwork/config/text.properties**: Fichero con los literales del interfase de usuario en inglés y los parámetros para la validación topológica
  - **extWaterNetwork/config/text\_es.properties** y **config/text\_eu.properties**: internacionalización de los literales del interfase de usuario en castellano y euskera.
  - **extWaterNetwork/config/docs/**: Directorio con los ficheros que contienen la descripción del geoprocso en inglés (por defecto), castellano y euskera y la imagen adjunta.
  - **extWaterNetwork/doc/**: Documentación del código en formato JAVADOC
  - **extWaterNetwork/images/**: Directorio con imágenes adicionales
  - **extWaterNetwork/src/**: Código fuente en JAVA
- **datosPrueba\_ErroresTopologicos/**: Directorio que contiene los datos de prueba para probar las reglas topológicas de tipo "Errores topológicos"
- **datosPrueba\_HechosTopologicos/**: Directorio que contiene los datos de prueba para probar las reglas topológicas de tipo "Hechos topológicos"
- **LeyendasResultados/**: Directorio que contiene los ficheros con leyendas para configurar la simbología de las capas de resultados

## ANEXO IV. Bibliografía

Bosque Sendra, Joaquín (2000) Sistemas de Información Geográfica. Ed. Rialp. S.A. Madrid.

IVER Tecnologías de la Información (2007). Manuales y tutoriales de gvSIG. En: <http://www.gvsig.gva.es/index.php?id=manuales>

IVER Tecnologías de la información (2008). Documentación disponible para desarrolladores. En: <https://gvsig.org/web/docdev>

VIVID Solutions. JTS Topology Suite - Version 1.4 . Developer s Guide & Tecnical Specifications. En: <http://www.vividsolutions.com/jts/jtshome.htm>

Equipo SEXTANTE (2008). Conceptos generales de SEXTANTE Edición v0.52 . En: [http://sextantegis-docs.googlecode.com/files/ConceptosSEXTANTE\\_v052.pdf](http://sextantegis-docs.googlecode.com/files/ConceptosSEXTANTE_v052.pdf)

Eclipse - An open development platform. En: [www.eclipse.org](http://www.eclipse.org)