

Herramientas de monitorización con capacidades de decisión geográficas.

Miguel García Coya⁽¹⁾ y José Ángel Chico Monzón⁽²⁾

⁽¹⁾ Analista Programador SIC Ingenieros, C/ Misterios, 65. 28027 Madrid Telf.:913676440
miguel.garcia@sicingenieros.com

⁽²⁾ Peopleware, Cañada Real Merinas 7, 2º. Centro de negocios Eisenhower 28042 Madrid.
Telf.:917462255 joseangel.chico@peopleware.com

RESUMEN

Descripción del proyecto

Osmius (<http://www.osmius.net>) es una herramienta de monitorización de redes que se distribuye bajo licencia GPL v2. Se encarga de monitorizar cualquier elemento conectado a nuestra red, servidores web, bases de datos, sensores... En su versión 9.07, liberada en julio del 2009, se ha iniciado un proyecto para la integración de capacidades SIG. Se busca conseguir funcionalidades SIG que aporten un valor añadido a la herramienta, y que permitan la entrada de elementos geográficos en la toma de decisiones. Mediante la integración se ha conseguido lograr un panel de control con información geográfica asociada que nos permite la toma de estas decisiones basándonos en criterios geográficos.

Descripción técnica

Osmius almacena toda la información generada durante su funcionamiento en una base de datos MySQL. Esta se organiza en instancias que se encargan de monitorizar y servicios que engloban estas instancias. Para la integración, se han aprovechado las propiedades espaciales de la base de datos MySQL como fuente de datos que sirve al SIG. Se presentan estos datos utilizando un servidor SIG web, GeoServer, que se nutre de ellos y los publica bajo estándares OGC. Como capa de interfaz con el usuario, se ha desarrollado un visor web, con tecnología OpenLayers, a través del cual se puede visualizar, generar y filtrar dicha información. Para la capa base sobre la que visualizamos toda la información se ha utilizado OpenStreetMap, aunando así en un solo desarrollo tres importantes proyectos del panorama SIG libre actual.

Actualmente se están desarrollando mas funcionalidades SIG para la visualización y gestión de la información intentando extraer todo el potencial de los proyectos libres que se están utilizando, utilizando las últimas versiones de estos y adaptando sus capacidades a la herramienta.

Palabras clave: Osmius, integración, Geoserver, OpenLayers, OpenStreetMap, MySQL Spatial

DESCRIPCIÓN DEL PROYECTO

Motivaciones

Osmius[1] es una herramienta que se encarga de la monitorización de cualquier sistema que se encuentre conectado a la red. Podemos detectar el estado de nuestros sistemas y actuar de una manera rápida evitando así causar en la medida de lo posible perjuicios a nuestros usuarios.

En la versión previa a la integración de sistemas de información geográficas, Osmius se encargaba exclusivamente de la monitorización de la red. Pensando en darle mayores capacidades, y fruto de la unión de esfuerzos de empresas con líneas de negocio diferentes, se procedió a realizar un proyecto de integración de Osmius con herramientas SIG. Esto permitirá no solamente controlar el funcionamiento de los sistemas, sino además, en combinación con las herramientas geográficas, el poder establecer políticas de actuación y de subsanación de errores. Tener un control geográfico de las instalaciones permite, además, planificar las actuaciones y planificar políticas de contingencia, ayudando al mantenimiento de los mismos.

Asimismo, en la misma línea de desarrollo de capacidades, se desarrolló paralelamente un proceso de integración similar, pero esta vez con herramientas de Business Intelligence [2], que permitirían a Osmius administrar y crear conocimiento mediante el análisis de los datos que se generan durante el proceso de monitorización, proporcionando a los usuarios finales de los servicios nuevas herramientas para la toma de decisión.

Las motivaciones surgen de explorar nuevas líneas de negocio en el mundo de las TIC, y de detectar posibles mejoras de las funcionalidades de un producto que ofrezcan mayor valor añadido, pudiendo realizar una gestión más completa de la información, incluyendo mas variables en el análisis de esa información, teniendo en cuenta la variable geográfica.

Osmius, la herramienta de monitorización

Osmius se encarga de la monitorización de equipos conectados a la red. Actualmente se encuentra en su versión 10.01.1. Es un producto Open Source, distribuido bajo licencia GNU GPL v2, desarrollado por la empresa PeopleWare.

Está compuesto por un servidor central (CS), los agentes maestros (MA) y los agentes (A). Utiliza una arquitectura basada en agentes. Son estos agentes los que monitorizan nuestro entorno y nos envían eventos que podemos consultar y gestionar de forma centralizada.

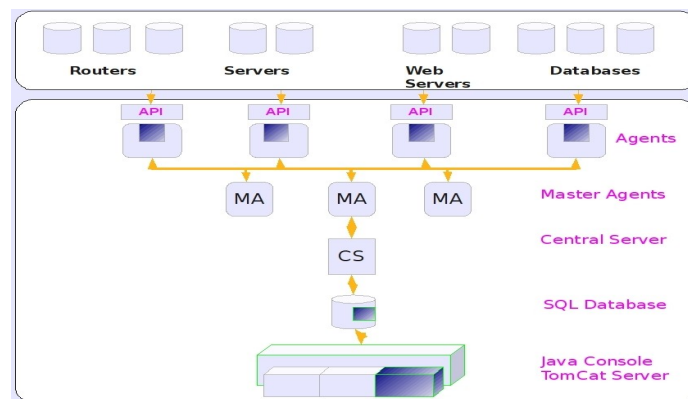






Fig 1. Arquitectura basada en agentes

Si empezamos desde la capa más técnica estaremos interesados en monitorizar una serie de variables de cada uno de los elementos que forman la infraestructura de Sistemas de Información. Esta infraestructura será desde decenas a miles de elementos.

Estos elementos se llaman en Osmius **INSTANCIAS**, y no todas son iguales, tendremos instancias de tipo "Servidor Linux", de tipo "Router ACME", bases de datos "Postgress", y así un largo etcétera. Desde el punto de vista de monitorización los tipos de instancia se diferencian entre sí por el tipo de variables por los que puedo preguntarles, a cada variable por la que puedo preguntar a un determinado tipo de instancia la llamamos TIPO DE EVENTO. Cuando preguntamos a una instancia por un tipo de evento, obtengo un **EVENTO**. Así que cuando tenga montado el Sistema de Monitorización estaré recibiendo eventos, cada uno de un Tipo de Evento, con unos valores de cada una de las Instancias que tengo en mis sistemas. En función del valor del evento para una instancia determinada estaré más o menos tranquilo con el resultado. Si el evento que nos devuelve la temperatura de la Sala de Servidores nos informa de que es de 21 grados Celsius estaré tranquilo, pero si dice que es de 51 grados Celsius tenemos un problema.

Tabla 1. Tabla de Estados

Color	Estado
	OK
	Warning
	Critical
	Error

Así para cada Tipo de Evento y cada instancia definimos unos umbrales para saber el **ESTADO** del evento, de la misma forma cada instancia tiene un **ESTADO DE INSTANCIA** que depende de los estados de los eventos activos de esa instancia. Osmius calcula el estado de una instancia como el más crítico de los estados de los eventos activos de esa instancia.

Cada Tipo de Evento que queramos monitorizar de una Instancia tiene asociado un **PERIODO de ejecución**.

Desde la capa de negocio primero definiremos los **SERVICIOS**. Un Servicio es una agrupación de instancias que ofrecen de manera conjunta una serie de funcionalidades a los usuarios. Existe también un **ESTADO DE SERVICIO**, que depende de los estados de las instancias que lo componen. De igual modo tenemos la **DISPONIBILIDAD de un SERVICIO** que se calcula como Disponible comprobando que todos sus instancias estén disponibles; en caso contrario el servicio se encontrará No-Disponible. Todo Servicio tiene asociado un **ANS o Acuerdo de Nivel de Servicio**.

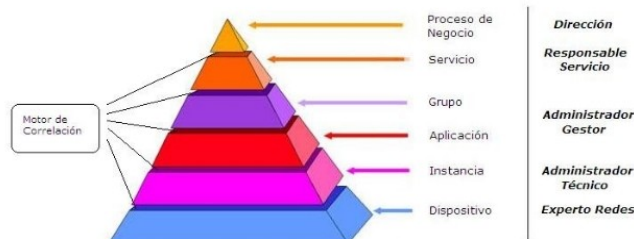


Fig 2. Esquema del modelo de negocio

En resumen, Osmius monitoriza las instancias de nuestra instalación preguntando periódicamente por unos tipos de eventos. Una vez estos son recibidos en el servidor central, son correlacionados y se actualizan los estados de las Instancias y de los Servicios.

Los componentes que sujetan el proceso son los siguientes:

- **Agentes**, que son los encargados de ejecutar periódicamente las acciones asociadas a cada Tipo de Evento. Un agente es especialista en recoger eventos de un único tipo de Instancia. Tendremos así agentes para instancias de tipo Oracle o Windows o Valores en Bolsa que normalmente utilizarán las API proporcionadas por los diferentes fabricantes de cada tipo de instancia.
- **Agentes Maestros**, cada Agente Maestro se encarga de controlar y gestionar un conjunto de Agentes Osmius dentro de su mismo servidor. Son los encargados de recibir los Eventos de los Agentes para a su vez reenviarlos por red y protocolo seguro al Servidor Central para su tratamiento, arrancar y parar los Agentes... Básicamente los Agentes Maestros se encargan y concentran todas las tareas administrativas para mantener nuestra infraestructura de monitorización.
- **Servidor Central**, es el que recibe los eventos de todos los Agentes de la infraestructura de monitorización que hayamos montado en nuestras instalaciones y además el encargado de enviar las tareas necesarias a los Agentes Maestro para su proceso.
- **Base de datos**, en la que se guardan diferentes tipos de Información. Se trata de una instancia de MySQL.
- **Consola de Gestión**, cualquier usuario con un navegador y los permisos adecuados puede conectarse a la consola y comenzar a monitorizar y gestionar toda la infraestructura. La consola de Osmius es una aplicación Web hecha en Java y basada en varios Frameworks que se ejecuta en un servidor TomCat con conexión a la base de datos.

DESCRIPCIÓN TÉCNICA

Objetivos

En una primera fase del proyecto se definieron unos objetivos iniciales que fuesen sentando una base para el proyecto:

1. Asociar información geográfica a la información generada en Osmius
2. Servir esa información. Necesidad de un servidor de cartografía
3. Desarrollo de una interfaz de usuario, integrada en la consola de Osmius para acceder a la información
4. Desarrollo de funcionalidades de gestión de la información desde la consola.

Se trata de un proceso común dentro de cualquier desarrollo de aplicaciones que trabajen con información geográfica, con el añadido de realizarlo integrándolo en una plataforma ya existente por lo que debe hacerse ateniéndose a unas condiciones de partida ya marcadas inicialmente.

Arquitectura inicial

Osmius está desarrollado bajo una estructura cliente-servidor, orientada a agentes. Se ha desarrollado de una manera modular, por la parte del cliente se desarrollan los agentes que se instalarán en el sistema a monitorizar, y por la parte del servidor, se desarrolla el Central Server. Los agentes maestros se desarrollan sobre el cliente y el servidor.

Para el desarrollo de Osmius se han utilizado proyectos Open Source, como Tomcat y MySQL.

Toda la información que recoge Osmius durante el proceso de monitorización es almacenada en la base de datos, a la vez que todo lo respectivo a infraestructura, usuarios, seguridad... La base de datos sobre la que trabaja Osmius es MySQL. Durante la instalación de Osmius se crea la instancia de la base de datos, y se recrea el modelo de datos para después poblarla con la información mínima necesaria para arrancar la aplicación.

El core de la aplicación está desarrollado con C++, igual que algunos de los agentes. Esta parte se encarga de almacenar en la base de datos toda la información que es generada por los agentes, se encarga de recibir la información de los agentes maestros y actualizar toda la información.

Por la parte de la interfaz con el usuario, se ha desarrollado una consola web que es accesible desde cualquier navegador. Esta se ha desarrollado bajo una arquitectura J2EE, utilizando algunos de los frameworks mas conocidos de dicha arquitectura, como Hibernate para la capa modelo, o Spring.

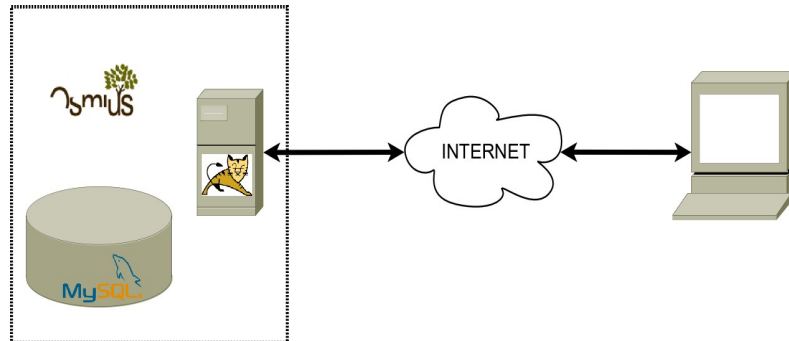


Fig 3. Arquitectura inicial consola web

Arquitectura integración

Partiendo de la arquitectura inicial, se buscó la mejor forma de dar solución a la situación que se nos planteaba:

- Arquitectura cliente-servidor
- Falta de información geográfica asociada.
- Base para el desarrollo posterior de funcionalidades mas complejas.
- Sencillez de integración, instalación...
- Creación de una interfaz de usuario integrada en la arquitectura de la consola.

Teniendo en cuenta la situación de inicio se realizó un estudio de las opciones que existían. Se revisó las bases de datos con capacidades geográficas, servidores web de cartografía y las APIs para el desarrollo de la interfaz de usuario.

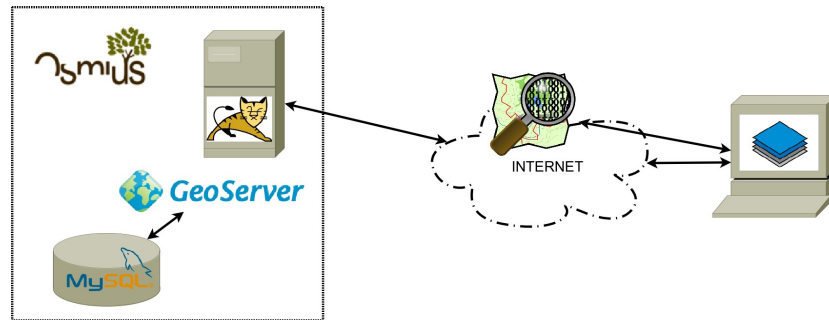


Fig 4. Arquitectura integración

Base de datos con capacidades espaciales

MySQL es la base de datos sobre la que trabaja Osmius. Este sistema incorpora de manera nativa extensiones para el tratamiento de información espacial. Ya desde la versión 5.0.16 estas funcionalidades están para todos los motores de bases de datos. Estas extensiones siguen la especificación de la Open Geospatial Consortium (OGC)[3], que propone varias maneras de extender el SQL del sistema gestor de base de datos para soportar datos espaciales.

Al tratarse de la base de datos que implementa Osmius, la existencia de capacidades espaciales minimiza el impacto a la hora de generar la información espacial asociada. Para generar esta, se revisa el modelo de datos y se generan tablas geográficas que se relacionan a las tablas con la información alfanumérica.

Estas tablas geográficas se crean utilizando el motor MyISAM [4], ya que es el que permite la creación de índices espaciales. La información que se almacenará en estas tablas serán los índices de los objetos con los que van a ser relacionados, y la información espacial.

En esta primera fase, se ha asociado información geográfica a las INSTANCIAS, y a los SERVICIOS que son las primeras que nos interesa geoposicionar. Poblar las tablas se realizará en una segunda parte del desarrollo mediante una herramienta de inserción de información geográfica desde la consola web.

Servidor web SIG

Una vez que tenemos creada la infraestructura para el almacenamiento de datos espaciales se procede a la incorporación al sistema de la parte que se encargará de servir la información geográfica que se genere.

Se han comparado varios de los proyectos mas representativos dentro de la oferta de servidores desarrollados de manera libre. Teniendo en cuenta la arquitectura de partida se definieron varias premisas que debía cumplir:

- Simplificación del proceso de integración
- Instalación sencilla, integración en proceso instalación
- Posibilidad de ampliación de funcionalidades

Tras revisar varios de los software, GeoServer fue el que mas se adecuó a las necesidades del proyecto. Geoserver es la implementación de referencia de los estándares del OGC. Permite una instalación sobre un servidor web que soporte servlets.

Dentro de la arquitectura de partida, Tomcat es el servidor sobre el que se ejecuta la consola web que da acceso a la información de Osmius. La posibilidad de desplegar Geoserver dentro del servidor de partida hizo que fuese la mejor opción para desarrollar en el proyecto.

Funcionalmente Geoserver soportaba los formatos con los que en principio íbamos a trabajar, WMS y WFS, existiendo en la actualidad una línea de desarrollo de WPS.

Para la conexión a la fuente de datos era necesario instalar el conector de Geoserver con MySQL, ya que este no viene por defecto en la instalación sino como una extensión. Este conector se distribuye desde el mismo proyecto y su instalación es sencilla, permitiéndonos acceder a las tablas con la información geográfica. Para el tratamiento fue necesario la creación de vistas en el gestor de bases de datos ya que Geoserver no permite realizar uniones de tabla.

Una vez que tenemos la información para poder publicarla se hizo necesaria la creación de un filtro visual para mostrar las instancias y servicios en función del estado. Para ello se crearon archivos de estilo de capa (SLD, Style Layer Descriptor) con reglas para el filtrado por los atributos de los objetos.

Desarrollo de la interfaz de usuario

Como parte final de la integración se necesitaba crear una interfaz para que el usuario pueda interactuar con la información. Esta debe estar integrada en la consola web de Osmius. La consola tiene una arquitectura J2EE. utiliza Hibernate para realizar la gestión de la base de datos, y está desarrollada sobre el framework Spring. Se buscó realizar una integración sencilla, apoyada en los estándares para evitar generar una gran cantidad de desarrollo.

Al igual que con los servidores web SIG, se revisaron los proyectos de web mapping open source, y se decidió el uso de OpenLayers [5]. Esta es una librería JavaScript que permite la creación de mapas interactivos en los navegadores web. A su vez se apoya en librerías JavaScript como Rico y Prototype. Permite la conexión con infinidad de servicios, existen en la librería muchos controles, y permite el desarrollo de controles personalizados. El manejo de la información recibida desde el servidor se facilita, al estar utilizando estándares tanto en el servidor como en el visor que los consume, ya que existen objetos que permiten un manejo sencillo de la información, como parseadores de GML, que evitan gran cantidad de desarrollo.

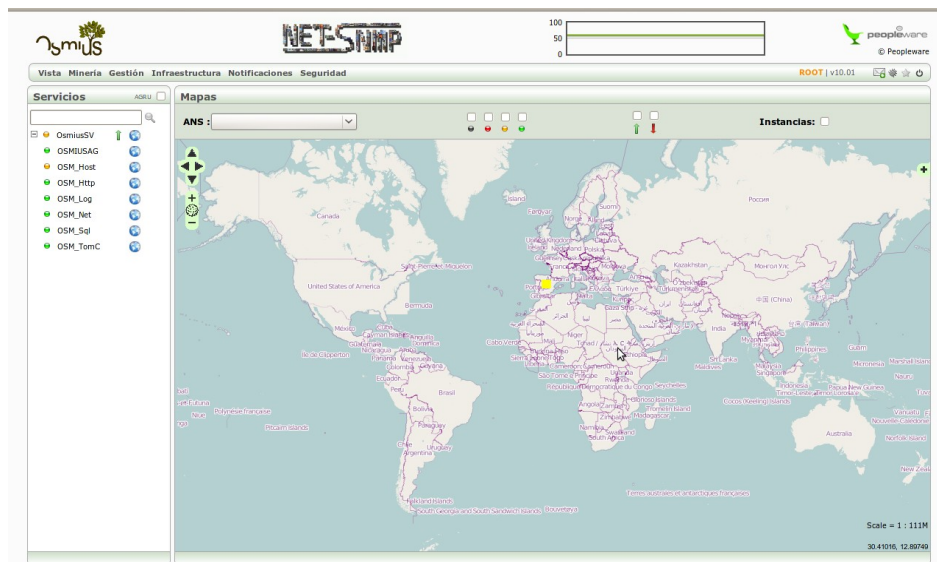


Fig 5. Consola acceso a información cartográfica

Se generaron las páginas necesarias que soportarían el visor dentro de la consola de Osmius. Se integró la gestión de la seguridad y se procedió al desarrollo del visor. Se integró un mapa con los controles básicos de navegación, y se incorporaron las capas con la información que se estaba sirviendo desde Geoserver, instancias y servicios, ya con los estilos que les fueron asignadas. Se desarrollaron, añadiendo filtros a las consultas que se le hacen al servidor, funcionalidades de filtrado de la información, por estado, ANS y disposición, para así permitir una navegación mas intuitiva por la información.

Para la inserción de la información geográfica se ha desarrollado una parte de la aplicación que permite recoger las coordenadas desde el visor y asignárselas a el objeto que tengamos seleccionado.

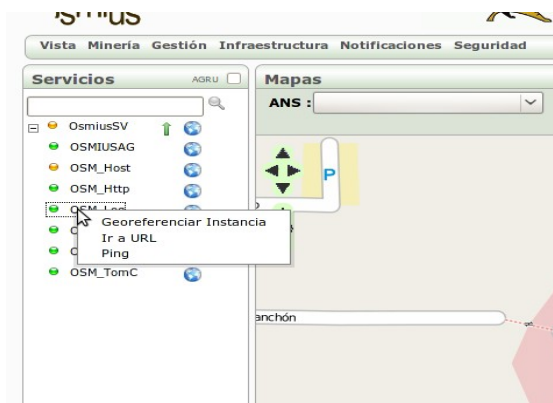


Fig 6. Detalle georeferenciación

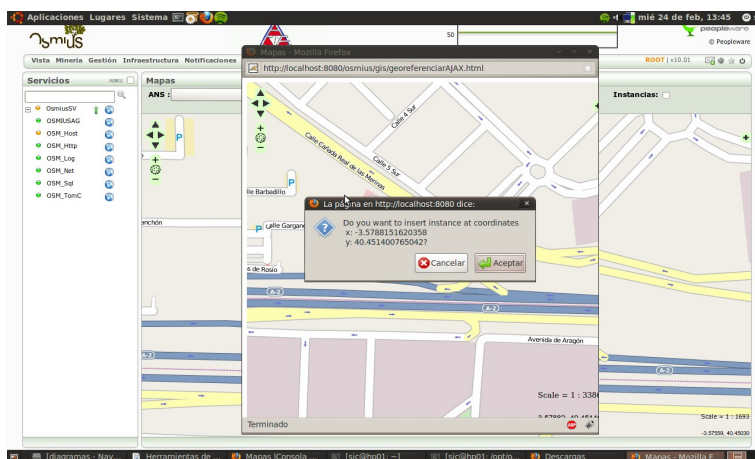


Fig 7. Proceso inserción coordenadas

A través de peticiones al servidor, adaptadas a los estándares, seremos capaces de traer la información necesaria de la instancia o servicio y mostrar en un menú dicha información, asimismo se puede dibujar una diagrama con figuras vectoriales con las instancias que corresponden al servicio que estamos consultado, mejorando la visualización de las relaciones entre estos objetos.

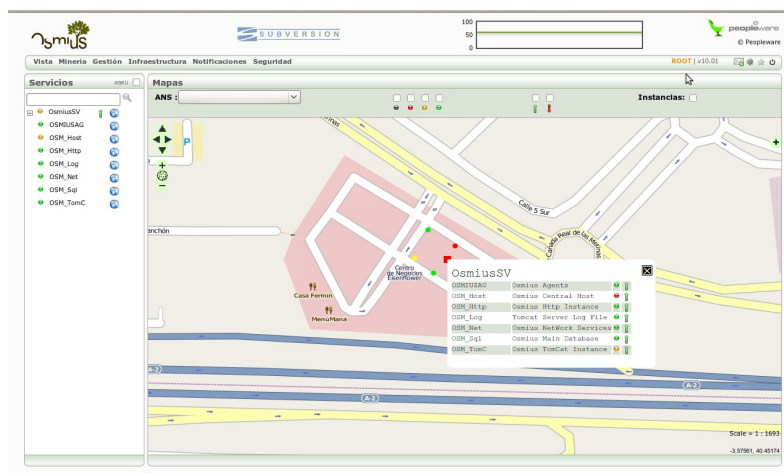


Fig 8. Presentación información instancias del servicio

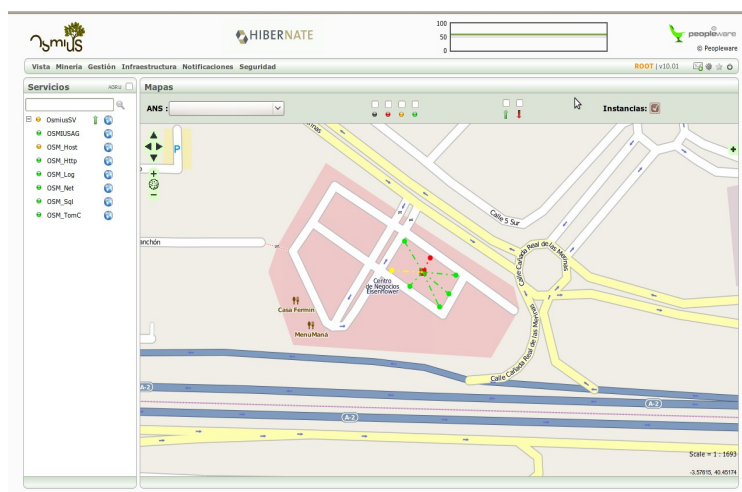


Fig 9. Vector presentación instancias

Era necesario disponer de una cartografía base sobre la que visualizar la información que servíamos. Se barajaron varias posibilidades, pero debido al carácter del proyecto, y la falta de limitaciones a la hora de utilizar la cartografía se decidió el uso de OpenStreetMap como cartografía base. OpenLayers, y OpenStreetMap, disponen ambas de opciones en su API para una integración de esta cartografía en los visores desarrollados con dicha librería.

Un caso práctico, MKD

Se ha desarrollado para uno de los clientes parte de la funcionalidad de la herramienta. Se trata de una adaptación de la herramienta de monitorización al despliegue y gestión de puestos de Marketing Dinámico. Se trata de poder obtener la información de los diferentes puestos y añadirles la capacidad de cruzar dicha información con la variable geográfica.

Se mostrará durante la presentación parte de la funcionalidad desarrollada.

Futuros desarrollos

Teniendo una base sobre la que seguir trabajando, aparece la necesidad de generar mayor funcionalidad:

- La principal línea de trabajo actual es el desarrollo de la capacidad de generar análisis geográficos con la información de la que disponemos. Se está planteando la posibilidad de colaborar en el desarrollo del módulo WPS del servidor para proporcionar esas funcionalidades.
- Mayor funcionalidad en el visor.
- Desarrollo de un interfaz para que el usuario pueda disponer de su propia cartografía, con la posibilidad de insertar y navegar por planos de planta de edificios, y no se descarta en el futuro la navegación por entornos 3D.
- Un servicio de búsqueda y geocodificación para la inserción de la información.
- Integración de una gestión de rutas

VÍDEO DEMO

Durante la presentación se mostrará un vídeo con la aplicación en funcionamiento

Referencias

- [1] Osmius <http://www.osmius.com/es/product/>
- [2] Wikipedia http://es.wikipedia.org/wiki/Inteligencia_empresarial
- [3] Open Geospatial Consortium <http://www.opengeospatial.org/>
- [4] Definición MyISAM <http://es.wikipedia.org/wiki/MyISAM>
- [5] Proyecto Openlayers <http://openlayers.org/>

CONCLUSIONES

A medida que se ha ido avanzando en el desarrollo se ha ido demostrando la importancia de añadir a los análisis la variable geográfica. Desde el mero echo de la visualización, que facilita el adquirir una idea mas global de la situación, por ejemplo a la hora de actuar ante determinadas incidencias al tener esa componente que te permite tomar una decisión mas adecuada, como el enviar a la unidad de reparación mas cercana de un simple vistazo, o con la capacidad de relacionar visualmente los diferentes objetos.

Las posibilidades de adaptación de la herramienta se nos muestran amplias. La posibilidad de monitorizar cualquier tipo de elemento con la simple creación de un agente adaptado permite añadir a la herramienta infinitas posibilidades añadido a las capacidades de análisis geográficos, calculo de rutas, análisis espacial,...

Técnicamente es importante definir desde un principio las líneas de integración ateniéndose a la base de partida, ya que una decisión errónea lleva a multiplicar los tiempos de desarrollo. La asunción de estándares y software de fuentes abiertas disminuye enormemente este tiempo, ya que aporta mucha de la funcionalidad y soluciones a situaciones que ya han sido planteadas.

AGRADECIMIENTOS

A la gente de PeopleWare, a todos, por su colaboración y su paciencia, a toda la gente que desarrolla en los proyectos de los que hemos hecho uso, los que hemos usado para la parte SIG, Geoserver, OpenLayers, OpenStreetMap y de los que estos a su vez derivan, sin los que sin duda no hubiese sido posible esto y de los proyectos que hace uso Osmius, que estoy seguro la gente de Peopleware estará agradecida.