

LOCALIZA: una herramienta SIG para resolver problemas de localización óptima.

J. Bosque Sendra⁽¹⁾, F. Palm^(2,3) y M. Gómez⁽¹⁾

⁽¹⁾ Departamento de Geografía, Universidad de Alcalá, Calle Colegios, 2, Alcalá de Henares, Madrid 28801. joaquin.bosque@uah.es, montserrat.gomez@uah.es.

⁽²⁾ Instituto de Estadística Aplicada y Computación, Universidad de Los Andes. Av. Las Américas, Núcleo Liria, Edif. "G", Piso 1, Mérida, Venezuela.

⁽³⁾ Centro de Simulación y Modelos, Universidad de Los Andes. Núcleo La Hechicera - Edif. B - 3er Nivel - Ala Norte, Mérida, Venezuela. fpalm@ula.ve.

RESUMEN

El programa LOCALIZA es un herramienta que permite aumentar las capacidades de las aplicaciones SIG para analizar y resolver problemas de localización óptima de equipamientos. Este sistema, que está siendo migrado desde Delphi a Python, permite estudiar de que manera la oferta de equipamientos existente cubre la demanda. Por otra parte incluye la resolución de un número elevado de modelos de localización-asignación clásicos y, en algún caso, de nueva redacción.

En esta comunicación se discutirán las ventajas y dificultades encontradas para generar, bajo el paradigma del software libre y de código abierto, herramientas que aumentan las funciones de los SIG privativos y abiertos. Igualmente se plantearán las posibilidades de reconstruir y mejorar esta herramienta utilizando todo el potencial de los procedimientos de desarrollo abierto y colaborativo disponibles en la actualidad.

Palabras clave: LOCALIZA, localización-asignación, Software Libre, Python.

ABSTRACT

LOCALIZA software is a tool that increases GIS applications possibilities to analyze and solve optimal facility location problems. This system, that it is being migrated from Delphi to Python, allows to evaluate how existing facility supply covers the demand. On the other hand, it includes the resolution of an elevated number of classic location-allocation models and, in some cases, includes new models.

This communication discuss advantages and difficulties to generate, using the free and open source software paradigm, tools that increase the functionality of privative and free GIS. The possibilities of reconstructing

and improving this tool using the full potential of open and collaborative development procedures available at the present time will also be discussed.

Key words: LOCALIZA, location-allocation, Software Libre, Python.

INTRODUCCIÓN

LOCALIZA es una herramienta para la realización de análisis de localización de instalaciones y equipamientos. Permite llevar a cabo dos tipos de actividades:

- La evaluación de una situación actual, es decir comprobar el grado en que la distribución espacial y el tamaño de la oferta de un servicio cubre las necesidades de la demanda existente en un momento dado
- Generar nuevas soluciones ante una distribución espacial de la demanda de un servicio y las posiciones de puntos que están dispuestos a ser sedes de nuevas instalaciones donde se ofrece el servicio estudiado.

Este sistema ha sido diseñado para, combinado con las potencialidades de un SIG, convertirse en un verdadero Sistema de ayuda a la decisión para los problemas de localización óptima de instalaciones y equipamientos.

En este texto se discute en primer lugar el problema concreto que se pretende resolver, y las funciones y posibilidades incluidas en LOCALIZA (en relación a otras posibles). En segundo lugar, se explica como se ha elaborado la primera versión, enumerando las herramientas empleadas para ello. Finalmente, en una tercera parte, se plantea la posibilidad de reconstruir el sistema haciéndolo menos dependiente de un SIG concreto y usando otras herramientas de programación mas modernas ligadas al Software Libre.

LA LOCALIZACIÓN ÓPTIMA DE INSTALACIONES

En la vida cotidiana una cuestión muy usual es la necesidad de desplazarse sobre el espacio geográfico para poder hacer uso de diversos servicios. Los trabajadores deben moverse de su domicilio a su lugar de empleo para poder llevar a cabo su actividad productiva, los estudiantes deben trasladarse hasta las escuelas, institutos o centros universitarios para recibir sus enseñanzas, los enfermos se desplazan hasta los centros de salud, etc.

Todos estos desplazamientos suponen un gasto económico o de tiempo o esfuerzo, por ello la organización de la sociedad y su plasmación sobre el territorio geográfico implica considerar la posibilidad de estudiar como son estos desplazamientos, tan numerosos y que afectan a tantos millones de personas de manera cotidiana, para intentar reducirlos y simplificarlos en una alguna medida [1].

Esta tarea ha sido abordada desde muchos puntos de vista, la Geografía ha sido una de las disciplinas tradicionalmente interesada en este tema, y desde hace ya

bastantes años existe una tradición de análisis de la cuestión empleando técnicas cuantitativas para valorar el problema y modelos de localización óptima para ofrecer buenas soluciones al problema [2].

Los Sistemas de información geográfica se han popularizado en los últimos decenios y se han planteado cuestiones relacionadas, en alguna medida, a esta problemática. Sin embargo, realmente este tema no está contemplado, de manera directa, entre las funciones y actividades usualmente incluidas en los SIG comerciales [3].

En especial, los modelos de localización óptima, herramientas muy útiles para tratar esta temática, no suelen estar incorporados a los SIG. Su desarrollo, importante desde el punto de vista matemático y aplicado, se ha producido al margen de los SIG [2].

La observación de esta realidad esta llevando a diversos autores a tratar de resolver esta anomalía, en este sentido y con esta finalidad se ha desarrollado LOCALIZA [3].

Los equipamientos y servicios analizados por LOCALIZA son los equipamientos o bienes suministrados en instalaciones situadas en lugares concretos del espacio, la población que los usa debe desplazarse hasta allí para utilizar ese servicio (a veces es el bien el que debe moverse para suministrar a la población: el servicio de bomberos por ejemplo). Un hecho esencial es que el coste de recorrer esta separación, entre instalaciones y población demandante, resulta de gran importancia en el uso del servicio [4].

Se pueden diferenciar varios tipos de equipamientos:

- Equipamientos deseables: producen un efecto beneficioso en su alrededor (generan externalidades positivas) y, por ello, la población la cual está interesada en tenerlos cerca de su residencia. Por ejemplo, escuelas, hospitales, comercios, etc.
- Equipamientos no deseables: su existencia produce un efecto perjudicial en sus alrededores (externalidades negativas), por lo tanto la población no desea tenerlos cerca de su domicilio. Por ejemplo, vertederos de basuras, centrales nucleares, cárceles, etc.

Los procedimientos que integran LOCALIZA parten, generalmente, de un dato básico: la distribución espacial de la demanda y de su tamaño.

En función de este dato básico, y, en ocasiones, de otras informaciones, LOCALIZA resuelve dos tipos de cuestiones:

- [1] Evaluar la situación de la oferta disponible en un caso real o las ofertas obtenidas mediante los modelos de localización asignación.
- [2] Buscar nuevas localizaciones para los centros de oferta.

En la resolución de cualquiera de estos problemas se suelen emplear dos principios básicos [5] y [6]:

[1] Principio de la eficiencia espacial: se buscan/evalúan positivamente configuraciones de la oferta que, dada una posición concreta de la demanda, determinen un valor mínimo (o máximo según el tipo de equipamientos) de alguna medida (la suma total de movimientos, la distancia máxima entre oferta y demanda, etc) de los desplazamientos necesarios para utilizar los servicios por toda la población.

[2] Principio de la justicia espacial: para gran número de los equipamientos estudiado es importante igualmente, que no existan grandes diferencias geográficas en la posibilidad de usar o ser afectado por los servicios planteados. Las distancias a recorrer entre demanda y oferta no deberían ser excesivamente diferentes, ya que esto implica grandes diferencias en el acceso a la oferta.

LA ESTRUCTURA Y FUNCIONES DE LOCALIZA

A partir de los conceptos e ideas anteriores la primera versión de LOCALIZA contempla la siguiente lista de funciones y capacidades, en cada caso el sistema lee los datos geográficos y temáticos necesarios, los procesa y genera los resultados y reportes básicos :

Funciones utilizables en la exploración de la situación existente con el fin de desarrollar hipótesis para la resolución de su problemática.

Estas funciones sirven para realizar análisis de tipo descriptivo que conducen

I. EVALUACIÓN DE LA ADECUACIÓN DE LA OFERTA A LA DEMANDA

- Medidas globales de la adecuación de la oferta a la demanda.
- Evaluación de la situación en cada zona de servicio.
- Población servida y no servida por la oferta dada su configuración espacial y un valor del alcance espacial.
- Cartografía de los puntos de demanda y de oferta. el mapa tipo "araña" y el mapa "anular".

II. MEDIDAS DE LA ACCESIBILIDAD DE LA DEMANDA (INSTALACIONES DESEABLES)

- Accesibilidad obtenida considerando la distancia (coste de recorrido) en cada punto de demanda al centro de oferta más próximo.
- Accesibilidad obtenida considerando la distancia a los centros de oferta y tamaño de la oferta en ellos.
- Accesibilidad (en cada punto de demanda) obtenida considerando la relación entre tamaño de la oferta y tamaño de la demanda dentro del alcance espacial del servicio.

**III. DETERMINACIÓN DE ÁREAS DE INFLUENCIA DE LOS CENTROS DE OFERTA.
PROBLEMAS DE ASIGNACIÓN.**

- En función de la distancia más corta entre oferta y demanda.
- En función del coste de recorrido a los diferentes centros de oferta y del tamaño de la oferta en cada uno de ellos, modelo de Huff.

Procedimientos que permiten llevar a cabo la generación de soluciones al problema de la localización óptima de instalaciones.

Consiste en la resolución de modelos de programación matemática para la selección de patrones espaciales óptimos para la demanda de un servicio en función de relaciones de distancia y magnitud con un patrón de demanda dado, estos son denominados modelos de localización-asignación, entre los que se distinguen los siguientes tipos básicos [4]

INSTALACIONES DESEABLES

- Modelo p-mediano (MINISUM)
- Modelo p-mediano con restricción de máxima distancia (MEDIRES).
- Modelo de cobertura máxima (COBEMAX).
- Modelo de máxima cobertura con restricción de alejamiento (COBERES).

```

1 Función Objetivo: 15186135.95
2
3 --Dist.Demanda--
4 Suma-Producto: 15186135.95
5 Promedio: 106.4655245
6 Desviación Típica: 406294.2971
7 Distancia Mínima: 0
8 Instalación: 2
9 Sitio: 78
10 Distancia Máxima: 1560.660034
11 Instalación: 20
12 Sitio: 38
13 Mediana: 11.39949369
14 Cuartil Inferior: 6.828427315
15 Cuartil Superior: 252
16
17
18 Id-INS N-SIT
19 Dist.Dem
20 1 12
21 2 21
22 3 11
23 4 11
24 5 29
25 6 15
26 7 5
27 8 1
28 20 1
    
```

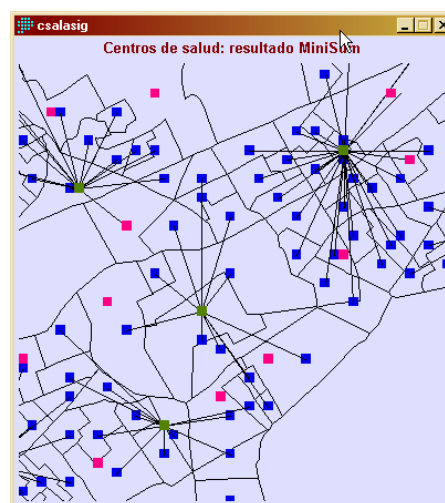


Figura 1: Reporte del módulo MINISUM

INSTALACIONES NO DESEABLES

- Modelo p-mediano invertido(MAXISUM)
- Modelo de cobertura mínima (COBEMIN).
- Modelo maximización de la mínima distancia (MAXIMIN).

INSTALACIONES COMERCIALES

- Modelo maximización de la asistencia. Objetivo de maximización de la demanda con elasticidad lineal respecto a la demanda (DECLIN):
- Modelo de maximización de la demanda con descenso de la demanda según una función exponencial negativa de los costes de transporte (DESNOLIN).

INSTALACIONES SEMI-NOCIVAS y/o SEMI-DESEABLES

Modelo de minimización de la distancia a los productores y maximización de la distancia a la población (MINMAXSUM).

EL DISEÑO Y PROGRAMACIÓN DE LOCALIZA

Tal y como se dijo mas arriba, Localiza está diseñado como un Sistema de Ayuda a la Decisión Espacial (SADE), de un modo similar al propuesto en [7]. El diseño inicial de LOCALIZA partió de la idea de ampliar las capacidades del sistema IDRISI [3]. Bajo este esquema, se delegaba en IDRISI las tareas relacionadas con la creación y el manejo de las bases de datos espaciales, y LOCALIZA se encargaba del manejo de modelos de análisis y la generación de reportes. Uno de los principios de LOCALIZA siempre ha sido presentar un interfaz gráfica de fácil uso para el usuario final, y una arquitectura abierta y adaptable para la investigación.



Figura 2: Estructura básica de LOCALIZA

IDRISI tiene en una excelente relación precio-valor dado por un extenso conjunto de herramientas de análisis, uno de los más completos entre los software Raster SIG existentes en el mercado. Sin embargo, el uso de IDRISI supuso limitar la

interoperabilidad de LOCALIZA en cuanto a formatos de datos e interacción con otras aplicaciones y plataformas [8].

Por otra parte, el diseño de las Interfaces de Programación de Aplicaciones (API) de Idrisi 2.0 e Idrisi32 tampoco favorecían la interoperabilidad, la arquitectura no está desagregada y no permite manipular de forma eficiente y directa el contenido de los archivos, esto último es fundamental para el desarrollo de LOCALIZA.

Con la finalidad de mejorar el rendimiento e interoperabilidad del sistema, se utilizó en LOCALIZA la librería InovaGIS, una librería gratuita de clases geográficas que tenía la finalidad de servir como tecnología para el desarrollo de aplicaciones de información geográfica. Estaba diseñada con la finalidad de manipular los datos geográficos de distintos formatos de una manera desagregada y estaba formada por clases donde se encapsulan los datos geográficos junto con la funcionalidad básica asociada a los mismos.

Sin embargo, InovaGIS también tenía problemas de estabilidad y funcionalidad que no fueron atendidos, y al igual que IDRISI 2.0, era muy sensible a las configuraciones regionales del Windows.



Figura 3: Ventana Principal de la Interfaz Gráfica de la primera versión de LOCALIZA

La programación de las primeras versiones de LOCALIZA se realizó empleando Delphi. Este no es un lenguaje de programación en el mismo sentido que lo son C, C++ o Java. En realidad, es una herramienta de programación que engloba: un lenguaje de programación: Object Pascal, un entorno integrado de desarrollo, y una librería de clases visuales (VCL), que permiten el desarrollo rápido de aplicaciones [9]. Todos estos elementos facilitaron mucho el trabajo de construcción de LOCALIZA.

En el desarrollo de LOCALIZA se hizo uso extensivo de componentes VCL adicionales que se podían obtener de forma gratuita y legal por Internet, y permitían mejorar las interfaces gráficas y el manejo del entorno integrado de programación. Además, se utilizaron otras herramientas gratuitas accesibles por Internet, entre las

que se encuentran un sistema instalador gratuito basado en Delphi (Inno Setup) y un generador automático de diagramas de clases (ESSModel).

En términos de Ingeniería de Software, se detectaron tardíamente problemas en el diseño de la arquitectura del sistema LOCALIZA, ya que hay componentes donde se mezclan el manejo de la interfaz de usuario y las operaciones de análisis. Este es un problema recurrente en el desarrollo de sistemas que ha llevado a propuestas como la arquitectura de “tres capas”: Presentación, Lógica y Datos, y el patrón de diseño MVC: modelo-vista-controlador.

LOCALIZALIBRE, MIGRACIÓN DE LOCALIZA A SOFTWARE LIBRE

El Desarrollo de LOCALIZA se inició en el año 2000, para entonces Delphi era sin lugar a dudas la mejor herramienta de desarrollo rápido de aplicaciones para Windows. En ese momento, Linux estaba lejos de ser considerado una opción para usuarios finales, GRASS estaba incluso más lejos aún de tener una versión nativa para Windows, y bajo ese marco, IDRISI resultaba una de las mejores opciones disponibles para análisis Raster.

Pero muchas cosas han cambiado desde entonces. Las principales herramientas de desarrollo de LOCALIZA han caído en la obsolescencia y la desatención por causa de intereses estrictamente comerciales. Mientras tanto el mundo seguía con atención el enorme auge del movimiento de Software Libre, que precisamente nace como una respuesta, entre otras, a los problemas generados en todos los ámbitos de la ciencia por las políticas depredadoras de la industria del software.

El adjetivo “libre” se refiere en este caso particular a:

- La libertad de leer, modificar y distribuir la información, en consecuencia se requiere el uso de formatos de archivo bajo estándares libres.
- La libertad de acceder al conocimiento subyacente en las aplicaciones, y por lo tanto disponer del código fuente de las herramientas informáticas usadas.
- La libertad de ejecutar el sistema en distintas plataformas computacionales y sistemas operativos.
- La libertad de conocer en detalle el proceso de desarrollo del software, e incluso participar activamente en el mismo. Esto es posible gracias a Internet y las herramientas teleinformáticas para el desarrollo colaborativo.

Una característica en el mundo del Software Libre es que rara vez existe una única alternativa para dar respuesta a un problema tecnológico determinado. Por esta razón, es necesario desarrollar un criterio propio para tomar decisiones, obliga a “pensar fuera de la caja”, fuera del cerco de las limitaciones del modelo tecnológico y de negocios impuesto por las industrias de software privativo. A continuación, se describirán las alternativas libres seleccionadas para sustituir cada componente con los fines de la migración.

Uso de Python en lugar de Delphi

Como principal herramienta de desarrollo utilizada ha sido el lenguaje de programación Python. Como plataforma de desarrollo debe dar cuenta de los tres aspectos mencionados arriba:

El lenguaje de programación: Python en sí mismo es un lenguaje de programación que forma parte de una nueva generación de lenguajes dinámicos, con enormes ventajas sobre Object Pascal: puede ejecutarse en varias plataformas (arquitecturas de computador y sistemas operativos) prácticamente sin cambios, admite diversos paradigmas de programación como el funcional y el orientado aspectos de forma bastante natural, y permite el uso de técnicas avanzadas de programación orientada a objetos como la metaprogramación.

Al igual que Delphi, Python está orientado a mejorar la productividad del programador pero usando una estrategia diferente: en lugar de intentar ser una versión ligera de lenguajes como C++ o Java, propone un enfoque totalmente novedoso de la programación que permite centrar la atención en la solución del problema y no tanto en las particularidades del lenguaje [11].

Los otros dos elementos, el entorno integrado de desarrollo y la librería de Componentes Visuales, son precisamente elementos de la estrategia de Delphi para mejorar la productividad (similar en esencia a las propuestas de mejora de la productividad dadas por Java y .Net). Sería necesaria una discusión extensa sobre estas cuestiones pero está fuera del alcance de este documento. La experiencia indica que el uso de Python en el modo interpretado con herramientas como IPython, y la propia naturaleza "introspectiva" de Python son factores mucho más significativos en la mejora de la productividad del programador.

Además, los editores de texto por excelencia en el software libre como Emacs y Vi, cuentan con soporte para Python. Esto sin mencionar los entornos integrados de desarrollo diseñados expresamente para Python tales como SPE, Eric o PIDA.

Otra parte fundamental para el desarrollo rápido de aplicaciones son las facilidades para el desarrollo de interfaces gráficas de usuario (GUI por *Graphic User Interface*). De nuevo, se cuenta con varias alternativas de donde escoger tales como Qt Designer o GLADE, basadas en las librerías gráficas Qt y GTK respectivamente. Estas herramientas permiten diseñar interfaces gráficas usando técnicas de "programación visual", de manera independiente a un lenguaje de programación en particular. Así, una misma interfaz puede ser enlazada a una aplicación desarrollada en C++, Java, Perl o Python, según requiera el programador.

Finalmente, la librería de clases visuales de Delphi es reemplazada por la librería estándar de Python que bajo la premisa de "pilas incluidas" (*batteries included*) señala que Python por defecto provee una gran cantidad de facilidades al programador, para interactuar con el sistema operativo y con Internet, y agregando funcionalidad como el uso de expresiones regulares. Aunado a la gran cantidad de componentes y herramientas disponibles en el repositorio oficial de Python denominado "Cheese

Shop” y otros repositorios como Sourceforge. Las extensiones para el desarrollo en Python superan con creces las que jamás tuvo Delphi.

Uso de GDAL en lugar de InovaGIS

GDAL (Geospatial Data Abstract Layer) es una librería para el manejo de datos geoespaciales que permite manejar una enorme cantidad de formatos geoespaciales, más de 50 formatos de archivos Raster, usando un único modelo abstracto de datos, lo que facilita en gran manera el desarrollo de aplicaciones. Curiosamente, es mucho más eficiente la manipulación de archivos RST de Idrisi con esta librería que con la propia API de Idrisi.

Dentro de la estructura de GDAL existe una librería similar denominada OGR que da soporte a más de 20 formatos de archivos Vectoriales. El nombre de OGR surgió inicialmente de “OpenGIS Simple Features Reference Implementation” pero no está aprobada como una implementación de referencia por el OpenGIS Consortium, por esta razón se cambió el nombre a “OGR Simple Features Library”, donde OGR tan sólo tiene un significado histórico.

GDAL es mucho más de lo que se pudo concebir en las previsiones más optimistas de InovaGIS. Es una librería de código abierto bajo licencia X/MIT, es utilizada por una gran cantidad de aplicaciones libres y algunas privativas como Google Earth o el propio ArcGIS 9.2+. Es mantenido activamente por una comunidad internacional de desarrolladores, y tiene garantizado el financiamiento a mediano y largo plazo. Es multiplataforma, cuenta con interfaces de programación para los lenguajes de programación C++, C, y Python soportados oficialmente, y para Perl, VisualBasic 6, Ruby, Java y C# gracias a iniciativas particulares.

Uso de GRASS en lugar de IDRISI

GRASS GIS es la herramienta por excelencia para la manipulación y el análisis de información geoespacial en el mundo del software libre. Este proyecto se inició en el año 1982 ante la total ausencia de herramientas SIG para el análisis ambiental avanzado. GRASS representa una experiencia pionera en el desarrollo colaborativo de software, ya que debido a restricciones presupuestarias para la adquisición de software y hardware decidieron confiar desde un principio en sus propias capacidades y en los fundamentos del código abierto y el desarrollo colaborativo [12].

En virtud de esta característica de ser una herramienta de código abierto, GRASS se ha mantenido en constante desarrollo, cada módulo de GRASS es revisado y mejorado bajo un proceso continuo por sus propios usuarios, quienes en caso de estar involucrados en el desarrollo de GRASS modifican directamente el sistema, o en caso contrario notifican estos fallos obteniendo respuestas en lapsos de tiempo de horas o días. La comunidad de usuarios de GRASS se encuentra en continuo crecimiento y existe una lista de discusión activa desde diciembre de 1991.

En relación al desarrollo de LOCALIZA, una de las tareas críticas consiste en calcular de matrices de distancia, con esta finalidad se determinan superficies de costo basadas en superficies de fricción. En IDRISI se utiliza para esta tarea los

I Jornadas de SIG Libre

módulos COSTPUSH o VARCOST, que son en esencia los mismos desde las versiones de IDRISI para DOS. Dado que no se dispone del código fuente de estos módulos es imposible depurarlos o personalizarlos. En GRASS los módulos equivalentes son r.cost y r.walk, el código fuente de estos módulos está disponible, su uso es mucho más flexible y existe una documentación disponible en Internet de su historial (donde se da crédito a todos los que han contribuido en su desarrollo). Esto ha dado pie incluso al desarrollo de nuevos módulos como r.terracost que hace uso de técnicas de optimización de entrada/salida y cómputo paralelo para conjuntos de datos masivos [13].

En cualquier caso, IDRISI continua siendo una excelente herramienta para el análisis Raster, pero su uso está limitado al escritorio Windows, su arquitectura es monolítica y las mejoras en las últimas versiones están enfocadas a mejoras en la usabilidad, es decir, en términos de geocomputación está dirigida a usuarios finales y no a investigadores. Por su parte, GRASS puede usarse de manera desagregada y se integra bien con otras herramientas de Software Libre, el desarrollo de la versión nativa para Windows lo convertirá en una verdadera herramienta multiplataforma.

Componentes de LocalizaLibre

El objetivo de LOCALIZA es convertirse en una herramienta sin dependencias privativas, tanto en el uso como en el desarrollo. Además, se espera que sea una herramienta que capte la atención de nuevos desarrolladores, y para tal fin se deben seguir las pautas de desarrollo de las comunidades de software libre, y garantizar las condiciones que permitan atraer los investigadores con el perfil adecuado para añadir herramientas para la toma de decisiones.

Se ha diseñado entonces una herramienta modular y interoperable basada en Python, que pueda extenderse con otras técnicas de análisis espacial como soporte a la toma de decisiones. A continuación, se mostrará una breve revisión de los componentes seleccionados para la implantación del sistema.

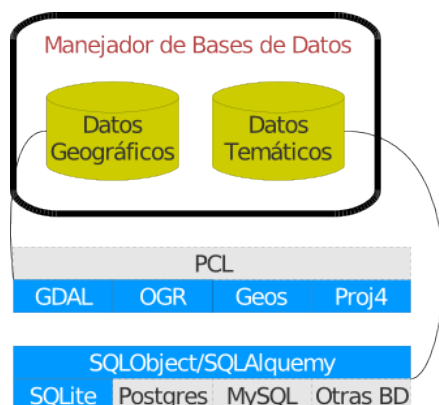


Figura 4: Componentes del manejador de bases de datos

Python cuenta con una gran cantidad de posibilidades para el desarrollo de sistemas conectados a bases de datos relacionales. Resultan muy interesantes los proyectos de transformación objeto-relacional (object relational mapper): SQLAlchemy y SQLObject. Estas herramientas diseñadas para los frameworks de desarrollo web permitan manejar bases de datos relacionales como objetos, utilizando un modelo abstracto transparente a los distintos sistemas manejadores de bases de datos libres y privativos. La próxima versión de LOCALIZA contempla el uso de SQLite, pudiéndose ampliar a otros manejadores de bases de datos en un futuro cercano.

Para los datos geográficos se cuenta con las librerías GDAL y OGR que permiten acceso directo a los mapas, la librería GEOS permite manipulaciones geométricas y de topología, y Proj.4 se encarga cualquier transformación requerida en los sistemas de proyección. Se está considerando el uso a futuro de la librería PCL (Python Cartographic Library), que consiste en una integración de estas librerías junto con Mapscript (el API de Mapserver) y Matplotlib para facilitar el manejo y visualización de datos geográficos desde Python.

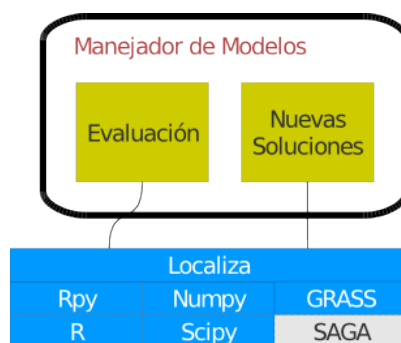


Figura 5: Componentes del manejador de modelos.

En la implementación de los modelos es donde Python destaca especialmente. Las rutinas de análisis propias de LOCALIZA están siendo implantadas usando principalmente la librería NumPy y llamadas puntuales a algunos módulos de GRASS (y en el futuro aumentar las capacidades de análisis usando el SIG SAGA). NumPy permite el manejo eficiente de matrices con programación de muy alto nivel, se considera parte del proyecto SciPy que provee una enorme cantidad de herramientas adicionales para el computo científico.

La librería Rpy permite la conexión directa de Python con el Sistema Estadístico R.y acceder a sus enormes capacidades de procesamiento y visualización estadística.

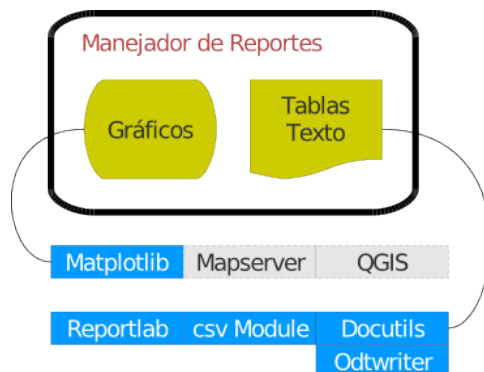


Figura 6: Componentes del manejador de reportes.

Para los reportes gráficos se piensa emplear la librería Matplotlib que permite generar desde Python gráficos de excelente calidad, incluidos mapas. Se espera a futuro integrar LOCALIZA a las librerías de QGIS, lo cual permitiría generar composiciones de capas configurables, y también proveer una interfaz web para LOCALIZA mediante Mapserver.

Los reportes de texto pueden generarse de forma automática con distintos módulos: ReportLab para PDF, el módulo CVS de la librería estándar de Python para archivos separados por comas, y en base al sistema de documentación Docutils pueden transformarse archivos de texto en HTML, LaTeX, DocBook, entre otros.

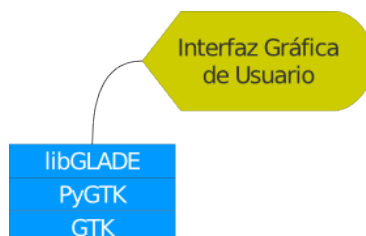


Figura 7: Componentes utilizados en la interfaz gráfica de usuario.

Finalmente, la interfaz gráfica se está desarrollando en base a la librería PyGTK, diseñando la interfaz con la herramienta de programación visual GLADE. Estas librerías han sido portadas exitosamente a Windows y MacOS X.

CONCLUSIONES

Se puede afirmar que LOCALIZA es un software necesario, muchas tareas de gestión pública necesitan de la asistencia de herramientas sofisticadas para la toma de decisiones basadas en información geoespacial. El desarrollo de este tipo de sistemas requieren a su vez la conformación de equipos interdisciplinarios, y tradicionalmente de la utilización de software privativo sumamente costoso que por añadidura no cuenta con características apropiadas para el trabajo científico. Bajo estas premisas, el desarrollo de un sistema como LOCALIZA sin dependencias de

software privativo provee enormes ventajas en términos económicos, sociales científicos y tecnológicos.

Python es un lenguaje de programación dinámico que está atrayendo poderosamente la atención de la comunidad internacional de SIG Libre. Actualmente, Python es el lenguaje de scripts "oficial" de la librería GDAL/OGR, puede ser utilizado para desarrollar servidores web de mapas usando Mapserver bajo modo Mapscript, y recientemente se están desarrollando los vínculos con las interfaces de programación de los sistemas GRASS, SAGA y QGIS. Si consideramos las enormes capacidades que cuenta Python actualmente para el computo científico, y otras experiencias similares de desarrollo de sistemas de análisis espacial tales como EPIGRASS [14], STARS [15], o URBANSIM [16] se puede afirmar que el nuevo sistema LOCALIZA representa un ladrillo más de una plataforma de computación geoespacial creciente para el desarrollo de sistemas de ayuda a la decisión espacial basados en software libre.

REFERENCIAS

- [1] Bosque Sendra, J. y Moreno Jiménez, A. (2004), "La localización óptima como problema: cuestiones teóricas y metodológicas". En Bosque Sendra, J. y Moreno Jiménez, A. (Eds.) "Sistemas de información geográfica y localización de equipamientos". Madrid, Edit. RA-MA.
- [2] DeVerteuil, G. (2001), "Reconsidering the legacy of urban public facility location theory in human geography". *Progress in Human Geography*, 24, 1, pp. 47-69.
- [3] Bosque Sendra, J., Gómez Delgado, M., Moreno Jiménez, A. y Dal Pozzo, F. (2000), "Hacia un sistema de ayuda a la decisión espacial para la localización de equipamientos". *Estudios geográficos*, 2000, tomo LXI, nº 241, pp. 567-598
- [4] Bosque Sendra, J. y Moreno Jiménez, A. (2004), "El uso de los sig para localizar equipamientos e instalaciones". En Bosque Sendra, J. y Moreno Jiménez, A. (Eds.) "Sistemas de información geográfica y localización de equipamientos". Madrid, Edit. RA-MA.
- [5] McCallister, D. M.. "Equity and Efficiency in Public Facility Location," *Geographical Analysis*, Vol. 8, (1976), pp. 47-63.
- [6] Morrill, R. L. Y Symons, J.. "Efficiency and Equity aspects of Optimum Location," *Geographical Analysis*, Vol. 9, (1977), pp.215-225.
- [7] Densham, P.J. (1991). "Spatial decision support systems", en *Geographical information systems: principles and applications*, Maguire, D. J., Goodchild, M. S. y Rhind, D.W., eds., (London: Longman), pp. 403-412.
- [8] Palm Rojas, F. (2004), "Aspectos del desarrollo de localiza". En Bosque Sendra, J. y Moreno Jiménez, A. (Eds.) "Sistemas de información geográfica y localización de equipamientos". Madrid, Edit. RA-MA.
- [9] Texeira, S. y Pacheco, X. (2000). "Guía de Desarrollo Delphi 5". Madrid, Pearson Educación.
- [10] Gonçalves, P.P. y Costa, M. (1999). "Local and Remote Geoprocessing Applications", *Computers, Environment and Urban Systems*, 23, pp. 287-303.
- [11] Raymond, E. (2000). "Why Python?", *Linux Journal*. May, 73
<http://www2.linuxjournal.com/1j-issues/issue73/3882.html>.

- [12] Westervelt, J. (2004). "GRASS Roots". Proceedings of the FOSS/GRASS Users Conference, Bangkok, Thailand, 12-14 September.

- [13] Hazel, T., Toma, L., Vahrenhold, J. y Wickremesinghe, R. (2006) "Terracost: A Versatile and Scalable Approach to Computing Least-Cost-Path Surfaces for Massive Grid-Based Terrains". Proceedings of the 21st Annual ACM Symposium on Applied Computing.
- [14] Codeço, C. y Celho, F. (2006), "EPIGRASS GIS: um GIS livre para apoio ao desenvolvimento de politicas publicas em saude"
<http://epigrass.sourceforge.net/>
- [15] Rey, S. y Janikas, M. (2006), "STARS: Space–Time Analysis of Regional Systems". Geographical Analysis 38: pp. 67-84.
- [16] Borning, A., Waddell, P., y Förste, R., (2006), "UrbanSim: Using Simulation to Inform Public Deliberation and Decision-Making", Preprint of paper to appear, Digital Government: Advanced Research and Case Studies, Hsinchun Chen et al. (eds.), Springer-Verlag, in press.