

---

---

## XVI Congreso Nacional de Tecnologías de la Información Geográfica 25, 26 y 27 de Junio de 2014. Alicante.

---

---

# Estrategias de diseño de software para la actualización de parcelas catastrales: el caso del proyecto Ramón Llull

Miguel Fernández Moreno<sup>a</sup>, José Tomás Navarro Carrión<sup>a</sup>, Jorge Piera Llodrá<sup>b</sup>

<sup>a</sup>Laboratorio de Geomática, Instituto Interuniversitario de Geografía, Universidad de Alicante

<sup>b</sup>Ingeniero informático responsable del área de desarrollo de Trac-Trac

---

### Resumen

El Catastro Inmobiliario es un registro administrativo dependiente del Ministerio de Hacienda y Administraciones Públicas regulado por el Texto Refundido de la Ley del Catastro Inmobiliario 1/2004, que establece que la inscripción de un inmueble en el Catastro es obligatoria y gratuita.

La representación cartográfica de los bienes inmuebles es uno de los aspectos fundamentales a la hora de inscribirlos en este registro oficial. Esta descripción gráfica cambia cuando se producen transacciones de fincas que afectan a la delimitación física de la propiedad, lo que conlleva tanto un control jurídico como técnico. El proyecto de investigación "Ramón Llull" se centra en el desarrollo de software de geoprocésamiento para comprobar y validar estos procesos de alteración catastral y subsanación de discrepancias en los que intervienen los Notarios. Uno de los logros fundamentales es una herramienta de gestión de expedientes de alteración que permite además coordinar el trabajo de la notaría con el del técnico competente e incorporar la información gráfica de la propiedad a la escritura.

Para el desarrollo de esta aplicación ha sido preciso planificar el proceso de diseño que ha seguido el equipo investigador a la hora de escribir código fuente estable y de calidad. En el presente trabajo se analiza cómo han influido estas decisiones en el resultado. La valoración y la elección de software libre, los patrones de diseño de software y la interoperabilidad de datos y servicios son los factores en los que se centra el estudio. Todos ellos han condicionado el producto final: un sistema de geoprocésamiento que permite que una transacción inmobiliaria entre particulares y la correspondiente actualización de la cartografía catastral coincidan en tiempo y forma.

Palabras clave: Cartografía catastral; proyecto Ramón Llull; geoprocésamiento; software libre; patrones de diseño; interoperabilidad.

---

## 1. Introducción

La información geográfica es un elemento indispensable para el conocimiento, la identificación y la planificación del territorio. Este trabajo se engloba en un proyecto de investigación que se ocupa concretamente de la información relativa a la delimitación de la propiedad - estratégica por sus implicaciones tributarias y legales - y de la actualización de la cartografía oficial sobre la cual se sustenta. En España la base cartográfica de referencia para la delimitación de la propiedad es la cartografía del Catastro Inmobiliario. Tiene valor jurídico ante cualquier operación de subsanación o de modificación catastral en todo el Estado en virtud de la Ley de Economía Sostenible (*Ministerio de Hacienda y Administraciones Públicas, 2011*) y el Texto refundido de la ley del Catastro (*Ministerio de Hacienda y Administraciones Públicas, 2013*). Además, es pública y accesible desde la implantación de la directiva europea INSPIRE (*IDEE, 2014*).

En el contexto de los procesos telemáticos de actualización de la cartografía catastral, el Instituto Interuniversitario de la Universidad de Alicante, en colaboración con el Departamento de Derecho Civil de la misma universidad, trabaja en el proyecto de investigación aplicada “Ramón Llull”. Uno de sus objetivos generales es desarrollar software de geoprocesamiento que asista a las notarías en la gestión de alteraciones catastrales producto de transacciones inmobiliarias entre particulares. Entre sus fines particulares cabe destacar la puesta en marcha de un sistema capaz de (i) procesar los casos contemplados por el formulario 903-N de Catastro, (ii) preparar la nueva información gráfica del inmueble para su presentación en Catastro e (iii) incorporarla simultáneamente a la escritura (*Llorens Cobos, Mira Martínez, Navarro Carrión & Ramón Morte, 2005*). De este modo, al hacer coincidir el procedimiento administrativo y el acto jurídico, se propicia que la resolución del trámite de delimitación de la propiedad y del cambio de titularidad se efectúe en el menor tiempo posible y se garantice la seguridad jurídica de la transacción.

Los desarrollos tecnológicos del Proyecto Ramón Llull entroncan con iniciativas similares de otros organismos en el ámbito de la gestión de la información catastral, como es el caso del Sistema de Información Geográfica Catastral (SIGCA) y de Geobase. El SIGCA, y actualmente también el SIGCA2, es el sistema de Catastro para la gestión de la información geográfica, y proporciona un interfaz a los técnicos de las áreas de rústica y urbana para la actualización de la cartografía catastral (*Conejo Fernández & Virgós Soriano, 2001*). Por su parte, Geobase es una aplicación web, promovida desde el colectivo de los Registradores de la Propiedad, que explota una base de datos geográfica corporativa que permite identificar las fincas registrales con alto índice de fiabilidad (*Montoro Rueda, 2011*).

El desarrollo principal del Proyecto Ramón Llull es un sistema de gestión de expedientes de agregación, segregación, división parcelaria y deslinde mediante el cual una notaría coordina el proceso de alteración con el técnico encargado de la delimitación física de la propiedad. El sistema no almacena cartografía, sino que se ocupa del geoprocesamiento de cada expediente para la actualización del parcelario catastral conforme se sustancia una transacción. Cada expediente constituye el ámbito de ejecución de una secuencia ordenada de geoprocesos sobre la cartografía de referencia y la información gráfica aportada por el técnico para que el volcado final de datos en Catastro se realice con exactitud (*Navarro Carrión & Ramón Morte, 2013*).

Los requerimientos funcionales de este sistema de geoprocesamiento han dado lugar a una aplicación compleja internamente. Entre estos aspectos funcionales, los más influyentes han sido los puramente computacionales y los relativos a usabilidad. Los primeros abarcan principalmente los procesos de validación topológica, que plantean no sólo cuestiones de precisión sino también de modelización, ya que finalmente es necesario manejar casos reales muy variables. En cuanto a la usabilidad, el gestor de expedientes es una

herramienta colaborativa que proporciona un interfaz común en el que un técnico competente y un notario coordinan información gráfica y jurídica de forma simultánea y transparente. A ello se suma la posibilidad de acceso al expediente por parte de otros perfiles profesionales, la conveniencia de consumir la cartografía catastral en línea y la capacidad de generar formatos de salida asumibles por parte de Catastro.

La proliferación de Infraestructuras de Datos Espaciales y la creciente participación ciudadana en su mantenimiento y actualización dan pie al desarrollo de aplicaciones enfocadas al geoprocesamiento. Si, además, manejan información relativa a la delimitación territorial de derechos, este tipo de aplicaciones han de enfrentarse a casuísticas variables de modificación y ser accesibles por perfiles muy diversos de usuario, como es el caso del gestor de expedientes del Proyecto Ramón Llull. Bajo estas premisas resulta imprescindible evaluar y escoger distintas estrategias de diseño de software. En el presente trabajo se detalla las estrategias y prácticas de programación fundamentales por las que se ha optado en el Proyecto Ramón Llull y se analiza cómo ha influido cada elección en el resultado final. El cuerpo central de esta comunicación se ocupa de dichas estrategias, que se clasifican en tres categorías, a saber, (i) licencias, (ii) arquitectura y patrones de diseño e (iii) interoperabilidad de datos y servicios.

## 2. Licencias

El gestor de expedientes del Proyecto Ramón Llull está desarrollado sobre software libre. Al comienzo del proyecto el equipo de investigación ya contaba con amplia experiencia en el diseño y explotación de bases de datos geográficas con PostgreSQL/PostGIS, y en el desarrollo de clientes *desktop* en C# para geoprocesamiento y visualización cartográfica. También se disponía de una serie de librerías y utilidades desarrolladas para otros proyectos sobre SharpMap y NetTopologySuite, el *port* de Java Topology Suite (JTS) para entornos de ejecución CLI. JTS implementa el estándar *Simple Feature Access* del OGC y proporciona acceso a funciones de geometría computacional y cálculo topológico idóneas para modelizar procesos de alteración entre pares de parcelas catastrales.

El software libre surge como reacción a la restricción de acceso a código fuente que en origen siempre fue abierto (*Stallman, 1999*) y regula y protege la libertad de lectura, ejecución, modificación y redistribución del código (*Free Software Foundation, 2014*), no su gratuidad. En este contexto, los desarrollos colaborativos de código abierto en el campo de las TIG han dado lugar a un ecosistema de librerías, aplicaciones e infraestructuras de soporte de datos geográficos que ha dado en llamarse FOSS4G y que ha pasado en pocos años de un estadio de adopción temprana a otro de madurez y adopción creciente (*Steiniger & Hunter, 2013*). Es un hecho que el panorama FOSS4G actual ofrece multitud de opciones y bases de código estables y fiables.

Además de usar PostgreSQL como sistema de base de datos, el Proyecto Ramón Llull emplea librerías FLOSS y FOSS4G entre las que cabe destacar Spring.NET, Npgsql, NetTopologySuite, SharpMap y PdfSharp. Los tipos de licencia libre abarcan (i) MIT, que puede considerarse la más liberal (Open Source Initiative, 2014), (ii) Apache 2.0, que no requiere que las versiones modificadas tengan que ser distribuidas como software libre (Apache, 2014) y (iii) LGPL, que revoca el comportamiento “viral” de la licencia GPL (Free Software Foundation, 2014).

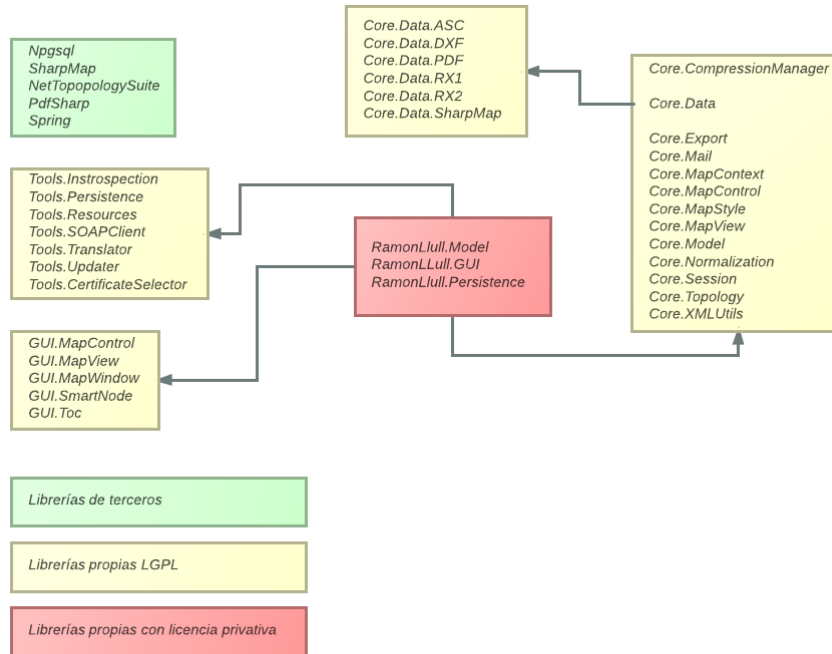


Fig. 1. Distribución de librerías por licencia.

En general, se trata de licencias permisivas en cuanto a la convivencia de software libre y propietario. Esto facilita la organización y distribución de los ejecutables de forma que se respeta tanto la propiedad intelectual como la potestad del financiador del proyecto para licenciar su código. Este es el caso, por ejemplo, del interfaz de usuario y de las librerías directamente acopladas al modelo de datos (véase Figura 1).

### 3. Arquitectura y patrones de diseño

#### 3.1. Inyección de dependencias

En ingeniería del software, la teoría de la inversión de control establece que, en un diseño modular correcto, un componente reutilizable o módulo de alto nivel no puede depender de un módulo de bajo nivel, es decir, de aquel que resuelve en la práctica un determinado aspecto operativo (Martin, 1996). Una de las principales implementaciones de la inversión de control es la denominada inyección de dependencias o DI, por su acrónimo en inglés. La DI es un patrón de instanciación usada en la programación orientada a objetos que permite cambiar el comportamiento de un objeto en tiempo de ejecución, y facilita una mayor reutilización del código (Prasanna, 2009). De este modo, es posible mantener determinados componentes o librerías inalterados, y a la vez adaptar el comportamiento de la aplicación a distintos escenarios y necesidades. El empleo de la DI es fundamental para automatizar los tests y, por tanto, para evitar regresiones en proyectos con miles de líneas de código y múltiples desarrolladores.

Para la implementación de este patrón se ha usado Spring.NET, un *port* a C# del *framework* Spring para Java. Con Spring se consigue que el código esté menos acoplado ya que las dependencias se establecen a nivel de especificación de API, y el enlace con la implementación concreta se produce en tiempo de ejecución. Esto hace que el código sea más sencillo de mantener y que se puedan generar tests de unidad normalmente inviables cuando el acoplamiento es mayor.

Al emplear la DI, los constructores de clase no se invocan explícitamente, sino que los objetos se instancian dinámicamente a partir de un interfaz y mediante un componente que sigue el patrón *Factory* o *Builder*. El uso de estos patrones de diseño de creación promueve a su vez la escalabilidad.

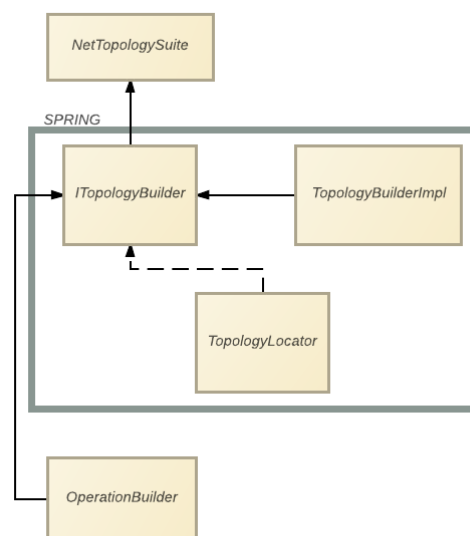


Fig. 2. Ejemplo de implementación con Spring de *Core.Topology*

En la Figura 2 se muestra un ejemplo de implementación del patrón *Builder* usando el *framework* Spring para el API encargado del cálculo topológico del gestor de expedientes. Desde la clase *OperationBuilder* se necesita instanciar un objeto que implemente la interfaz *ITopologyBuilder*, sin importar la implementación que exista por debajo. Mediante un fichero de configuración xml, Spring se encarga de la inyección de dependencias - es decir, de ensamblar la implementación con su correspondiente interfaz - usando *ApplicationContext*, un registro de configuración que gestiona la instanciación de objetos en tiempo de ejecución (*Pivotal Software, Inc, 2014*). Todo el código que hace uso de Spring se encapsula en un contenedor (*TopologyLocator* en el ejemplo de la Figura 2), de forma que fuera del ámbito de este objeto el uso de este *framework* es transparente.

### 3.2. Model-View-Presenter

Un sistema de geoprocésamiento con un alto grado de usabilidad necesariamente cuenta con un interfaz gráfico de usuario (GUI) complejo, independientemente de su apariencia. De hecho, a pesar de su apariencia sencilla, el GUI del gestor de expedientes del Proyecto Ramón Llull consta de unos cuarenta elementos de diálogo. Abordar este tipo de desarrollo implica escoger una arquitectura de capas de software que permita separar la capa lógica del GUI. En nuestro caso se seleccionó el patrón *Model-View-Presenter* (MVP), una derivación de la arquitectura *Model-View-Controller*. Este patrón añade un intermediario (*Presenter*) que controla la comunicación y el paso de mensajes entre la lógica operativa y el modelo de datos (*Model*) y cada elemento de diálogo del GUI (*View*). En general las peticiones se realizan mediante funciones delegadas que permiten al usuario interactuar con la capa lógica. La delegación ayuda a manejar comportamientos en tiempo de ejecución, y aunque dificulta la lectura secuencial del código, promueve la modularidad y facilita enormemente la parametrización (*Gamma, Helm, Johnson & Vlissides, 1995*).

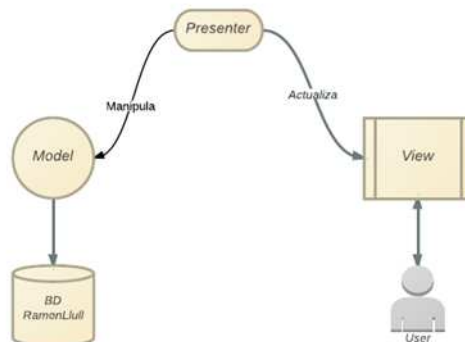


Fig. 3. Patrón Model-View-Presenter.

La implementación típica del patrón MVP en el gestor de expedientes (véase Figura 3) separa cada transacción de base de datos (*Model*) del renderizado de su correspondiente cuadro de diálogo (*View*). El componente *Presenter* comunica ambas capas mediante delegados. Al reducir el acoplamiento, los componentes del GUI y de la capa lógica se pueden desarrollar independientemente, se da prioridad al principio de separación de funciones y se consigue una base de código más flexible a los cambios. Por otra

parte, y al igual que ocurre con la inyección de dependencias, este tipo de patrón favorece (i) la generación de tests de unidad en la capa lógica e (ii) incrementa la posibilidad de reutilización de código.

### 3.3. Patrón Strategy

En programación orientada a objetos, el patrón *Strategy* pertenece a la categoría de patrones de comportamiento. Se usa cuando se necesitan distintas variantes de un algoritmo cuya implementación no es visible para el cliente. A partir de un conjunto algoritmos, este patrón permite encapsularlos por separado y hacerlos intercambiables. Si se desciende a su implementación, se parte de una clase que define muchos comportamientos, es decir, el contexto en el que se diseña una estrategia. Dicha clase ha de declarar una interfaz común a todos los algoritmos, y evitar así la profusión de estructuras condicionales anidadas (Gamma, Helm, Johnson & Vlissides, 1995).

El patrón *Strategy* es una solución natural a la hora de implementar un motor de geoprocésamiento, puesto que prima el principio de composición frente al de herencia. En el gestor de expedientes del Proyecto Ramón Llull, cada tipo de alteración catastral lleva aparejado un algoritmo de validación topológica en forma de secuencia ordenada de geoprocésos. Por tanto, este patrón permite componer estos procesos de forma modular, descentralizar y simplificar el código - dado que los geoprocésos no se concentran en una única clase - y hacer escalable el modelo, dado que para añadir un tipo de alteración catastral basta con implementar un interfaz y encapsular su algoritmo de cálculo topológico.

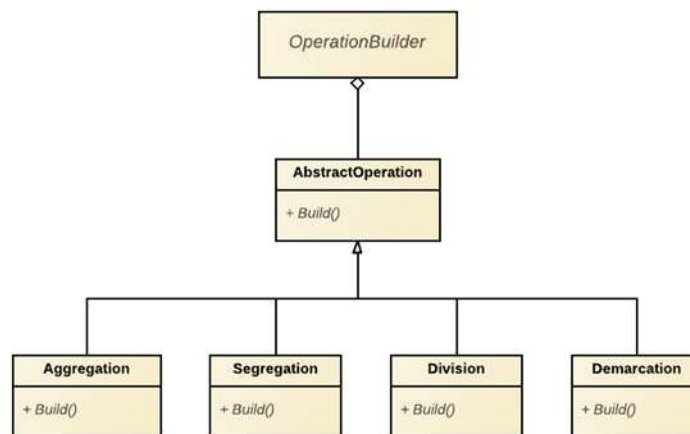


Fig. 4. Implementación del patrón Strategy

Como se ve en la Figura 4, el contexto en que se define la estrategia de geoprocésamiento es la clase *OperationBuilder*, que declara una interfaz común a todas las operaciones de alteración catastral (*AbstractOperation*). Cada tipo concreto de operación de alteración catastral ha de implementar el método *Build()*, donde se encapsula el algoritmo correspondiente. Así pues, agregar un nuevo algoritmo de geoprocésamiento consiste en definir una clase derivada de *AbstractOperation* e implementar en el método

Build() una determinada secuencia de llamadas a funciones de volcado en memoria de geometrías, cálculo de superficies, generación de grafos topológicos y evaluación de predicados relacionales.

## 4. Interoperabilidad

### 4.1. Datos

La interoperabilidad de datos es un aspecto clave en entornos colaborativos. Dado que es un ámbito de coordinación de datos catastrales, el gestor de expedientes del Proyecto Ramón Llull emplea FXCC - especificación oficial de intercambio para la actualización de cartografía catastral (*Dirección General de Catastro, 2014*) - como estándar de entrada de datos para una referencia catastral concreta. La especificación FXCC describe una finca a partir de dos ficheros ASCII, uno de información alfanumérica estructurada en campos y otro de geometría vectorial en formato DXF. El formato DXF es un estándar *de facto* para entornos CAD y por ello se usa a su vez como base para la integración automática de los datos del técnico competente en el expediente. Con ello se consigue un alto grado de interoperabilidad, dado que se facilita la libre elección de la herramienta CAD. La adaptación del formato DXF para la coordinación entre la notaría y el técnico implica normalizar las secciones de cabecera (*HEADER*) y de definición de capas (*TABLES*). A nivel de cabecera se reserva un conjunto de variables para el almacenamiento de metadatos del expediente (véase Tabla 1), mientras que en la sección de capas se define (i) una capa de sólo lectura para los perímetros procedentes de la base cartográfica de referencia y (ii) otra de lectura/escritura para la delimitación de lindes por parte del técnico.

Tabla 1. Variables de cabecera del DXF que normalizan el intercambio de datos entre notaría y técnico.

Variable de sección HEADER	Descripción
\$USERI1	Código de notaría
\$USERI2	Nº de expediente
\$USERI3	Nº de operación
\$USERI4	Código del municipio de la notaría
\$USERR1	Metadatos

### 4.2. Servicios

La especificación Web Map Service (WMS) es un protocolo estandarizado para la publicación sobre HTTP de imágenes georeferenciadas desde repositorios de información geográfica (*OGC, 2014*). La IDEE cuenta con un amplio catálogo de servicios WMS de libre acceso mantenidos por organismos públicos.

El visor de cartografía del gestor de expedientes incorpora un cliente basado en el proveedor de datos WMS de SharpMap y configurable por código. De forma predeterminada el cliente consume (i) el servicio de Catastro, por ser la fuente de datos y la base cartográfica de referencia, y (ii) el del Plan Nacional de Ortofotografía Aérea, útil en el proceso de identificación visual de fincas. El acceso a ambos está integrado en la TOC de la ventana de mapa y el *feedback* de los usuarios confirma que es un factor clave para la usabilidad del aplicativo.



## 5. Resultados y conclusiones

El documento de análisis de requerimientos para aplicaciones de geoprocesamiento catastral debería contemplar tres aspectos fundamentales: licencias, patrones de diseño e interoperabilidad. Seleccionadas adecuadamente, las licencias libres permiten liberar el código a la vez que preservar los derechos de propiedad intelectual. Se debe planificar el uso de patrones de diseño que faciliten la reutilización y testabilidad del código. Por último, la interoperabilidad de los datos es un factor clave en el grado de usabilidad de la aplicación final.

Los desarrollos de software del Proyecto Ramón Llull se basan en librerías FLOSS propias o de terceros que cubren de manera eficiente y sin restricciones técnicas aspectos como el acceso a datos, el análisis espacial, el acceso a servicios WMS, el renderizado cartográfico o las salidas impresas. Esto ha permitido ganar tiempo y centrar los esfuerzos en el desarrollo del modelo de datos propiamente dicho, la operativa de validación de alteraciones catastrales, la lógica de colaboración entre usuarios y el interfaz de usuario. Por otra parte, la elección de licencias permisivas como LGPL preserva la propiedad intelectual del financiador sobre el producto final y a la vez abre el código a otros desarrolladores.

Los patrones de diseño son fundamentales en la planificación y el desarrollo de un proyecto de la envergadura del gestor de expedientes analizado. La aplicación supera las cien mil líneas de código, repartidas entre más de cincuenta librerías, y la implementación de los diferentes patrones descritos ha permitido (i) extender el modelo de geoprocesamiento, (ii) hacer modificaciones en la interfaz de usuario sin condicionar la funcionalidad del modelo, (iii) diseñar un visor cartográfico fácilmente extensible e (iv) introducir tests de unidad para controlar la calidad del nuevo código y prevenir regresiones en el código estable. Este tipo de decisiones en cuanto a la arquitectura del software tendrá un impacto favorable de cara a futuros desarrollos. Particularmente, el uso del patrón DI facilitará la extensión del modelo de datos para incorporar parcialmente la norma ISO 19152 LADM o Land Administration Model (International Organization for Standardization, 2012), tarea prevista en fases subsiguientes del Proyecto Ramón Llull. Esta norma, aunque es prescriptiva y de carácter general, se puede ampliar o modificar según el marco legal de cada país, y en nuestro caso se analizará cómo adaptar el paquete Land Administration para transacciones de propiedades inmobiliarias entre particulares.

Se ha aprovechado la especificación DXF de manera que, sin abandonar su entorno CAD habitual y siguiendo convenciones simples, el técnico puede interoperar con el expediente de alteración catastral usando directamente el fichero de dibujo. Por último, el uso de servicios WMS como apoyo a la identificación y a la representación cartográfica de una finca en su contexto ha repercutido favorablemente en la usabilidad de la aplicación y ha mejorado la experiencia de los usuarios de perfil jurídico formación previa en TIG.

## Agradecimientos

Este trabajo se enmarca dentro del proyecto de investigación Ramón Llull dirigido por Alfredo Ramón Morte, profesor titular del Departamento de Análisis Geográfico Regional y Geografía Física, y por Antonio Jiménez Clar, profesor asociado del Departamento de Derecho Civil, y ha sido financiado por el Colegio Notarial de Valencia. Además, se inscribe en el marco del proyecto de investigación «El Registro de la Propiedad como instrumento vertebrador de la información territorial; datos espaciales, metadatos y Directiva INSPIRE (II)» (DER 2011-23321), financiado por el Ministerio de Economía y Competitividad.

Igualmente agradecemos el trabajo de todos los profesionales que han colaborado durante las sucesivas fases del proyecto.

## Referencias

- Conejo Fernández, C. & Virgós Soriano, L.I., 2001, SIGCA 2. Cartografía catastral digital, disponible para todos, *Mapping*, ISSN 1131-9100, N° 78, 2002, págs. 58-73
- Consejo Superior Geográfico, Geoportail IDEE, 2014, Directiva INSPIRE, Ministerio de Fomento, <http://www.idee.es/web/guest/europeo-inspire>
- Free Software Foundation, 2014, What is Free Software?, <https://www.gnu.org/philosophy/free-sw.html>
- Free Software Foundation, 2014, GNU Lesser General Public License v3.0, [www.gnu.org/licenses/lgpl.html](http://www.gnu.org/licenses/lgpl.html)
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J., 1995, Patrones de diseño: Elementos de software orientado a objetos reusable, Addison Wesley
- International Organization for Standardization, 2012 Land Administration Model (LADM), <http://www.iso.org>
- Llorens Cobos, F., Mira Martínez, J.M., Navarro Carrión, J.T. & Ramón Morte, A., 2005. Proyecto Ramón Llull : Sistema de gestión de alteraciones catastrales para las notarías de la Comunidad Valenciana. *I Jornadas De SIG Libre*, 1–12.
- Martin, R.C., 1996, The Dependency Inversion Principle, C++ Report, Vol. 8, Mayo 1996, SIGS Publications Group
- Ministerio de Hacienda y Administraciones Públicas, Ley de economía sostenible del 2/2011, Boletín Oficial del Estado, núm. 55, del 5 de marzo de 2011 [www.boe.es/boe/dias/2011/03/05/pdfs/BOE-A-2011-4117.pdf](http://www.boe.es/boe/dias/2011/03/05/pdfs/BOE-A-2011-4117.pdf)
- Ministerio de Hacienda y Administraciones Públicas, 2014, Norma de entrega de actualizaciones de la cartografía catastral rústica y urbana, Dirección General del Catastro, [http://www.catastro.minhap.es/documentos/formatos\\_intercambio/formato\\_fxcc.pdf](http://www.catastro.minhap.es/documentos/formatos_intercambio/formato_fxcc.pdf)
- Ministerio de Hacienda y Administraciones Públicas, Texto refundido de la ley del Catastro del 1/2004, Boletín Oficial del Estado, núm. 58, 30 de octubre del 2013, [www.boe.es/buscar/act.php?id=BOE-A-2004-4163](http://www.boe.es/buscar/act.php?id=BOE-A-2004-4163)
- Montoro Rueda, R. M., 2011, Sobre el catastro parcelario: su necesidad y funciones en el comercio jurídico y la urgencia de su conexión con el registro de la propiedad, *Revista Crítica de Derecho Inmobiliario* ISSN: 0210-0444, Núm. 728, Noviembre 2011
- Navarro Carrión, J. T. & Ramón Morte, A., 2013, Importancia de los procesos de validación topológica en la gestión de alteraciones catastrales, *Investigaciones Geográficas*. 2013, 60: 117-138. doi:10.14198/INGEO2013.60.07
- Prasanna, D.R., 2009, Dependency injection, Manning Publications
- Pivotal Software, Inc, 2014, Spring.NET Application Framework, [springframework.net/](http://springframework.net/)
- Open Source Initiative, 2014, License MIT, <http://opensource.org/licenses/MIT>
- Stallman, R. M. (1999). The GNU operating system and the free software movement. En C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: Voices from the open source revolution* (pp. 53–50). Sebastopol: O'Reilly.
- Steiniger, S., & Hunter, A. J. S. (2013). The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39, 136–150. doi:10.1016/j.compenvurbysys.2012.10.003
- The Apache Software Foundation, 2014, “Apache License 2.0”, [www.apache.org/licenses/LICENSE-2.0.html](http://www.apache.org/licenses/LICENSE-2.0.html)