

TRABAJO DE FIN DE MÁSTER

Cálculo de métricas del paisaje a partir del SIOSE: Una propuesta escalable basada en Postgres/PostGIS

Andrea Rosado Abad

Directores: Raquel Montorio Llovería y Daniel Borini Alves

*Máster Universitario en Tecnologías de la Información Geográfica para
la Ordenación del Territorio: Sistemas de Información Geográfica y
Teledetección*

Universidad de Zaragoza

Noviembre 2017

Agradecimientos

Este Trabajo de Fin de Máster me ha dado la oportunidad de asentar los conocimientos adquiridos durante el último curso en Zaragoza. Nada de esto hubiera sido posible sin el apoyo y la guía de muchas personas, a las que me gustaría agradecer el tiempo y la dedicación que me han regalado/proporcionado/dado a lo largo de esta etapa.

En primer lugar, quiero dar las gracias a los profesores Raquel Montorio Llovería y Daniel Borini Alves de la Universidad de Zaragoza, directores de este trabajo, por su atención y el esfuerzo que han hecho en cualquier circunstancia. También me gustaría agradecer al profesor Alfredo Ramón Morte por darme la oportunidad de volver a realizar, por segundo año consecutivo, las prácticas de empresa en el Laboratorio de Geomática de la Universidad de Alicante. Esta estancia, sin duda, ha influido en la forma final de este trabajo y posiblemente en mi futuro profesional.

Durante mi paso por el Laboratorio de Geomática, he trabajado y aprendido con muchos profesionales, ahora amigos, que me han hecho sentir como si fuera una compañera más. Entre otros compañeros, Benito Zaragozí, Pepe Torres y J.T. Navarro, con los que he estado todo el tiempo, me gustaría expresarles todo mi agradecimiento por su colaboración y paciencia a lo largo de las prácticas, como también por su apoyo y ayuda cuando más lo necesitaba.

En este punto, creo que es cuando más puedo valorar todo lo aprendido en las distintas asignaturas del máster. Entre todas las asignaturas, hay algunas que han resultado imprescindibles para que yo pudiera sacar el máximo partido de esta primera experiencia profesional. Quiero agradecer las asignaturas de ... (usos del suelo, geodatabases, programación, etc)

Gracias también a mis amigos y compañeros del máster, que han hecho que esta etapa sea inolvidable y me han contagiado con sus ganas de seguir aprendiendo.

Finalmente, mi familia siempre ha estado ahí...

Prólogo

Prácticas externas

Este trabajo se ha realizado en el marco del convenio de prácticas de empresa entre la Universidad de Zaragoza y el Laboratorio de Geomática del Instituto Interuniversitario de Geografía de la Universidad de Alicante. El periodo de prácticas ha tenido una duración desde julio hasta noviembre de 2017, sumando un total de 440 horas presenciales.

En el Laboratorio de Geomática se desarrollan actualmente varios proyectos de investigación, entre ellos el Sistema de Información Geográfica de la Universidad de Alicante (SIGUA) es el proyecto de más largo recorrido. SIGUA se puso en marcha en 1997, lo que supone que **los expertos del laboratorio cuentan con más de 20 años de experiencia en el diseño y gestión de SIG corporativos**. A esta experiencia hay que sumarle numerosos desarrollos de aplicaciones y colaboraciones en otros proyectos de geografía aplicada. Cabe mencionar que el equipo del Laboratorio de Geomática está formado por licenciados, ingenieros y doctores, tanto en Geografía como en Informática, los cuales desarrollan su trabajo en las Tecnologías de la Información Geográfica basadas en *software libre*.

Actualmente, el Laboratorio de Geomática reparte sus esfuerzos entre el mantenimiento e innovación de SIGUA y un proyecto de investigación oficial conocido por su acrónimo como SIOSE-INNOVA. Se trata de un proyecto de alcance internacional a través de la **colaboración con el Instituto Geográfico Nacional (IGN)**.

Proyecto SIOSE-INNOVA

El presente Trabajo Fin de Máster en Tecnologías de la Información Geográfica para la Ordenación del Territorio: SIG y Teledetección, detalla todas las tareas desarrolladas durante la participación en el Proyecto SIOSE-INNOVA: Innovaciones técnicas y metodológicas en el Sistema de Información sobre Ocupación del Suelo de España (SIOSE) y su aplicación en

estudios geográficos. El Investigador Principal de este proyecto es el profesor Alfredo Ramón Morte, quien dirige y coordina una investigación, en la que **colaboran varias universidades junto con el Servicio de Ocupación del Suelo del Instituto Geográfico Nacional (IGN), que es el equipo responsable de la base de datos del SIOSE.**

SIOSE-INNOVA es un proyecto de investigación financiado por el Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, dentro del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016. Los objetivos principales de este proyecto tienen **una parte innovadora**, que consiste en comprobar qué tecnologías *NoSQL* (no sólo SQL) pueden aportar mejores soluciones para la explotación de la base de datos del SIOSE, y **una parte aplicada**, que consiste en poner en práctica las nuevas tecnologías en casos de estudios reales.

Durante el desarrollo del proyecto SIOSE-INNOVA, se quieren alcanzar los siguientes objetivos específicos:

1. Crear un marco de experimentación *reproducible y fácilmente utilizable* por un gran número de usuarios.
2. Analizar las necesidades y rendimiento de distintas tecnologías de **bases de datos NoSQL para la explotación del SIOSE**.
3. Desarrollar e implementar **un nuevo modelo de datos auxiliar** que permita extender las posibilidades de análisis del SIOSE con técnicas de Big Data o Data Mining.
4. **Evaluar la usabilidad de los datos SIOSE** en distintas plataformas tecnológicas, mediante su aplicación en casos de uso reales en los que utilizar datos de ocupación del suelo resulte esencial.

En la introducción de este trabajo se explica el gran potencial del SIOSE, junto a las dificultades que surgen en bases de datos de ocupación del suelo de estas dimensiones y complejidad. **El proyecto SIOSE-INNOVA pretende facilitar el uso de la información que contiene el SIOSE.**

Concretamente, este Trabajo Fin de Máster se enmarca dentro del proyecto SIOSE-INNOVA cubriendo la primera etapa del proyecto, en la que se plantea **una plataforma tecnológica** lo suficientemente potente como **para analizar el SIOSE de una manera ágil, intuitiva e interactiva**.

Debido a su fuerte componente tecnológico, es posible que la lectura de este trabajo pueda resultar árdua y compleja en algunos apartados. Sin embargo, es importante **no perder de vista la aplicabilidad final del trabajo realizado**. Así pues, se ha desarrollado una nueva extensión sobre PostgreSQL/PostGIS, denominada *pg_landmetrics*, capaz de calcular métricas del paisaje a partir de la base de datos del SIOSE, superando así determinados problemas de *usabilidad* y de escalabilidad (gran volumen de datos). La plataforma de desarrollo descrita en este trabajo (*git, dockers, pgxn, etc*), **promueve la reproducibilidad** de esta investigación y asegura **que otros usuarios sean capaces de utilizar lo que aquí se plantea** de una manera lo más sencilla posible, quizás similar a descargar y usar una *app* para teléfonos inteligentes.

El desarrollo del presente trabajo gira entorno a **un único caso de uso** que sirve para valorar la *usabilidad* de todo lo que se desarrollará en el contexto del proyecto SIOSE-INNOVA. Este caso de uso consiste en crear **un visor de cartografía web** en el que se puedan consultar y analizar los datos del SIOSE. En este trabajo se plantea que dicho visor sirva para **calcular métricas del paisaje a partir de los datos del SIOSE**. A modo de ejemplo, en la Figura 1 se describe una posible aplicación para calcular las métricas del paisaje de dos zonas geográficas distintas y así comparar su estructura. Cabe adelantar que la experiencia computacional descrita en el Capítulo 3 simula este tipo de aplicación.

Lo deseable en una aplicación como la que descrita (ver Figura 1) es poder seleccionar una región de España o áreas más pequeñas y **consultar el SIOSE de un modo directo**, sin necesidad de descargar pesados ficheros SIG (ESRI Shapefile), unirlos, procesarlos y obtener los resultados tras un proceso relativamente costoso. El tiempo de respuesta de una aplicación similar es muy importante para la experiencia de los usuarios y también para conseguir dar servicio a un gran número de usuarios.

La extensión *pg_landmetrics* encapsula consultas SQL más complejas y **hace posible calcular múltiples métricas del paisaje en sentencias de una o muy pocas líneas**. Además, esta extensión se instala de un modo sencillo y **permite trabajar con bases de datos voluminosas** como la del SIOSE (millones de registros; Gigabytes de memoria; ver Tabla 2.1).

Una cuestión que va más allá de los objetivos de este trabajo tiene que ver con el verdadero potencial de la plataforma de *contenerización* con la que se ha desarrollado *pg_landmetrics*. Al tratarse de un *software libre* y *contenerizado* (ver capítulo 2), se facilita la distribución de esta aplicación a otros equipos (servidores y/o PCs), lo cual **aporta una escalabilidad que ningún entorno de escritorio puede lograr**. Es decir que a más usuarios del SIOSE,

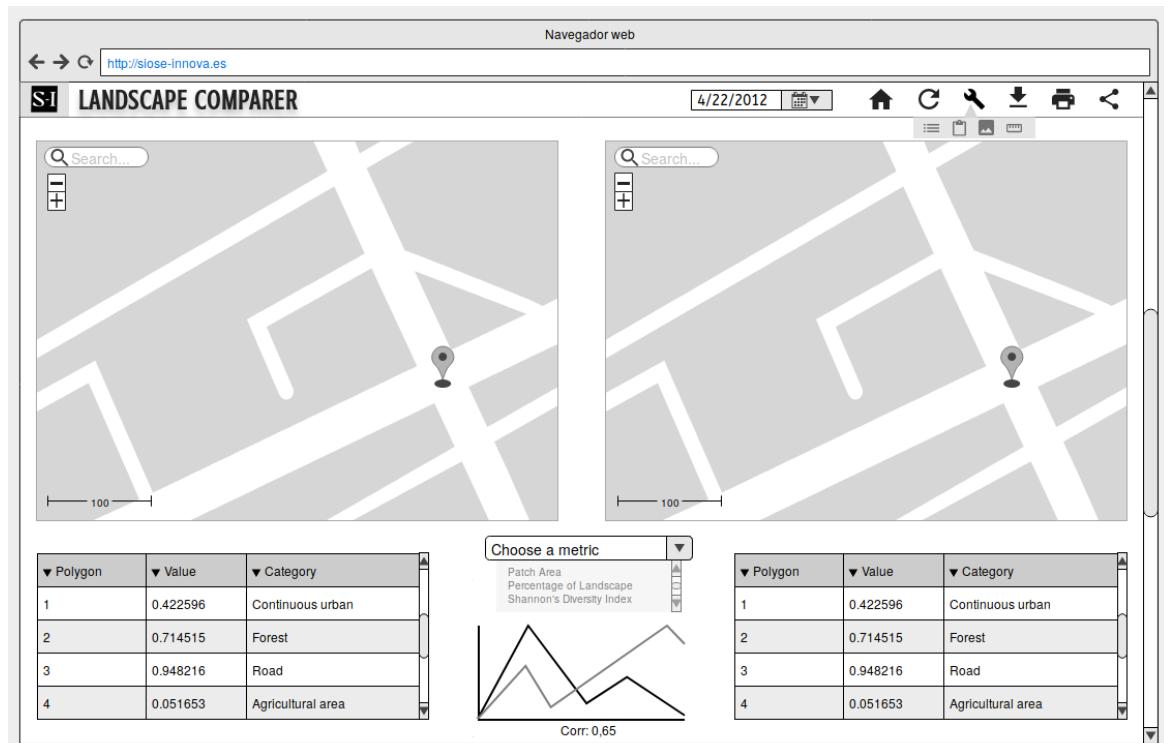


Figura 1 Prototipo de un visor cartográfico para el análisis (comparación) de la estructura del paisaje a partir del SIOSE. La extensión sobre PostgreSQL/PostGIS desarrollada en este trabajo (*pg_landmetrics*) es la base para este tipo de aplicaciones.

resultaría sencillo añadir nuevos servidores para realizar el análisis (p. ej. con una única instrucción sería: `docker-compose scale web=2 database=3`).

Estructura del trabajo

Este trabajo se organiza en cuatro capítulos, a parte las referencias bibliográficas y los anexos. De un modo general, la estructura seguida es la siguiente:

- En el capítulo 1 se revisa el uso de las bases de datos de ocupación del suelo y su papel en el análisis de la estructura del paisaje a partir de métricas de paisaje. Al final se presentan los objetivos generales y específicos de este trabajo.
- En el capítulo 2 se describen los conjuntos de datos, herramientas, plataformas tecnológicas y metodología seguida para diseñar e implementar una nueva extensión sobre PostgreSQL/PostGIS. La metodología incluye desde el trabajo colaborativo en varias plataformas de desarrollo, hasta las tareas diarias, la incorporación de funciones y la documentación de la extensión. Finalmente, se plantean una serie de experiencias computacionales para evaluar la extensión de acuerdo con los objetivos del trabajo.
- En el capítulo 3 se detallan todos los resultados obtenidos, tanto en forma de código, como aquellos resultados obtenidos en un caso de estudio que simula una aplicación real (ver Figura 1).
- Finalmente, en el capítulo 4 se revisa el trabajo realizado para valorar cómo se han alcanzado los objetivos propuestos en la introducción. El capítulo termina por detallar los próximos pasos que seguirá el equipo de desarrollo para completar una extensión en *fase de producción* (p. ej. un visor publicado desde una página web del IGN).

Resumen

Este trabajo demuestra que es posible hacer un uso más ágil y directo de las últimas bases de datos de ocupación del suelo. Estas bases de datos son cada vez más voluminosas, por lo que se ven en la necesidad de utilizar modelos de datos relativamente complejos para manejar tanta información. En este caso se ha analizado esta cuestión sobre el caso del SIOSE, pero las tendencias en este campo hacen que cuanto se desarrolla en las próximas páginas sea extrapolable a otras geodatabases europeas e internacionales.

SIOSE es el Sistema de Información sobre Ocupación del Suelo de España. La base de datos del SIOSE contiene un gran volumen de información (decenas de millones de registros) con un gran potencial de aplicación. Además, esta base de datos implementa un modelo de datos *orientado a objetos* que aporta una **mayor riqueza descriptiva de la que se da en otras bases de datos de ocupación del suelo anteriores**. Así pues, mientras que en otras bases de datos se da una correspondencia de una etiqueta descriptiva por cada polígono, en el SIOSE cada polígono puede contener toda la información descriptiva que sea necesaria (decenas de etiquetas descriptivas o más). No obstante, **esta gran cantidad de información y la complejidad del modelo de datos del SIOSE, se manifiestan en problemas de usabilidad de esta información por parte de los usuarios**. Cabe anticipar que, según aumenta la capacidad para generar nueva información geográfica, dichos problemas irán en aumento en los próximos años.

Este trabajo se enmarca dentro de los objetivos propuestos por el Proyecto SIOSE-INNOVA (CSO2016-79420-R). Este proyecto plantea dos líneas de trabajo complementarias, una de innovación técnica y otra aplicada, que buscan potenciar la *usabilidad* del SIOSE en distintos tipos de estudios (biodiversidad, riesgos naturales, análisis paisajístico, estudios de accesibilidad, etc).

Entre el gran número de aplicaciones posibles a partir de bases de datos como la del SIOSE, **las métricas del paisaje son utiles para analizar la estructura y comportamiento del paisaje**. Actualmente, existen muchos *softwares* diseñados para ofrecer cálculos y análisis

de patrones de paisaje a partir de bases de datos de ocupación del suelo (FRAGSTATS, Patch Analyst, etc). Sin embargo, dada la gran diversidad de métricas posibles, ningún programa permite calcular *todas* las métricas, ni está pensado para trabajar con las geodatabases actuales, más complejas y voluminosas que hace unos años, lo cual se refiere también a los problemas de *usabilidad* relacionados con el SIOSE (volumen y complejidad).

El objetivo principal de este trabajo es crear una extensión sobre PostgreSQL-/PostGIS capaz de calcular métricas del paisaje a partir de la base de datos del SIOSE, haciendo frente a los mencionados problemas de *usabilidad* (volumen de datos y complejidad de consultas). **La pregunta central de este trabajo es si esta extensión** (en combinación con otras nuevas tecnologías), **será capaz de hacer frente a los mencionados problemas de *usabilidad* que afectan al SIOSE y a otras geodatabases similares.**

El desarrollo de esta extensión, denominada *pg_landmetrics*, ha sido posible gracias al uso de herramientas de desarrollo colaborativo (*Git*, *GitHub*) y al uso de plataformas de contenerización o virtualización de servicios (*Docker*, *DockerHub*). Estas herramientas han permitido establecer flujos de trabajo de *integración continua* e ir generando, paso a paso, una extensión funcional que es capaz de calcular métricas del paisaje a partir de la base de datos del SIOSE-2011. Se han utilizado toda una variedad de técnicas de programación en bases de datos para que los usuarios de esta extensión sean capaces de calcular un gran número de métricas del paisaje en sentencias de una o unas pocas líneas.

Finalmente, para poner a prueba la extensión desarrollada se ha llevado a cabo **una experiencia computacional completa sobre la base de datos del SIOSE-2011**. Esta experiencia ha consistido en simular repetidas consultas de métricas por parte de un usuario sobre dos grandes áreas y a distintas escalas (ver Figura 1). Dado que estas consultas se realizan en pocos segundos en un PC, **los resultados son prometedores** y es seguro que mejorarán cuando este *software* funcione en un servidor de Internet o sea distribuido en *La Nube*.

Globalmente, este trabajo sirve para poner en valor el **trabajo colaborativo** basado en una serie de herramientas de control de versiones, contenerización y orquestación, que pueden ser aplicadas en distintos contextos. Esta metodología facilitará enormemente seguir añadiendo nuevas métricas del paisaje y aplicar esta extensión en nuevos estudios relacionados con la estructura del paisaje.

Palabras clave: SIOSE, *usabilidad*, métricas de paisaje, PostGIS, reproducibilidad, contenerización

Abstract

Landscape metrics are used to analyzed the structure and behavior of the landscape as well as the temporary modifications, either by natural or human factors. Given the utility for a variety of applications, there are many softwares designed to offer calculations and analysis of landscape patterns. The main objective of this work is to create a reproducible and extensible PostgreSQL/PostGIS extension capable of calculating landscape metrics for vector input data. Later, this extension should make it possible to add new metrics and investigate new studies related to landscape structure. Finally, the implementation of the extension is valued because it requires a collaborative work methodology based on a series of tools for containerization and orchestration.

Key words: landscape metrics, extension, reproducibility, containerization, orchestration.

Índice general

Índice de figuras	xvii
Índice de tablas	xxi
Ejemplos de código	xxii
1. Introducción	1
1.1. El SIOSE como fuente para el estudio de la estructura del paisaje	2
1.2. Métricas de paisaje y software que las calcula	5
1.3. Objetivos	7
2. Metodología	9
2.1. Integración continua y desarrollo colaborativo	10
2.1.1. Control de versiones	11
2.1.2. Contenerización y orquestación de servicios	15
2.1.3. Extensibilidad en PostgreSQL	17
2.1.4. Otras aplicaciones	19
2.2. Conjuntos de datos	21
2.3. Selección de métricas	24
2.4. Desarrollo de funciones en PostgreSQL	26

2.5. Experimento sobre la capacidad de cálculo de <i>pg_landmetrics</i>	32
2.6. Documentación de la extensión	33
3. Resultados y Discusión	35
3.1. pg_landmetrics	36
3.2. Caso de uso sobre el SIOSE-2011	39
4. Conclusiones y trabajo futuro	45
Apéndice A. Funciones tipo	47
Apéndice B. Resultados finales	55
Apéndice C. Documentación SVG	65

Índice de figuras

1.	Prototipo de un visor cartográfico para el análisis (comparación) de la estructura del paisaje a partir del SIOSE. La extensión sobre PostgreSQL/PostGIS desarrollada en este trabajo (<i>pg_landmetrics</i>) es la base para este tipo de aplicaciones.	VIII
1.1.	Riqueza descriptiva del modelo de datos orientado a objetos del SIOSE frente a una clasificación jerárquica. Fuente: www.siose.es	3
1.2.	Captura de pantalla del diálogo inicial de FRAGSTATS 4.0. Fuente: www.landscapetoolbox.org	6
2.1.	Estadísticas de la comunidad de desarrolladores de PostGIS en la plataforma GitHub. Paul Ramsey es un desarrollador muy activo también en foros de usuarios.	12
2.2.	Diagrama de secuencia en el que se describe el flujo de integración continua más habitual en este trabajo (varias veces al día). Cada vez que se realizaban cambios importantes en la extensión <i>pg_landmetrics</i> , se añadían al control de versiones local y al de <i>La Nube de GitHub</i>	14
2.3.	Diagrama de secuencia sobre el desarrollo colaborativo entre repositorios. Aproximadamente una vez al día, se proponen una serie de cambios al repositorio oficial (<i>pull request</i>), si estos cambios son aceptados después es posible sincronizar los dos repositorios.	16

2.4. Diagrama de secuencia. Integración continua en la implementación y desarrollo de funciones SQL con <i>dockers</i> . Cada vez que una nueva función es añadida a la base de datos, se elimina cualquier rastro de la versión anterior y se vuelve a lanzar un nuevo contenedor para evaluar si los cambios son válidos.	18
2.5. Vista de la interfaz web de PgAdmin4.	20
2.6. Coberturas del suelo del paisaje de ejemplo. Estas geometrías se distribuyen junto con la extensión <i>pg_landmetrics</i> para que los usuarios realicen pruebas antes de importar sus propios datos.	22
2.7. Ejemplo de cálculo de AREA, NCORE y TCA a partir de la geometría del SIOSE 2011.	31
3.1. Calendario de actividad y contribuciones de este trabajo al proyecto <i>pg_landmetrics (heatmap calendar)</i>	36
3.2. Diagrama de red de GitHub.	38
3.3. Calendario de actividad y contribuciones de este trabajo al proyecto <i>pg_landmetrics (heatmap calendar)</i>	42
3.4. Calendario de actividad y contribuciones de este trabajo al proyecto <i>pg_landmetrics (heatmap calendar)</i>	43
3.5. Calendario de actividad y contribuciones de este trabajo al proyecto <i>pg_landmetrics (heatmap calendar)</i>	44
B.1. Título.	56
B.2. Título.	57
B.3. Título.	58
B.4. Título.	59
B.5. Título.	60
B.6. Título.	61
B.7. Título.	62

B.8. Título.	63
B.9. Título.	64

Índice de tablas

2.1. Características de los conjuntos de datos utilizados.	24
2.2. Métricas de paisaje disponibles en la extensión.	25
3.1. Listado de las métricas de paisaje disponibles en la extensión, según su nivel de complejidad y si han sido aceptadas en el repositorio oficial.	40

Ejemplos de código

2.1.	Consultas SQL para la métrica AREA.	27
2.2.	Función simple para calcular la métrica AREA por polígonos.	27
2.3.	Ejemplo de uso para calcular la métrica AREA.	28
2.4.	Consulta SQL para la métrica TCA.	29
2.5.	Función de agregado para calcular TCA por categorías.	29
2.6.	Ejemplo de uso para calcular la métrica TCA.	30
2.7.	Extracción de los resultados de las métricas en diferentes escalas de referencia a partir de grids.	32
A.1.	Función para calcular AREA de un polígono de PostGIS de tipo <i>geography</i>	47
A.2.	Función para calcular CA por categorías.	49
A.3.	Función para calcular TA del paisaje.	52
C.1.	Función para pintar gráficos vectoriales escalables (SVG).	65

1. INTRODUCCIÓN

La interacción de los factores naturales y antrópicos es el origen de la estructura espacial compleja y heterogénea que presenta el paisaje (??). En las últimas décadas, la ecología del paisaje ha estudiado la configuración, el tamaño y la forma de los componentes que estructuran el territorio utilizando **métricas de paisaje** (?). Hoy en día, para estudiar la estructura del paisaje se dispone de **gran cantidad de información y herramientas**.

Las bases de datos de ocupación del suelo, como por ejemplo el Sistema de Información de Ocupación del Suelo en España (SIOSE), representan el territorio de un modo adecuado para aplicar conceptos fundamentales como los de conectividad o diversidad del paisaje. Además, estas bases de datos aumentan progresivamente en riqueza semántica y resolución geométrica, por lo que **la información disponible no para de crecer**.

Complementariamente, también existe software específico o de carácter más general que facilita el cálculo de métricas del paisaje. Sin embargo, **ninguna de las aplicaciones encontradas es lo suficientemente escalable y extensible como para analizar grandes bases de datos de ocupación del suelo tan complejas como las actuales** (p.ej. SIOSE).

Evidentemente, los problemas existentes al analizar las geodatabases actuales irá en aumento con la cada vez mayor disponibilidad de datos obtenidos a partir de imágenes de satélite o datos de campo. Este trabajo se enmarca en este contexto de creciente complejidad y busca **proponer herramientas más sencillas y eficientes en el cálculo de métricas del paisaje**.

1.1. El SIOSE como fuente para el estudio de la estructura del paisaje

Puntos de interés:

- El SIOSE es una valiosa **base de datos de ocupación del suelo** que contiene un gran volumen de información territorial de toda España.
- Desde su aparición en 2005, SIOSE se ha convertido en un repositorio de referencia para sus homólogos europeos, llegando a ser un **modelo para la iniciativa EAGLE (SIOSE europeo)**.
- A pesar de su gran potencial, el SIOSE presenta ciertos problemas de *usabilidad* debidos a su gran volumen y complejidad.

El Sistema de Información de Ocupación del Suelo en España (SIOSE) se lanzó en el año 2005 por la Dirección General del Instituto Geográfico Nacional de España (IGN)¹ ante la necesidad de adquirir información más detallada a nivel nacional. El SIOSE está integrado en el Plan Nacional de Observación del Territorio (PNOT) con el objetivo de alcanzar una infraestructura de datos espaciales multidisciplinar. Este conjunto de datos va a ser un componente imprescindible para llevar a cabo los objetivos de este trabajo.

El SIOSE es una base de datos que recoge información de la ocupación del suelo de España en forma de malla continua de polígonos a partir de la fotointerpretación de imágenes. Cada polígono se especifica por dos componentes: la cobertura del suelo (*Land Cover, LC*) se refiere a las características de la cubierta natural, como por ejemplo cuerpos de agua, bosques, superficies urbanas, zonas agrícolas, etc., y el uso del suelo (*Land Use, LU*) se define por las funciones socioeconómicas en el territorio, como por ejemplo uso industrial, residencial, forestal, agrícola, etc.

La escala de referencia es 1:25.000 y el sistema geodésico de referencia es European Terrestrial Reference System 1989 (ETRS89) con proyección Universal Transversa de Mercator (UTM). El tamaño mínimo de los polígonos depende del tipo de cobertura: 2 Ha para las zonas agrícolas, forestales y naturales, 1 Ha para las superficies artificiales y 0,5 Ha para agua, cultivos forzados, coberturas húmedas, playas, vegetación de ribera y acantilados. La base de datos del SIOSE ha sido diseñada utilizando un **modelo orientado a objetos** que

¹<http://www.ign.es/web/ign/portal>

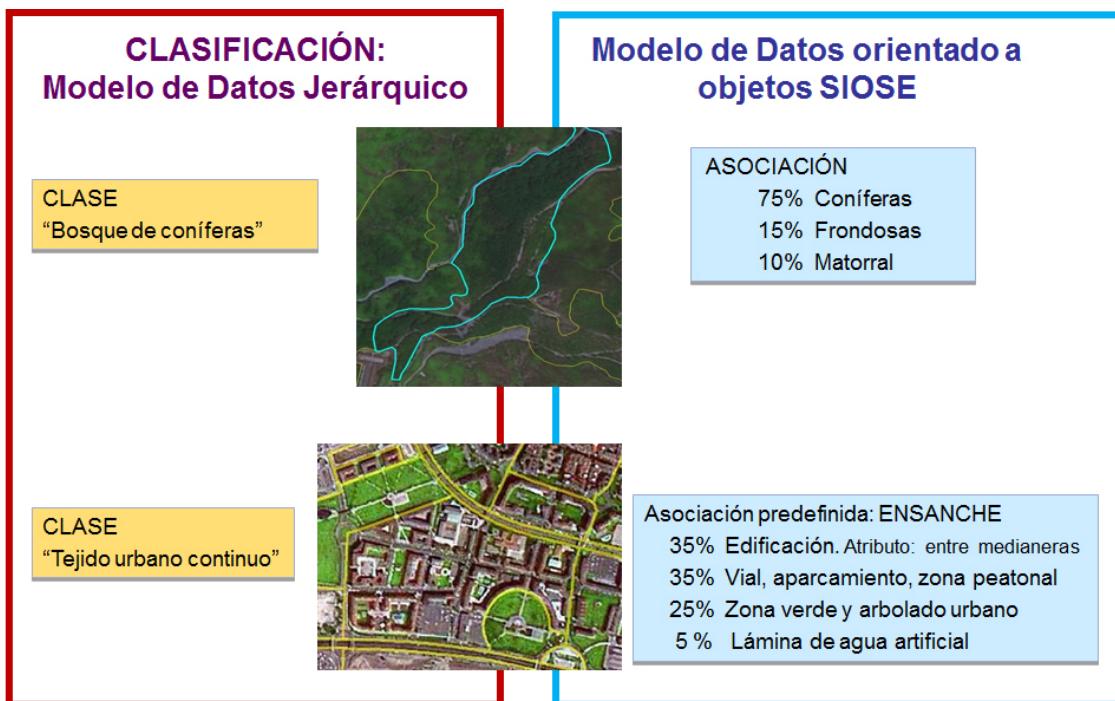


Figura 1.1 Riqueza descriptiva del modelo de datos orientado a objetos del SIOSE frente a una clasificación jerárquica. Fuente: www.siose.es.

describe los objetos, atributos y relaciones de los elementos del territorio. Esto permite la asignación de una o varias coberturas de suelo a un único polígono (datos semiestructurados). Cuando el polígono presente una única cobertura tendrá una *cobertura simple*, pero cuando esté formado por dos o más coberturas tendrá una *cobertura compuesta*, o también conocido como multietiqueta o *multilabel* (?). El hecho de que sea un modelo orientado a objetos, garantiza la compatibilidad y comparabilidad con otras bases de datos de ocupación del suelo como por ejemplo el *Corine Land Cover (CLC)*. Sin embargo, físicamente, este modelo se ha descrito como un modelo **entidad-relación** para poder implementarlo también en bases de datos espaciales relacionales, que son las más utilizadas de hoy en día y las más habituales en los SIG.

El SIOSE tiene una proyección internacional ya que hay iniciativas similares en otros muchos países. Concretamente, el grupo EAGLE (Eionet Action Group on Land monitoring in Europe) tiene como objetivo solucionar la vigilancia de la tierra sobre la información europea de las fuentes de datos nacionales para una mejor integración y armonización a partir del concepto *bottom-up*, además de facilitar el intercambio y comparación de datos entre países europeos (?).

El Instituto Geográfico Nacional de España (IGN) creó el SIOSE muy tempranamente (2005) siguiendo los últimos estándares internacionales, por lo que con el tiempo se ha convertido en un repositorio de ocupación del suelo de referencia a nivel europeo (?). Del mismo modo, uno de los objetivos futuros del SIOSE es avanzar hacia una base de datos de ocupación del suelo europea para que resulte más fácil trabajar a nivel internacional.

El análisis de la estructura del paisaje a partir de datos de usos y coberturas del suelo ha sido aplicado habitualmente desde diversas disciplinas y ámbitos de estudio. Por ejemplo, se han realizado estudios sobre:

- Medio Ambiente y hábitats naturales (???), (???).
- Demografía, urbanismo y planificación del territorio (?), (?????).
- Infraestructuras, energía y transporte.
- Dinámica de la ocupación del suelo (???), (????).

También se han realizado estudios sobre abandono agrícola (?) y otros sobre el riesgo de incendio asociado a nuevas formas de ocupación del suelo ?.

Los principales usuarios que trabajan con información sobre ocupación del suelo son la Administración General, gobiernos autonómicos, universidades, organismos de investigación, organismos europeos e internacionales, empresas públicas y privadas y, en menor medida, los usuarios particulares. Todos estos usuarios del SIOSE se ven afectados por dos dificultades relacionadas con la *usabilidad* de los datos: **el gran volumen de datos y la complejidad del modelo de datos**. La base de datos está formada por unos 2,5 millones de geometrías poligonales con sus coberturas de suelo. Este volumen de datos influye de manera importante en la capacidad de los usuarios para consultar o manejar esta información. La complejidad del modelo de datos es mayor que en bases de datos más tradicionales. El modelo SIOSE se compone de 85 clases, que forman un total de 820.632 casos de coberturas de suelo diferentes (simples y compuestas (?)). Este nivel de complejidad del modelo de datos hace que el SIOSE sea difícil de utilizar por parte de usuarios que no conozcan el modelo o que no son especialistas en geodatabases. La gran cantidad de geometrías y la complejidad de las clasificaciones dificultan gestionar esta información mediante a aplicaciones SIG (Sistemas de Información Geográfica) convencionales, ya que se puede llegar a superar la capacidad de éstas. Todo ello hace que sea necesario estudiar otras nuevas tecnologías (?).

En el proyecto SIOSE-INNOVA se plantea investigar y proponer soluciones para los problemas de *usabilidad* descritos por el mismo equipo de desarrollo del SIOSE en ?. Durante

el desarrollo de este proyecto de tres años de duración, se quieren alcanzar los siguientes objetivos específicos:

1. Crear un marco de experimentación reproducible y fácilmente utilizable por un gran número de usuarios.
2. Analizar las necesidades y rendimiento de distintas tecnologías de bases de datos NoSQL para la explotación del SIOSE.
3. Desarrollar e implementar un nuevo modelo de datos auxiliar que permita extender las posibilidades de análisis del SIOSE con técnicas de *Big Data* o *Data Mining*.
4. Evaluar la *usabilidad* de los datos SIOSE en distintas plataformas tecnológicas, mediante su aplicación en casos de uso reales en los que utilizar datos de ocupación del suelo resulte esencial.

1.2. Métricas de paisaje y software que las calcula

Puntos de interés:

- Las **métricas de paisaje** son métodos cuantitativos que sirven para analizar la estructura del paisaje y otros fenómenos (p.ej. evolución del paisaje, conectividad de ecosistemas, entre otros).
- FRAGSTATS, Conefor Sensinode, Patch Analyst, entre otros, son aplicaciones de escritorio muy utilizadas para el cálculo de métricas del paisaje. No obstante, **no hay ninguna aplicación** que sea fácilmente **escalable** y **extensible** como para realizar análisis sobre una geodatabase similar a la del SIOSE.

El paisaje comprende la interacción entre los factores naturales y artificiales, causantes de la evolución y estructura compleja y heterogénea que presenta el suelo. Por este motivo, se utilizan las métricas de paisaje como técnica/metodología para el estudio del paisaje y otros fenómenos. Las métricas son métodos cuantitativos que funcionan como algoritmos matemáticos encargados de aportar resultados numéricos (?).

Hay cientos de métricas de paisaje, correlacionadas entre sí, pero no todas las métricas tendrán significado en todos los contextos y/o estudios. Algunas de las investigaciones que

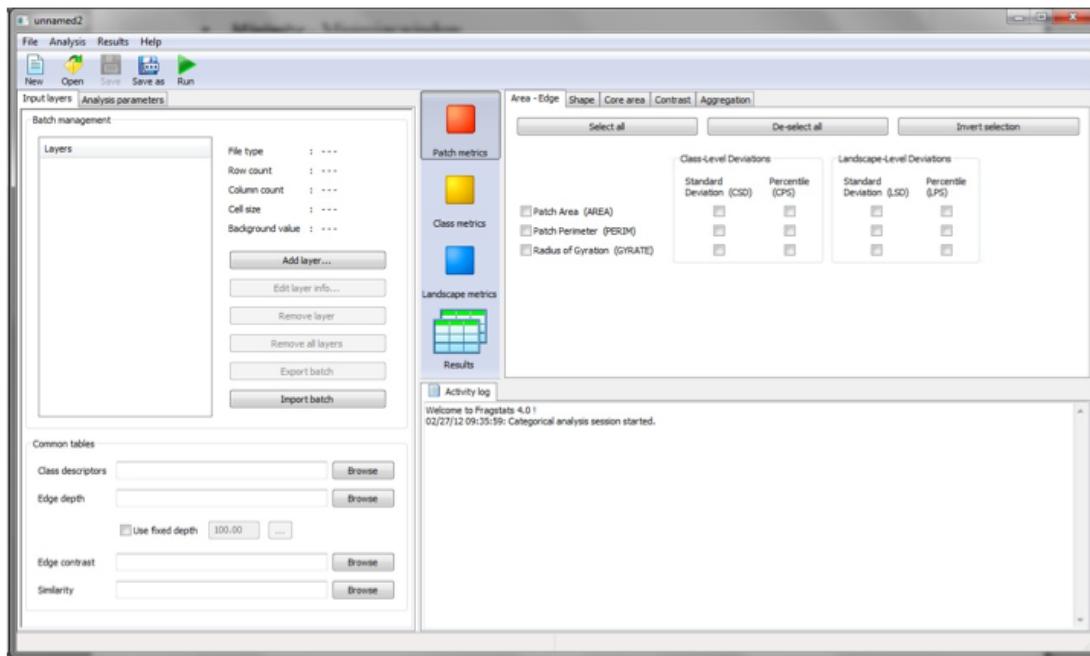


Figura 1.2 Captura de pantalla del diálogo inicial de FRAGSTATS 4.0. Fuente: www.landscapetoolbox.org

utilizan las métricas de paisaje son aquellas relacionadas con biodiversidad, hábitats, aplicaciones de agua, cambios de suelo, estructura urbana, infraestructura vial, riesgos naturales, estética del paisaje, planificación territorial, entre otros (?). Por ejemplo, en ? se investigan aquellas métricas que parecen estar más relacionadas con los estudios forestales para explicar las relaciones entre los procesos ecológicos y los patrones espaciales existentes en una zona. **Esto indica que los investigadores necesitarán calcular un gran número de métricas para cada paisaje y después aplicar algún criterio de selección para determinar cuales son las más descriptivas en cada caso.**

Las métricas de paisaje se pueden calcular a partir de aplicaciones de escritorio. En ? se establece una comparativa entre **programas específicos** para el cálculo de métricas del paisaje, destacando entre ellos FRAGSTATS (??). Por otro lado, hay otros programas como Conefor Sensinode (?), Patch Analyst, varios módulos de GRASS GIS, LecoS (?), ZonalMetrics (?), entre otros. Evidentemente, esta lista no puede estar completa, ya que hay muchos otros programas que pueden calcular un número de métricas variable según un gran número de factores.

Existen muchos otros programas especializados que permiten el cálculo de este tipo de métricas/índices pero no siempre tiene que ser una aplicación específica que las calcule.

También es posible calcular fácilmente determinadas métricas con **herramientas típicas de un SIG** de escritorio (calculadora de campos y/o calculadora raster, entre otras posibilidades).

En cualquier caso, no hay ninguna aplicación escalable y extensible diseñada para realizar análisis sobre bases de datos de ocupación del suelo tan voluminosas y complejas como lo es la del SIOSE. Por ejemplo, según sus especificaciones técnicas, FRAGSTATS 4.0 es un software de 32 bits, lo cual significa que puede utilizar un máximo de 2GB de memoria, tampoco está diseñado para trabajar en *La Nube* o con geometrías vectoriales.

1.3. Objetivos

El principal objetivo de este trabajo es **desarrollar una extensión (PostgreSQL/PostGIS) que facilite el cálculo de métricas de paisaje**. Se quiere facilitar la realización de consultas y permitir el manejo de bases de datos voluminosas como lo es la base de datos *actual* del SIOSE.

Previsiblemente, las bases de datos de ocupación del suelo no harán sino aumentar en volumen y complejidad, por lo que si hoy en día existen los mencionados problemas de *usabilidad* del SIOSE, estos no harán sino ir a más. De este modo el trabajo desarrollado tendrá continuidad en el tiempo.

Enlazando con el objetivo principal de este trabajo surgen una serie de objetivos más específicos relacionados con la metodología planteada en el proyecto SIOSE-INNOVA. En este sentido, se han considerado los siguientes **objetivos específicos**:

1. Aplicar herramientas de **desarrollo colaborativo** para trabajar con los otros investigadores del proyecto SIOSE-INNOVA.
2. Validar sistemáticamente que la extensión desarrollada funciona correctamente (*integración continua*; tests de unidad).
3. Aplicar las prácticas y estándares de desarrollo más novedosos.
4. Realizar un **experimento con una geodatabase de usos del suelo de gran complejidad y volumen**, como es el SIOSE (2011).
5. Adquirir **nuevos conocimientos** a lo largo de este trabajo sobre herramientas de desarrollo colaborativo, contenerización y orquestación, lenguajes de programación y lenguajes procedurales.

6. Poner en práctica los **conocimientos adquiridos durante el aprendizaje en el máster** de las distintas asignaturas impartidas sobre teoría e implementación de bases de datos, lenguajes de programación, software libre, aplicaciones infográficas y análisis espacial.

2. METODOLOGÍA

Este trabajo aplica una metodología de **integración continua y desarrollo colaborativo** de acuerdo con los objetivos del proyecto SIOSE-INNOVA y con un marco de trabajo actual aplicado tanto en el Laboratorio de Geomática de la Universidad de Alicante, como por otras empresas conocidas del sector (CARTO, Geographica, entre otros). En ? se presentan distintos flujos de trabajo basados en esta metodología, los cuales están muy relacionados con lo que se expone en este capítulo y son de uso frecuente en el Laboratorio de Geomática. En las próximas páginas se presenta un flujo de trabajo para el desarrollo de una extensión de base de datos. No obstante, utilizando la misma metodología se pueden crear y mantener páginas web, realizar simulaciones con modelos geográficos o hasta se pueden crear documentos imprimibles (como este mismo trabajo).

2.1. Integración continua y desarrollo colaborativo

Puntos de interés:

- El trabajo colaborativo se ha coordinado utilizando *Git* que es el sistema de **control de versiones** más popular de los últimos años (p.ej. utilizado en PostGIS, QGIS, CARTO y centenares de proyectos de ESRI, entre muchos otros).
- La **contenerización o dockers** es una novedosa tecnología para la virtualización de software/servicios, frente a la virtualización de sistemas operativos (p.ej. máquinas virtuales). La *orquestación de dockers* permite organizar complejos sistemas de información con muchas facilidades.
- PostgreSQL/PostGIS es la *geodatabase libre* más potente del mercado, destacando por sus opciones de **extensibilidad** (p.ej. PostGIS en sí misma es una extensión de PostgreSQL).
- Este trabajo ha sido redactado, editado y formateado utilizando las mismas plataformas de desarrollo utilizadas para desarrollar la extensión *pg_landmetrics* (*git* y *dockers*).

El trabajo colaborativo ha sido fundamental en el desarrollo de la extensión *pg_landmetrics* y por este motivo se dedica una parte importante de este trabajo a describir las herramientas colaborativas, bases de datos y técnicas que se ha aplicado para desarrollar y materializar los objetivos de este trabajo.

En esta sección se describen los procedimientos que se ha seguido periodicamente en este trabajo. La metodología de integración continua y el desarrollo colaborativo permiten colaborar con los miembros de un equipo de trabajo, lo cual asegura **un exhaustivo control de calidad y una mayor agilidad en la solución de problemas**.

En la subsección 2.1.1 se describe el uso que se ha hecho de *Git*, que es el sistema de control de versiones más popular de los últimos años. Siendo utilizado en un gran número de proyectos y empresas dedicadas a los SIG (p.ej. PostGIS¹, QGIS², CARTO³ y ESRI⁴). A continuación, en la subsección 2.1.2 se presentan los servicios de contenerización o *dockers*

¹<https://github.com/postgis/postgis>

²<https://github.com/qgis/QGIS>

³<https://github.com/CartoDB/cartodb>

⁴<https://github.com/Esri>

que se han utilizado para la virtualización del software/servicios necesarios para el desarrollo de la extensión, y la orquestación de éstos (despliegue sencillo de plataformas tecnológicas). La subsección 2.1.3 está dedicada a PostgreSQL, el Sistema Gestor de Bases de Datos (SGDB) *libre* más potente del mercado y a su extensión espacial PostGIS. Finalmente, en la subsección 2.1.4 se describen otras aplicaciones de escritorio que han sido utilizadas para analizar los resultados de esta metodología.

2.1.1. Control de versiones

El control de versiones es la gestión de los cambios que se realizan sobre un archivo o conjunto de archivos en un repositorio y que se utiliza para controlar las versiones del código de fuente, de modo que se puedan recuperar versiones anteriores en un momento específico. Un repositorio es donde se almacenan todos los datos actualizados y los registros históricos de los cambios realizados, principalmente en un servidor.

El control de versiones se aplica hoy en día en muchas plataformas de uso cotidiano (p.ej. Dropbox, Gdocs, etc). Sin embargo, en proyectos profesionales se trabaja de forma colaborativa siguiendo una serie de pasos que aseguran que un usuario **no pueda sobrescribir el trabajo de otro compañero sin su consentimiento**. Esta dinámica consiste en que cada usuario modifica ficheros desde su máquina local, los guarda en Internet y luego propone los cambios al resto del equipo. El sistema de control combina las modificaciones **si todos los usuarios están de acuerdo**. En este trabajo se ha utilizado un software que permite trabajar de este modo:

- **Git**⁵ es un sistema de control para el mantenimiento de versiones de código fuente de archivos. Git se ha convertido en la opción más popular de los últimos años y es utilizado por muchos proyectos como por ejemplo PostGIS, QGIS, CARTO y centenares de proyectos de ESRI, entre otros también muy conocidos. El control de versiones se realiza mediante una serie de verbos o instrucciones intuitivas como *clone* (clonar un repositorio), *pull* (traer cambios), *push* (enviar cambios), *add* (añadir nuevos ficheros) o *commit* (registrar cambios en local), entre otras opciones menos habituales.

⁵La documentación referente a Git está disponible a través de la página oficial del proyecto (<https://git-scm.com/>).

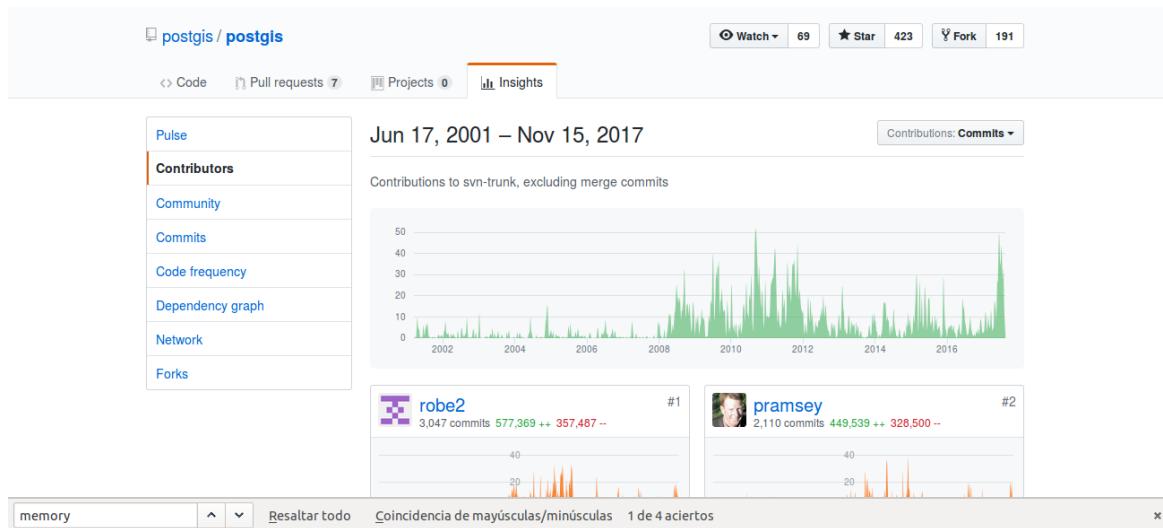


Figura 2.1 Estadísticas de la comunidad de desarrolladores de PostGIS en la plataforma GitHub. Paul Ramsey es un desarrollador muy activo también en foros de usuarios.

- **GitHub⁶** es una plataforma de desarrollo colaborativo que alberga proyectos y almacena el código fuente utilizando el sistema de control de versiones de *Git*. Esto quiere decir que muchas de las tareas de Git pueden realizarse desde un navegador de Web de un modo sencillo. Además, GitHub es completamente gratuito a condición de que el código sea público, lo cual no es un problema en proyectos de software libre. En otras circunstancias, también existen otras plataformas similares como Bitbucket⁷ o GitLab⁸ con las que es posible que los repositorios privados sean gratuitos.

La extensión *pg_landmetrics*, principal objetivo de este trabajo, ha sido desarrollada y alojada en un repositorio personal y público en GitHub⁹. Cuando se crean nuevas partes de la extensión, y funcionan correctamente, se proponen cambios al repositorio oficial del proyecto SIOSE-INNOVA¹⁰ y, si el resto de los colaboradores están de acuerdo, estos cambios pasan a formar parte de la versión oficial del proyecto.

El trabajo con Git es abierto y hay muchas maneras distintas de utilizar el software y colaborar con los otros usuarios. En el Laboratorio de Geomática de la Universidad de Alicante se trabaja como se detalla a continuación:

⁶La documentación de GitHub muestra distintos flujos de trabajo con Git así como documentación para distintas aplicaciones. Por ejemplo, es posible utilizar GitHub para crear una página web o una wiki (<https://github.com/>).

⁷<https://bitbucket.org>

⁸<https://about.gitlab.com/>

⁹https://github.com/andrearosado/pg_landmetrics

¹⁰https://github.com/siose-innova/pg_landmetrics

- El trabajo individual parte siempre de un ***fork del proyecto oficial***. Esto es un ***repositorio remoto*** (en GitHub) idéntico, pero disociado del proyecto original en GitHub. Así es posible experimentar libremente para después compartir cambios o incluso derivar hacia un proyecto independiente. En cualquier caso la autoría del código seguirá siempre vinculada a sus creadores. En este punto, el usuario puede hacer un nuevo *clone* local para trabajar desde un PC (***repositorio local***).
- **Actualización de ficheros de la extensión *pg_landmetrics* desde el repositorio local hacia el *fork* en GitHub** (ver la figura 2.2). El procedimiento consiste en ejecutar el comando *pull* desde la máquina local. Cuando el sistema de ficheros de la extensión no está actualizado, Git se encarga de obtener la última versión. Este paso es opcional, ya que si la versión local coincide con la remota, *git pull* no tiene ningún efecto. Una vez el usuario recibe un mensaje de notificación de que su sistema de fichero estaba actualizado con la última versión, se puede proceder a editar los archivos de *pg_landmetrics*. Cuando la edición termina (p. ej. por que se ha añadido una nueva métrica), el usuario puede guardar los cambios utilizando los comandos *git add* y *git commit*. Al guardar estos cambios, Git proporciona un mensaje con un *hash* (identificador único) por cada cambio que se haya realizado, o dicho de otro modo, se obtiene un identificador único para cada versión. Finalmente, desde la máquina local, el usuario sube esta última versión de la extensión a su repositorio del trabajo remoto (en GitHub), haciendo *git pull*, con lo que la versión más reciente de su trabajo estaría en local y en su repositorio de GitHub (no el oficial).
- **Actualización de la extensión desde el repositorio del trabajo local hacia el repositorio oficial del proyecto SIOSE-INNOVA** (ver la figura 2.3). Una de las primeras tareas de cualquier flujo de trabajo es realizar un *fork*, es decir, una copia de todos los ficheros del repositorio oficial (*upstream*) a un nuevo repositorio de este trabajo. La ventaja es que si en cualquier momento el repositorio *copiado* sufriera algún contratiempo, se podría volver a conseguir una copia del repositorio oficial. Lo mismo ocurre con la versión local obtenida a partir del comando *clone*. Éste último comando se ejecuta solamente cuando en la máquina local no existe una versión de *pg_landmetrics*. Esto implica que **siempre hay un mínimo de dos copias de respaldo con las que trabajar, además de todas las versiones anteriores**. A partir de este punto, se inicia el trabajo colaborativo entre usuarios y repositorios en GitHub. En este caso, se vuelve a repetir las mismas acciones de edición y actualización de archivos, como se ha visto en la Figura 2.2. Una vez subidos los cambios al *fork* de GitHub, es posible enviar una petición al repositorio oficial (*upstream*) para sincronizar ambos repositorios. Esta

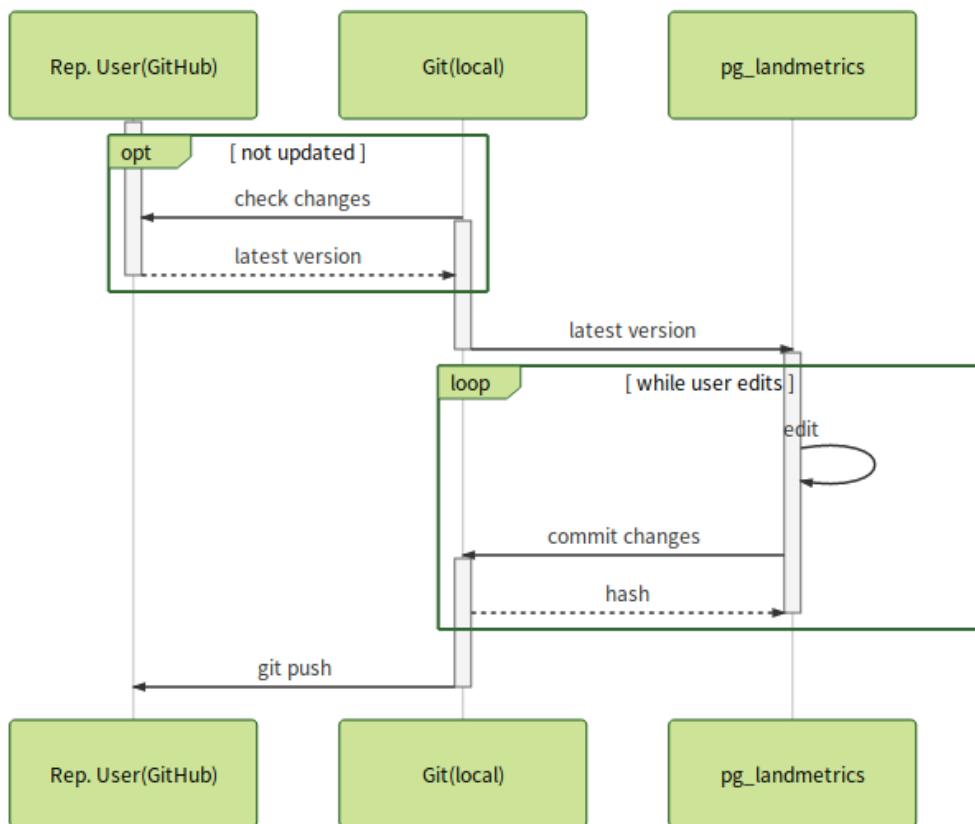


Figura 2.2 Diagrama de secuencia en el que se describe el flujo de integración continua más habitual en este trabajo (varias veces al día). Cada vez que se realizaban cambios importantes en la extensión *pg_landmetrics*, se añadían al control de versiones local y al de *La Nube de GitHub*.

operación se denomina *pull request* y permite compartir código entre repositorios alojados en la plataforma *GitHub*. Si dicha petición (*pull request*) es aceptada por el personal encargado del repositorio oficial (*upstream*), se realiza una operación de unión (*merge*) y se recibe una notificación de sincronización completada. En este punto, los repositorios en GitHub están casi sincronizados, salvo porque el repositorio oficial siempre cuenta con una nueva versión debida al *pull request*. En el último paso, el usuario que ha efectuado solicitado el *pull* descarga la última versión haciendo *pull upstream* y subiendo los cambios a su versión en GitHub (*fork*).

A pesar de lo complejo que parece este flujo de trabajo, se trata de **tareas sencillas y repetitivas en las que hay que comprobar que los cambios propuestos no son incompatibles con el trabajo anterior o de otros colaboradores.**

2.1.2. Contenerización y orquestación de servicios

La contenerización o *docker* es una novedosa tecnología que aporta portabilidad y que **virtualiza aplicaciones y servicios frente a la virtualización de sistemas operativos completos (p. ej. máquinas virtuales)**.

El trabajo con *Dockers* es similar al trabajo con Git. Existen una serie de instrucciones o verbos que permiten construir (*build*), desplegar (*up*) o detener (*down*) servicios que están empaquetados en contenedores (*dockers*). Los contenedores son como “cajas” que contienen todo lo necesario para ejecutar una determinada aplicación o servicio. Los contenedores pueden trabajar aisladamente o conectados. La técnica que permite que los contenedores se conecten de una manera más sencilla se llama *orquestación de contenedores* y es una técnica que permite organizar complejos sistemas de información con muchas facilidades. Existen distintas opciones de contenerización. Por ejemplo, la opción de Google se llama Kubernetes¹¹. Sin embargo en este trabajo se ha utilizado la plataforma más conocida que comprende varias herramientas de las que solamente presentamos las más importantes:

- **Docker**¹²: es un proyecto de código abierto que gestiona el despliegue de contenedores a través de *La Nube*. Docker permite separar a las aplicaciones (*software*) de la infraestructura (*hardware*), a los desarrolladores de las tareas de mantenimiento, para crear un modelo colaborativo e innovador.

¹¹<https://kubernetes.io/>

¹²<https://www.docker.com/>

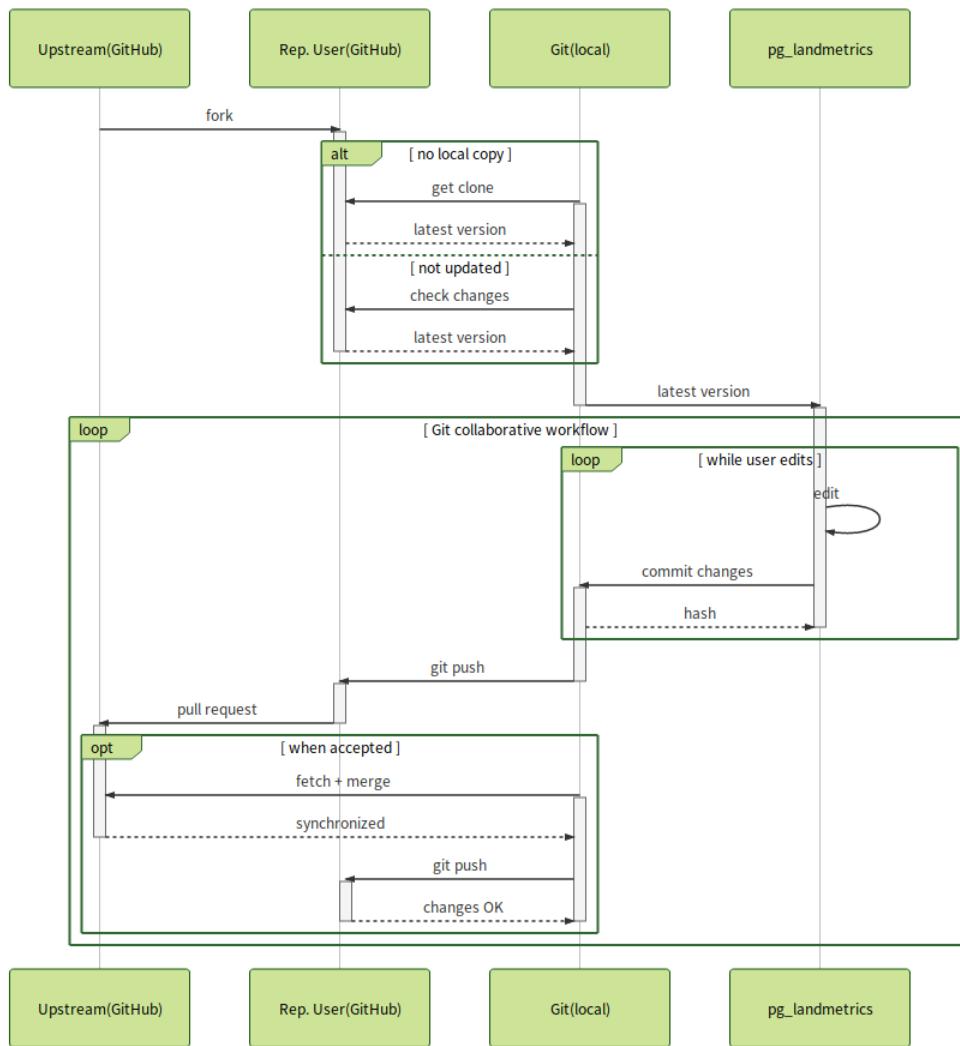


Figura 2.3 Diagrama de secuencia sobre el desarrollo colaborativo entre repositorios. Aproximadamente una vez al día, se proponen una serie de cambios al repositorio oficial (*pull request*), si estos cambios son aceptados después es posible sincronizar los dos repositorios.

- **Docker Hub**¹³: es un servicio en la nube que permite vincular repositorios y crear y almacenar sus imágenes. Es una plataforma que centraliza y distribuye las imágenes en contenedores, además de facilitar la automatización en el proceso de desarrollo.
- **Docker-compose**: utiliza un sencillo fichero de configuración que despliega todos los servicios necesarios de manera rápida y eficaz.

El uso de *dockers* permite aplicar una metodología de integración continua a la hora de desarrollar e implementar una extensión como *pg_landmetrics* (ver la figura 2.4). El usuario no necesita trabajar directamente con *dockers* sino que se puede automatizar en gran medida utilizando *docker-compose*. Gracias a un *software de orquestación* las pruebas sucesivas de la extensión se pueden realizar con un número mínimo de comandos: *docker-compose build*, *up* y *down*

docker-compose build se utiliza únicamente si la extensión ha sido modificada (p. ej. al añadir una nueva métrica). En caso contrario no sería necesario volver a construir todos los contenedores, solamente se utilizaría el comando *docker-compose up* para lanzar la última versión de la extensión. En el proyecto actual, la instrucción *docker-compose up* se ocupa de lanzar dos contenedores: PostgreSQL/PostGIS y PgAdmin4. Una vez han sido lanzados dichos contenedores y se han recibido las notificaciones de que la conexión ha tenido éxito, **el usuario ya dispone de la última versión del SGBD con la última versión de pg_landmetrics instalada y acceso a los datos de ejemplo, todo ello con una única instrucción**. Durante la *orquestación* el SGBS y PgAdmin4 quedan conectados y configurados para empezar a trabajar en nuevas métricas del paisaje, las nuevas funciones creadas quedan guardadas en el sistema y es posible operar con Git como se ha explicado en la subsección 2.1.1. Una vez modificada la extensión y guardados los cambios en el control de versiones, todo el software utilizado se *desinstala o se desmonta* con una única instrucción (*docker-compose down*).

2.1.3. Extensibilidad en PostgreSQL

PostgreSQL, y su extensión espacial PostGIS, fueron utilizados en las asignaturas del Máster a un nivel de usuario, lo cual ha resultado muy útil en la creación de funciones y métricas. Sin embargo, en este trabajo hay que destacar una de las características más notables de este SGBD: la extensibilidad.

¹³<https://hub.docker.com/>

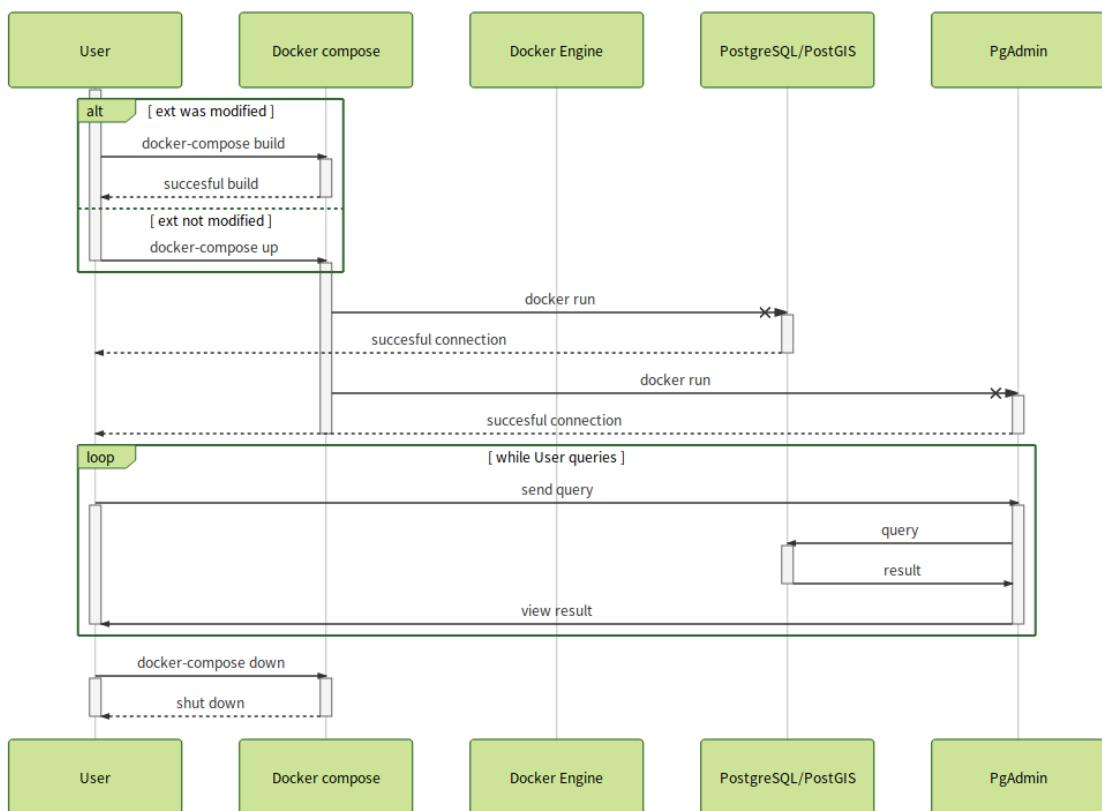


Figura 2.4 Diagrama de secuencia. Integración continua en la implementación y desarrollo de funciones SQL con *dockers*. Cada vez que una nueva función es añadida a la base de datos, se elimina cualquier rastro de la versión anterior y se vuelve a lanzar un nuevo contenedor para evaluar si los cambios son válidos.

La *extensibilidad* es la adaptabilidad de un *software* frente a la introducción de nuevos cambios y funcionalidades. En este sentido PostgreSQL es un SGBD altamente extensible, ya que **toda su funcionalidad se añade como si de un catálogo se tratase**. El sistema de tipos, funciones, índices, operadores y funciones agregadas, son solo algunas de las características que pueden ser extendidas en PostgreSQL. Este diseño extensible ha servido para crear el propio PostGIS, que es una *extensión*. Una extensión de Postgres sería un conjunto de nuevas funcionalidades que son añadidas de manera conjunta. Además, de las características ya mencionadas, los propios lenguajes procedurales serían extensiones en cierto modo. Lo más interesante es que estas extensiones se pueden programar en cualquiera de los lenguajes disponibles en PostgreSQL, incluido SQL.

Físicamente, una extensión de PostgreSQL consiste en varios ficheros que automatizan la instalación en una base de datos de nuevos tipos de datos, funciones, operadores (espaciales o no), funciones agregadas, entre otros objetos¹⁴. **La mayor parte de aportaciones de este trabajo al proyecto SIOSE-INNOVA encajan en este punto, creando funciones y funciones agregadas, (en su mayor parte en lenguaje SQL), para ser incorporadas en la extensión pg_landmetrics.**

La plataforma de contenerización descrita en la subsección 2.1.2 realiza muchas tareas de instalación y configuración. Entre estas tareas, la más significativa consiste en la instalación automática de la última versión de la extensión a partir de un fichero *makefile* en el que se organiza todo el código de la extensión¹⁵. Junto con el desarrollo de código SQL, la actualización de este fichero es esencial para que la extensión se instale correctamente.

2.1.4. Otras aplicaciones

La mayor parte de este trabajo se ha realizado utilizando Git, Dockers y PostgreSQL, que funcionan como servicios y no necesitan de un interfaz gráfico para operar. Dado el carácter abierto y multiplataforma de estas aplicaciones, el Sistema Operativo tampoco es una limitación (Linux GNU, Windows o Mac). Sin embargo, hay tareas en las cuales contar con un interfaz gráfico bien organizado puede resultar de gran ayuda.

¹⁴La creación de extensiones en PostgreSQL es relativamente sencilla pero tiene muchas opciones (ver <https://www.postgresql.org/docs/10/static/extend-extensions.html>)

¹⁵GNU Make es un software muy conocido para automatizar la compilación de nuevos programas de un modo determinado. PostgreSQL hace un uso particular de esta tecnología para facilitar la instalación de extensiones pero para saber más es posible consultar la documentación oficial en <https://www.gnu.org/software/make/manual/make.html>

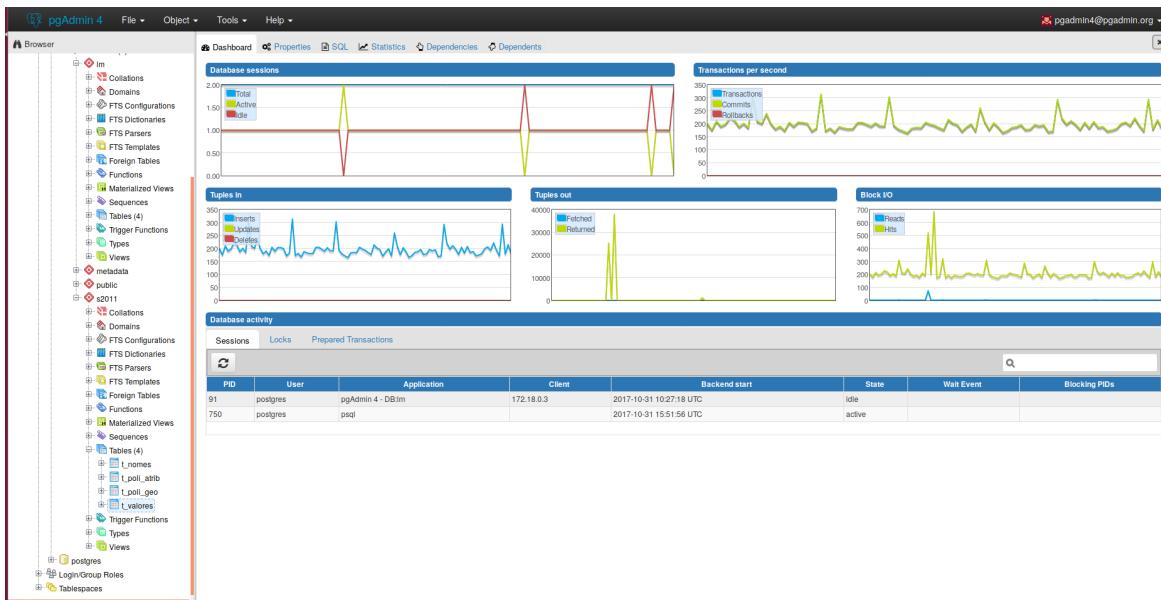


Figura 2.5 Vista de la interfaz web de PgAdmin4.

La preparación de datos de ejemplo y la visualización de las consultas realizadas con PostgreSQL se han realizado con las siguientes aplicaciones:

- **PgAdmin4¹⁶**: es una plataforma que administra, gestiona y desarrolla código abierto en bases de datos PostgreSQL. En la Figura 2.5 se muestra una captura de pantalla de PgAdmin4 mientras se realiza una carga masiva de datos procedentes del SIOSE. Cabe decir que en el Máster se trabajo con la versión anterior de este programa, la funcionalidad es muy similar aunque cambie el aspecto, pero el despliegue con Dockers hace que la instalación de este programa en cualquier ordenador sea más sencilla y que pueda funcionar también desde un servidor de Internet.
- **QGIS 2.18¹⁷**: es una aplicación de escritorio SIG, de código abierto, que analiza, maneja y opera con datos como *raster/vector* y bases de datos. Además, facilita la conexión entre bases de datos espaciales como PostGIS. Este programa ya se conocía ampliamente, aunque en este trabajo se ha trabajado fundamentalmente con el gestor de bases de datos y la visualización de geometrías provenientes de PostgreSQL/PostGIS.

¹⁶<https://www.pgadmin.org/>

¹⁷<https://www.qgis.org/es/site/>

2.2. Conjuntos de datos

Puntos de interés:

En este trabajo se han utilizado los siguientes conjuntos de datos:

- **Un capa con un paisaje de ejemplo** para comprobar que la extensión *pg_landmetrics* da resultados correctos para las métricas.
- **La base de datos SIOSE-2011** para realizar una prueba de capacidad de consulta sobre un gran volumen de datos.
- Varios **grids a distintas escalas** (1:50.000, 1:100.000, 1:200.000 y 1:500.000) para simular una serie de consultas por parte de un usuario (ver ejemplo de la Figura 1).

En esta sección se describen los dos conjuntos de datos utilizados para poner a prueba la extensión *pg_landmetrics*: un paisaje de ejemplo y el SIOSE de 2011.

El **paisaje de ejemplo** se ha utilizado para comprobar si las métricas de paisaje funcionan correctamente, comparando los resultados ofrecidos desde PgAdmin4, como también desde QGIS (DB-Manager). La escala de digitalización para esta capa es 1:50.000 y se ha creado en dos sistemas geodésicos de referencia: European Terrestrial Reference System 1989 (ETRS89) y World Geodetic System 84 (WGS84), ya que las métricas no se calculan igual si se trabaja con geometrías proyectadas (*geometry*) o si se utiliza un sistema de coordenadas esféricas (*geography*).

En este pequeño conjunto de datos hay un total de 51 polígonos que se han digitalizado utilizando las herramientas de edición avanzada y geoprocreso de QGIS. Se han definido 8 categorías de cobertura de suelo donde a cada polígono le corresponde un tipo de categoría representada por un color (ver figura 2.6). Se han definido los colores a partir de los 147 que recoge la Scalable Vector Graphics (SVG) Specification¹⁸ además de tener en cuenta la clasificación de color que especifica el *Corine Land Cover (CLC)*.

El segundo conjunto de datos que se ha utilizado es la base de datos **SIOSE 2011** facilitada por el Servicio de Ocupación del Suelo del Instituto Geográfico Nacional (IGN), gracias a su participación en el proyecto SIOSE-INNOVA. Cabe decir que esta base de datos contiene más información y es más compleja que la versión descargable desde el Centro de Descargas del Centro Nacional de Información Geográfica.

¹⁸<http://www.december.com/html/spec/colorshtml.html>

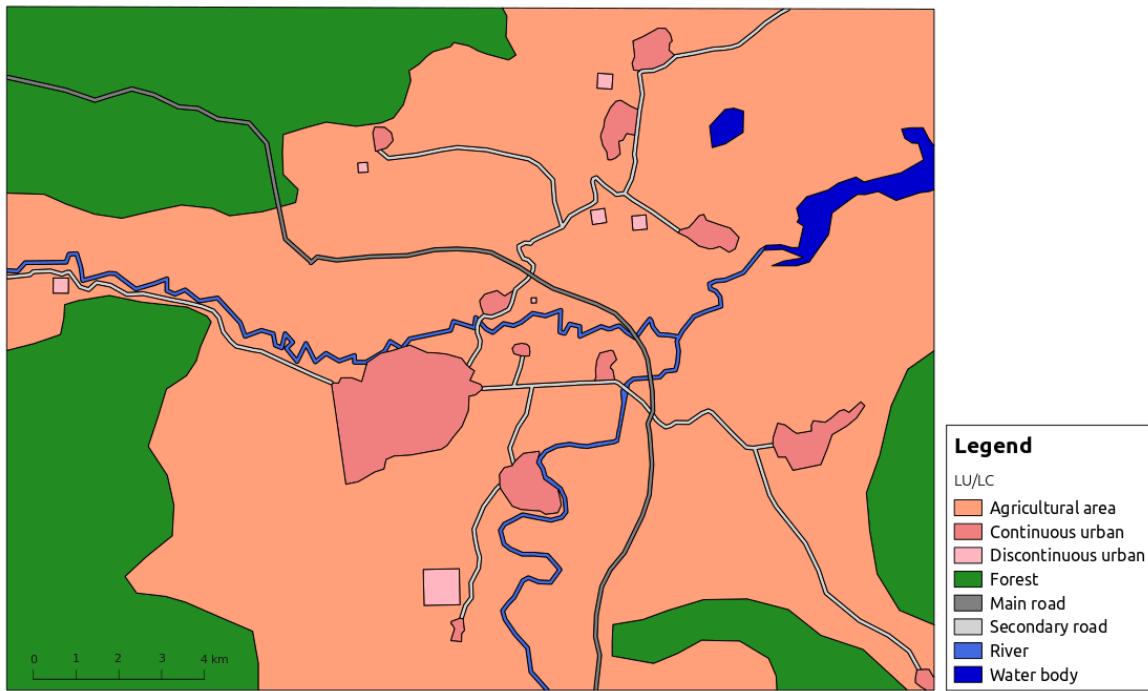


Figura 2.6 Coberturas del suelo del paisaje de ejemplo. Estas geometrías se distribuyen junto con la extensión *pg_landmetrics* para que los usuarios realicen pruebas antes de importar sus propios datos.

Esta base de datos tan voluminosa se compone de 4 tablas. En la tabla de datos t_nomes se encuentran todas las nomenclaturas de cada polígono, en t_poli_atrib todos los atributos de los polígonos como por ejemplo el código multietiqueta de clasificación del SIOSE o de clasificación de las distintas nomenclaturas (iberpix, perfil ambiental, toponomía, *Corine Land Cover*, etc.), en cuanto a la tabla t_poli_geo ofrece la geometría de cada uno de los 2.562.800 de polígonos existentes y, finalmente, en la tabla t_valores aparecen los valores de coberturas y usos del suelo codificados y asociados a los polígonos, así como también cálculos de valores en Hectáreas.

Como se menciona en la sección 1.1, los polígonos del SIOSE pueden describirse por una *cobertura simple* o una *cobertura compuesta*. Dentro de esta última se encuentran las coberturas no predefinidas, es decir, que un mismo polígono puede contener varias coberturas de suelo. A raíz de esta característica, se ha realizado una reclasificación de los polígonos a partir de la cobertura prevalente que presenta la jerarquización de la clasificación de los usos y coberturas del suelo del SIOSE. Con ello se ha obtenido una nueva tabla con 85 coberturas prevalentes asignadas a cada uno de los polígonos que se encuentran en la tabla t_poli_geo. Para que las consultas sean más rápidas entre las distintas tablas se han generado índices para indexar los datos de cada una de las tablas. Esta reclasificación es necesaria ya que trabajar con clasificaciones *multietiqueta* de usos y coberturas del suelo es un problema más amplio que no se pretende abordar en este trabajo.

Una vez se han reclasificado todos los polígonos del SIOSE-2011, se han obtenido *grids* (cuadrículas cartográficas) para el cálculo de métricas del paisaje a distintas escalas de referencia, a través del Centro de Descargas del Instituto Geográfico Nacional (CNIG)¹⁹.

A partir del grid 1:500.000 se han escogido dos celdas de las zonas de estudio en las cuales se han calculado algunas métricas de paisaje. Estas dos zonas contienen la mayor parte de las provincias de Alicante y Zaragoza, así como los alrededores de cada una²⁰. A continuación se han seleccionado y extraído todas aquellas celdas que comprenden las dos celdas de las áreas de estudio con escala de referencia 1:25.000, 1:50.000 y 1:100.000. A partir de todas estas celdas ha simulado una serie de consultas como las que podrían darse en un visor web. La Figura 1 ayuda a visualizar en qué consiste la simulación o experimento que se analiza en el capítulo 3.

Volumen de los datos (TABLA). La base de datos del SIOSE 2011 proporcionada por el equipo nacional del SIOSE era muy voluminosa ()

¹⁹<http://centrodedescargas.cnig.es/CentroDescargas/index.jsp>

²⁰Se han escogido estas dos zonas es por hacer referencia al lugar donde se han realizado las prácticas de empresa y dónde se ha cursado el *máster*.

Tabla 2.1 Características de los conjuntos de datos utilizados.

Tipo	Tablas	Filas	Tamaño total
SIOSE-2011	t_nomes	36.790.972	6116 MB
	t_poli_atrib	2.562.800	451 MB
	t_poli_geo	2.562.800	3981 MB
	t_valores	10.932.639	1041 MB
Grids	grid_25k	756	232,3 kB
	grid_50k	192	57,8 kB
	grid_100k	48	13,8 kB
	grid_500k	2	677bytes
Sample	sample_25830	51	122,6 kB
	sample_4326	51	122,5 kB

2.3. Selección de métricas

Puntos de interés:

- El número potencial de métricas del paisaje es **indeterminado** y depende de muchos factores (p.ej. objetivos del estudio, modelos de datos como *raster/vector* o en red, niveles de agregación y/o escala, etc).
- Resulta esencial determinar unas **métricas representativas** para esta primera propuesta.

Como se indica en la sección 1.2, existen cientos de métricas de paisaje que no siempre tienen significado para todas las aplicaciones de estudio, ya que su interés depende de muchos factores (p. ej. los objetivos del estudio, el modelo de datos, la escala, nivel de agregación etc). Por este mismo motivo y según los objetivos de este trabajo, se han seleccionado unas métricas del paisaje representativas y que aparecen frecuentemente en la bibliografía consultada.

En la tabla 2.2 se listan las métricas acompañadas por su abreviatura. Se ha escogido un número de métricas repartidas por igual en tres niveles de agregación: a nivel de polígono (*patch*), nivel de categoría (*class*) y nivel de paisaje (*landscape*). Además, se han escogido métricas que calculan operaciones simples, como por ejemplo el área o el perímetro, y por otro lado, métricas cuyos cálculos son más complejos, como por ejemplo la distancia al

Tabla 2.2 Métricas de paisaje disponibles en la extensión.

Nivel	Métrica	Abreviatura
Patch	Patch Area	AREA
	Patch Perimeter	PERIM
	Perimeter-Area-Ratio	PARA
	Shape Index	SHAPE
	Core Area	CORE
	Number of Core Areas	NCORE
	Core Area Index	CAI
	Euclidean Nearest Neighbour Distance	ENN
Class	Total (Class) Area	CA
	Percentage of Landscape	PLAND
	Total Edge	TE
	Edge Density	ED
	Total Core Area	TCA
	Core Area Percentage of Landscape	CPLAND
	Number of Patches	NP
	Patch Density	PD
Landscape	Total Area	TA
	Total Edge	TE
	Edge Density	ED
	Number of Patches	NP
	Patch Density	PD
	Patch Richness	PR
	Patch Richness Density	PRD
	Shannon's Diversity Index	SHDI
	Simpson's Diversity Index	SHIDI

vecino más próximo o la densidad. El objetivo es encontrar la mayor variedad posible de contratiempos en el desarrollo de nuevas métricas.

2.4. Desarrollo de funciones en PostgreSQL

Puntos de interés:

- Los desarrollos en PostgreSQL se pueden realizar en lenguajes de programación como ANSI C, SQL y/o distintos lenguajes procedurales (p.ej. PL/pgSQL, PL/R, PL/Python, entre muchos otros), **dependiendo de las necesidades**.
- El lenguaje SQL es el más sencillo para programar en PostgreSQL y está mejor integrado.

La implementación y desarrollo en PostgreSQL se puede realizar desde distintos lenguajes de programación como por ejemplo ANSI C que es un estándar del lenguaje de programación C. Es el más popular en el desarrollo de softwares y aplicaciones, y además su código es portable entre distintas plataformas. Otro de los lenguajes de programación más conocido y utilizado hasta la fecha es **SQL (*Structured Query Language*)**. Es un lenguaje de consulta estructurada en gestión de bases de datos relacionales que maneja álgebra y cálculo relacional. Este tipo de lenguaje es declarativo ya que ofrece la posibilidad de realizar consultas para obtener un resultado, además de estar orientados a resolver problemas. Finalmente, PostgreSQL da la posibilidad de utilizar distintos lenguajes procedurales para realizar tareas que en SQL resulten demasiado complicadas. El lenguaje procedural más habitual es PL/pgSQL, pero también hay otros como PL/R, PL/Python, o PL/Bash que permiten programar en la base de datos utilizando una gran variedad de recursos de análisis.

Dado que se ha trabajado sobre una base de datos relacional y orientada a objetos como es la del SIOSE y se ha utilizado tanto PgAdmin4 como la extensión de PostgreSQL/PostGIS, se ha empleado como lenguaje de programación SQL en la mayoría de los casos para simplificar o compactar lo máximo posible todas las funciones de la extensión y que éstas no dependan de otras. SQL es un lenguaje estandarizado y resultaría viable migrar estas consultas a otros SGBD como Oracle Spatial o SQLite. Además, el motor de consultas de PostgreSQL solamente puede ser aprovechado al máximo al ejecutar SQL, siendo las funciones de otros lenguajes procedurales como “cajas negras” que impiden optimizar más las consultas. Sin embargo, es cierto que en algunos casos se ha tenido que emplear PL/pgSQL cuando no era posible elaborar las funciones en SQL.

Una vez decidido qué lenguaje de programación utilizar para implementar y desarrollar las funciones, se han construido consultas SQL a partir de las métricas seleccionadas (ver Tabla

2.2). Las fórmulas de estas métricas han sido extraídas de la documentación del software FRAGSTATS, que es referencia para otros programas mencionados en la introducción. Los resultados de cada consulta SQL se validaron manualmente sobre el conjunto de datos de ejemplo, utilizando QGIS para calcular las métricas por pasos.

Cuando estas últimas comprobaciones han resultado satisfactorias, se ha procedido a desarrollar la función de PostgreSQL por cada una de las métricas. Cabe mencionar que todas las funciones se han desarrollado por duplicado, para calcular las métricas proyectadas (*geometry*) o en coordenadas esféricas (*geography*). Así pues, la extensión permite trabajar con conjuntos de datos con prácticamente cualquier sistema geodésico de referencia.

Dentro de las opciones que permite PostgreSQL hay métricas que se han podido implementar como **funciones simples**, pero en otros casos ha sido necesario crear **funciones agregadas** más complejas.

Funciones simples

Las funciones simples corresponden a las métricas de paisaje a nivel de polígono. Como ejemplo de ello, la métrica de paisaje *Patch Area* (AREA) devuelve la suma del área del polígono dividido por 10.000 para obtener un valor en unidades de hectárea. Como se puede observar en el ejemplo de código (2.4), a partir de la fórmula matemática que calcula esta métrica, se ha construido su consulta SQL, la cual resulta sencilla de leer.

Ejemplo de código 2.1 Consultas SQL para la métrica AREA.

```
1 SELECT St_Area(geom)/10000 FROM sample_patches_25830;  
2 SELECT St_Area(geom)/10000 FROM sample_patches_4326;
```

Una vez escrita una consulta SQL (ejemplo 2.4), se puede desarrollar una función de un modo bastante estructurado. Como se observa en el ejemplo de código 2.4, la función se define a partir del nombre de la propia función, el tipo de valor que devuelve, la consulta SQL que se ha construido anteriormente y finalmente, un comentario en el que explica lo que la función tiene que devolver como resultado. Como se ha mencionado anteriormente, la función se ha elaborado tanto para *geometry* como para *geography*, ya que PostGIS lo permite.

Ejemplo de código 2.2 Función simple para calcular la métrica AREA por polígonos.

```
1 CREATE OR REPLACE FUNCTION p_area(geom geometry)
```

```

2 RETURNS metric AS
3 $$
4
5 SELECT (1, St_Area(geom)/10000)::metric;
6
7 $$
8 LANGUAGE SQL
9 IMMUTABLE
10 RETURNS NULL ON NULL INPUT;
11
12 COMMENT ON FUNCTION p_area(geom geometry) IS 'Divide el área en
      metros cuadrados de un polígono por 10.000 para devolver un
      valor en Hectáreas.';
```

Una vez se ha creado la función, se valida el resultado devuelto a través de un ejemplo de uso. Por ejemplo, en el ejemplo de código 2.4) se calcula la métrica AREA a para todo el conjunto de datos de ejemplo.

Ejemplo de código 2.3 Ejemplo de uso para calcular la métrica AREA.

```

1 SELECT (p_area(geom)).value As p_area FROM sample_patches_25830;
2 SELECT (p_area(geom)).value As p_area FROM sample_patches_4326;
```

Funciones de agregación

No todas las métricas se pueden calcular como funciones simples de PostgreSQL. Las funciones simples aplican un cálculo individualmente a filas o polígonos de la base de datos. Cuando una métrica implica un sumatorio y/o operaciones entre varias filas o polígonos (diversidad, vecindad, etc), entonces es necesario utilizar funciones de agregación. Normalmente, todas las métricas a nivel de categoría o a nivel de paisaje serán funciones de agregación. Por ejemplo, las funciones SUM() o COUNT() de PostgreSQL, son algunas de las más utilizadas, pero **como se ha dicho en la subsección 2.1.3 es posible extender PostgreSQL/PostGIS con funciones de agregación propias.**

A modo de ejemplo, la métrica *Total Core Area (TCA)* devuelve la suma de los núcleos de las áreas (m²) de cada polígono, dividido por 10.000 (para obtener Hectáreas). Al igual que en el ejemplo de código 2.4, se ha preparado una nueva consulta basada en la función

matemática utilizada en FRAGSTATS. En el ejemplo de código (2.4) se aprecian que en este caso es necesario especificar una *profundidad de borde* (un buffer negativo) para calcular la métrica.

Ejemplo de código 2.4 Consulta SQL para la métrica TCA.

```

1 SELECT SUM(St_Area(St_Buffer(geom, -50)))/10000 FROM
      sample_patches_25830;
2 SELECT SUM(St_Area(St_Buffer(geom, -50)))/10000 FROM
      sample_patches_4326;
```

A partir de ésta consulta (ver ejemplo 2.4), se ha desarrollado la función de agregado. La diferencia que tiene esta función respecto a la de *Patch Area* es que se necesita crear un agregado dentro de la función ya que el resultado que se quería obtener era agrupado según el tipo de categoría. Como se observa en el ejemplo de código 2.4, la función se estructura por el nombre de la propia función, el tipo de valor que devuelve que en este caso será etiquetado el resultado según el tipo cobertura de suelo al que corresponde, la consulta SQL que se ha construido anteriormente desglosada de distinta forma que la anterior ya que se quería obtener el sumatorio del resultado de los polígonos agrupados por categoría, un agregado que forma la estructura del resultado con el valor y la etiqueta de cobertura, y finalmente, un comentario en el que explique qué tiene que devolver como resultado. Como se ha mencionado, la función se ha elaborado tanto para *geometry* como para *geography*.

Ejemplo de código 2.5 Función de agregado para calcular TCA por categorías.

```

1 CREATE OR REPLACE FUNCTION c_totalcorearea_state(
2     current_state metric_labeled,
3     geom geometry,
4     category text)
5 RETURNS metric_labeled
6 LANGUAGE 'sql',
7
8 AS
9 $BODY$
10
11 WITH inputs AS (
12     SELECT current_state AS cstate
13 ), melt AS (
14     SELECT unnest((cstate).pairs) AS m2 FROM inputs
```

```

15      UNION
16          SELECT (category, (p_corearea(geom)).value):::
17              metric_labeled_pair AS m2
18      ), summarize AS (
19          SELECT (m2).label, SUM((m2).value) AS value FROM melt
20              GROUP BY (m2).label
21      )
22      SELECT (13, ARRAY(SELECT (label, value)::metric_labeled_pair
23          FROM summarize))::metric_labeled;
24
25 $BODY$;
26
27
28 CREATE AGGREGATE c_totalcorearea(geometry, text)(
29     SFUNC=c_totalcorearea_state,
30     STYPE=metric_labeled,
31     INITCOND='(13,{} )'
32 );
33
34 COMMENT ON AGGREGATE c_totalcorearea(geom geometry, category
35     text) IS 'Suma las áreas de los núcleos de cada polígono de
36     la misma categoría dividido por 10.000 para devolver un valor
37     en Hectáreas.';
```

Una vez se ha desarrollado la función de agregado, se ha ejecutado un ejemplo de uso (ver ejemplo de código 2.4) para obtener el cálculo de la métrica TCA a partir de un conjunto de datos (ver en la figura 2.7).

Ejemplo de código 2.6 Ejemplo de uso para calcular la métrica TCA.

```

1  SELECT c_totalcorearea(geom,category).value AS c_totalcorearea
2      FROM sample_patches_25830;
3  SELECT c_totalcorearea(geom,category).value AS c_totalcorearea
4      FROM sample_patches_4326;
```

En cuanto a las métricas a nivel de paisaje también se han desarrollado de tipo agregado para obtener **un solo resultado para el conjunto de todo el paisaje**. A partir de este punto, se han elaborado todas las funciones de las métricas que se han seleccionado en la sección 2.3.



Métrica de paisaje: Patch Area, Core Area y Number of Core Areas

Nivel de agregación: Patch y Class

Categoría: Matorral

Profundidad buffer = -23 metros

AREA = 2'92977759817602 Ha.

NCORE = 2

TCA = 0'84769189771803 Ha.



Métrica de paisaje: Patch Area, Core Area y Number of Core Areas

Nivel de agregación: Patch y Class

Categoría: Afloramientos Rocosos y Roquedos

Profundidad buffer = -23 metros

AREA = 2.11612635285956 Ha.

NCORE = 1

TCA = 0'509579068880577 Ha.

Figura 2.7 Ejemplo de cálculo de AREA, NCORE y TCA a partir de la geometría del SIOSE 2011.

2.5. Experimento sobre la capacidad de cálculo de *pg_landmetrics*

Finalmente, con todas las funciones implementadas, desarrolladas y testeadas sobre el conjunto de datos de ejemplo, resulta sencillo realizar pruebas de rendimiento de la extensión *pg_landmetrics* sobre el conjunto de datos del SIOSE-2011.

Este experimento busca simular la consulta que un usuario podría realizar en un visor web del SIOSE que calculase métricas del paisaje de un modo ágil e interactivo (ver figura 1). Cada vez que el usuario hace zoom o se desplaza por el visor Web SIG, se envian consultas sobre las métricas del paisaje para la superficie del mapa visible en ese momento (normalmente será una superficie rectangular).

El movimiento del usuario se simula a partir de una secuencia de consultas utilizando como referencia los grids descritos en la sección 2.2. En este sentido se ha colaborado en la elaboración de una última función (ajena a la extensión) para automatizar la realización de numerosas consultas sobre el SIOSE-2011 (ver ejemplo de código 2.5). Los resultados de este experimento se explican en el capítulo 3.

Ejemplo de código 2.7 Extracción de los resultados de las métricas en diferentes escalas de referencia a partir de grids.

```

1 CREATE OR REPLACE FUNCTION test.runtests()
2   RETURNS TABLE (
3     grid_tablename regclass,
4     zone_id int,
5     cell_id int,
6     total_landscape_area numeric,
7     avg_patch_area numeric,
8     total_class_area numeric)
9 LANGUAGE 'plpgsql',
10 AS $$
11 DECLARE
12   script text;
13
14 BEGIN
15   SELECT DISTINCT string_agg('SELECT * from test.runtests(' || 
16     quote_literal(table_schema || '.' || table_name) || ')', ', '
17     UNION ALL ') into script

```

```
16   FROM information_schema.tables WHERE table_schema = 'test' AND
17     table_name like 'grid_%';
18
19   RETURN QUERY EXECUTE script;
20
21 END
21 $$;
```

Esta función construye una tabla virtual con los resultados organizados por columnas según la zona de estudio a la que corresponden (Zaragoza o Alicante), el número de celda sobre el que está intersectado, y los valores de cada métrica en su columna correcta. A continuación se ha ejecutado un *script*, proporcionado por el personal del Laboratorio de Geomática, el cual extrae todos los valores calculados, para todas las escalas de referencia y en una sola tabla en formato CSV.

2.6. Documentación de la extensión

Puntos de interés:

- Una parte fundamental de esta metodología es la **documentación** del desarrollo y uso de la extensión.
- Una buena documentación con ejemplos facilitará el cálculo de métricas del paisaje en grandes repositorios, **sobretodo para aquellos usuarios con menos experiencia** en PostgreSQL/PostGIS.

La documentación del desarrollo y uso de la extensión es una de las partes más importante de la metodología de este trabajo. Una buena documentación facilitará el despliegue y aprovechamiento de la extensión a otros usuarios, sobretodo a aquellos menos expertos en la materia. La documentación ha consistido en introducir de manera sistemática comentarios en el código de la extensión (documentación para desarrolladores) y en crear documentos de apoyo (documentación para usuarios).

La documentación de proyectos también se presta a aplicar una metodología de integración continua y a trabajar en plataformas colaborativas. Ha sido necesario aprender a manejar distintos **lenguajes de marcado** (tipo HTML) para los distintos tipos de documentos:

- **Markdown para documentación Web²¹**. Markdown es un lenguaje ligero que permite una escritura sencilla y de fácil lectura usando texto plano. Se ha utilizado para documentar el uso de la extensión en GitHub.
- **TeX para documentación en PDF²²**. TeX es el lenguaje que se utiliza en el sistema de textos LaTeX y que crea documentos con una alta calidad tipográfica. Desde hace tiempo este lenguaje se emplea por un gran número de usuarios para escribir artículos o libros científicos. Para trabajar con este lenguaje, se ha utilizado la aplicación Texmaker. Este trabajo ha sido creado en formato PDF mediante el uso de estos programas.
- **Scalable Vector Graphics (SVG) para la generación de gráficos vectoriales a partir de las geometrías de PostGIS²³**. SVG es un lenguaje capaz de crear gráficos basados en vectores escalables de alta calidad de resolución. A partir de este lenguaje se ha desarrollado una función capaz de representar gráficos vectoriales a partir de las geometrías de cualquier geodatabase (ver en el anexo C).
- **Mermaid para la creación de diagramas.²⁴** Mermaid es un lenguaje que genera gráficos a partir de texto mediante JavaScript. Se han generado desde diagramas de flujo, diagramas de secuencia y de Gantt. Algunos de los cuales se encuentran en la página Web del GitHub.

²¹<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

²²<https://www.latex-project.org/>

²³https://www.w3schools.com/graphics/svg_intro.asp

²⁴<https://mermaidjs.github.io/>

3. RESULTADOS Y DISCUSIÓN

Puntos de interés:

- El principal resultado de este trabajo es la extensión pg_landmetrics, en la cual se ha colaborado en un importante porcentaje de funcionalidades (*commits*).
- Las métricas implementadas en forma de funciones se han validado sistemáticamente para asegurar que devuelven el resultado correcto.
- Se ha desarrollado un caso de uso completo basado en la aplicación propuesta en la introducción (ver Figura 1) con el que se han obtenido resultados prometedores (*usabilidad* y volumen).

En este capítulo se presentan los resultados de este trabajo siguiendo una secuencia lógica y utilizando figuras que ayuden a **visualizar cómo pg_landmetrics se integra en una aplicación como la descrita en la figura 1**.

Los resultados de este trabajo son prometedores y apuntan a que sí que se podrían publicar servicios de consulta, cómo el cálculo de las métricas del paisaje, a partir de la base de datos del SIOSE u otras de un volumen o complejidad similares.

En este capítulo también se destacan los aspectos más novedosos y relevantes de este trabajo, así como las implicaciones de carácter práctico.

En la subsección 3.1 se describen las aportaciones realizadas al proyecto SIOSE-INNOVA y que corresponden al grueso del presente trabajo. A continuación, en la subsección 3.2 se presentan los resultados de un caso de uso o experiencia computacional en el que se ha puesto a prueba la extensión desarrollada. Entre los objetivos de este trabajo era importante permitir que los usuarios del SIOSE calculasen métricas de una manera sencilla e intuitiva, pero también que pudiesen manejar un gran volumen de datos, cosa que en otras aplicaciones muy conocidas no es posible (ver capítulo [Introducción](#)).

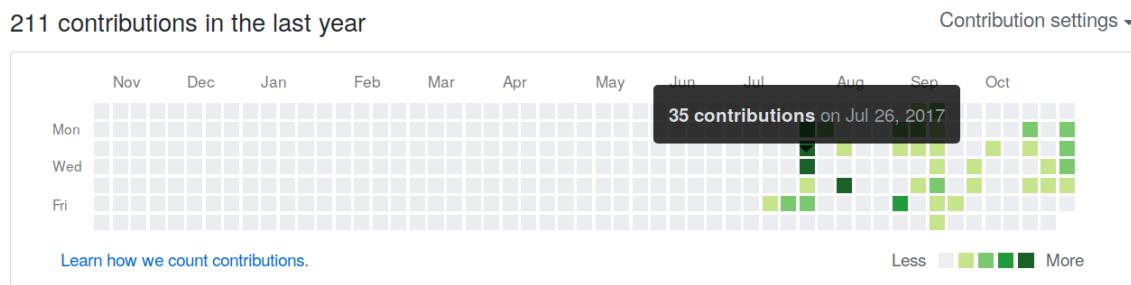


Figura 3.1 Calendario de actividad y contribuciones de este trabajo al proyecto *pg_landmetrics* (*heatmap calendar*).

3.1. pg_landmetrics

Una parte importante de este trabajo ha consistido en aprender a colaborar en un equipo de desarrolladores de geodatabases. En la metodología de integración continua aplicada en el Laboratorio de Geomática de la Universidad de Alicante resulta relativamente sencillo hacer un repaso del trabajo realizado en cada fase del proyecto.

El uso de un software de control de versiones como Git y la gestión del proyecto en la plataforma GitHub, hacen posible seguir y analizar las aportaciones que los miembros del proyecto SIOSE-INNOVA han hecho en la extensión *pg_landmetrics*.

La plataforma GitHub ofrece numerosas estadísticas del trabajo de cada usuario y también de la vitalidad de cada proyecto. Por ejemplo, en la figura 3.1 se puede ver un diagrama de tipo *calendar heatmap* extraido de GitHub, en el que se aprecia el número total de aportaciones al proyecto (*commits*).

Siguiendo con la figura 3.1, los colores más intensos indican días de una mayor actividad, siendo el día 26 de julio el día en el que más aportaciones se realizó. Un numero tan grande de aportaciones es normal en la puesta en marcha de un proyecto de este tipo y en particular fue el día en el que se incorporaban sentencias SQL que habían sido testeadas en días anteriores. Según la complejidad de las métricas o las funciones a desarrollar, hay días con una menor actividad y, en aquellos casos más problemáticos, también hay días en los que los desarrollos los hacían otros miembros del equipo y había tiempo de preparar conjuntos de datos de ejemplo, preparar documentación o trabajar en la redacción del presente trabajo.

En total se ha contribuido al proyecto con 211 aportaciones, que posteriormente han sido aceptadas en la mayoría de los casos. El proyecto oficial únicamente muestra 209

contribuciones, pero ambas cifras no están directamente relacionadas. Hay que tener en cuenta que las acciones de *pull request* explicadas en la figura 2.3 de la metodología suelen empaquetar varias actualizaciones o *commits* que en el repositorio oficial aparecen como una única aportación. GitHub permite analizar el trabajo y la contribución de cada miembro del equipo de un modo muy minucioso.

El control de versiones sirve para recuperar un punto de trabajo anterior o también para supervisar cómo se ha ido desarrollando un trabajo. Aproximadamente, a partir de la figura 3.1 y otras similares es posible reconstruir las fases principales en que se desarrolló este trabajo. En julio se hizo una revisión bibliográfica y se aprendió a utilizar las herramientas vistas en la metodología. A finales de julio se seleccionaron y se crearon consultas SQL para calcular la mayoría de las métricas. En agosto se creó un conjunto de datos de ejemplo y se validaron los resultados de cada consulta. En septiembre se convirtieron las consultas en funciones simples y, dado que no todas las métricas se podían implementar del mismo modo, en octubre se modificaron varias funciones para convertirlas en funciones de agregación. Finalmente, en noviembre se realizó el experimento computacional cuyos resultados se muestra en la sección 3.2.

En el diagrama de red de GitHub (ver figura 3.2) se puede visualizar el trabajo colaborativo desarrollado desde el principio hasta la realización de la experiencia computacional presentada en este trabajo. Se trata de un diagrama dinámico en el cual se puede consultar el contenido de cada contribución, ya sea la creación de nuevas funciones o cambios. En la figura 3.2 se pueden ver varios *commits* por parte de distintos colaboradores, una operación de *pull upstream*, cuatro operaciones *push* y una serie de contribuciones propias que aún no habían sido aceptadas por el equipo del proyecto SIOSE-INNOVA. Los colores de las líneas ayudan a interpretar mejor este tipo de diagramas:

- La línea negra muestra los *commits* integrados en el repositorio oficial.
- Las líneas verdes y azules se refieren a aportaciones de este trabajo que finalmente fueron aceptadas por el equipo del SIOSE-INNOVA.
- La línea morada hace referencia a las últimas funciones creadas, pendientes de aceptación al término de este trabajo.

En cuanto a los resultados concretos de este trabajo, se han desarrollado varias funciones sobre PostgreSQL/PostGIS que calculan métricas del paisaje de distintos tipos. Todas las funciones se han creado por duplicado (*funciones sobrecargadas*) para trabajar con sistemas

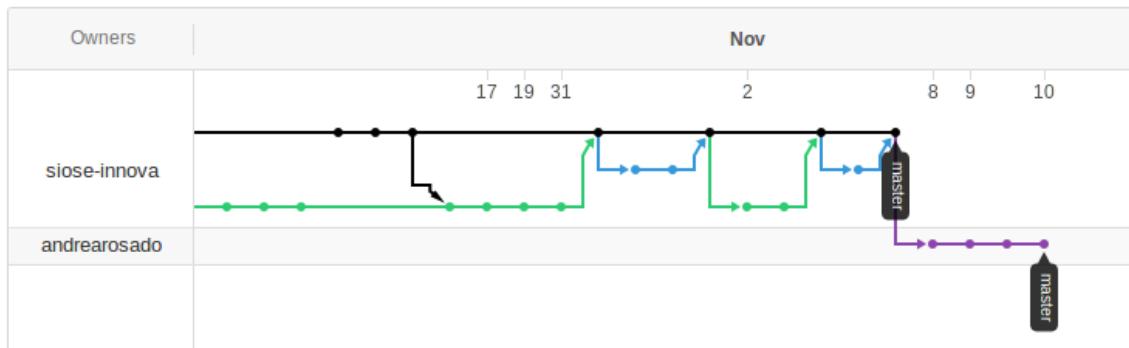


Figura 3.2 Diagrama de red de GitHub.

de coordenadas planas y con coordenadas geográficas. Esta decisión tiene que ver con que en bases de datos tan voluminosas como la del SIOSE, en ocasiones será necesario realizar cálculos con polígonos que se encuentren muy distantes, llegando incluso a representarse en usos o sistemas de referencia espacial distintos. El uso de coordenadas esféricas permite trabajar con una geodatabase sin preocuparse de este tipo de problemas, pero a cambio de poder utilizar un número menor de funciones de PostGIS. No obstante, resulta que todas las métricas seleccionadas a partir de la documentación de FRAGSTATS han podido ser calculadas partiendo del tipo *geography* de PostGIS. Esto hace que las operaciones con el SIOSE resulten más sencillas si no es necesario dividir la base de datos por usos o comunidades autónomas.

En la tabla 3.1 se listan todas las métricas del paisaje implementadas en la extensión. Los colores indican si la métrica se ha podido desarrollar como una función SQL simple o si por el contrario ha sido necesario crear una función de agregación (combinación de varias funciones).

En esta tabla de resumen también se puede ver que prácticamente todas las consultas se podían calcular manualmente (independientemente de la dificultad) salvo la distancia euclídea al vecino más próximo (ENN; Euclidean Nearest Neighbour Distance). Esta métrica implica numerosos cálculos de distancia entre polígonos, el uso de índices espaciales y la aplicación de una serie de filtros. Evidentemente, el cálculo del ENN resultaría muy trabajoso si no se utiliza ningún lenguaje de programación. La consulta SQL para el ENN se realizó de un modo más o menos sencillo. Sin embargo, su implementación como función SQL de agregación está aún siendo revisada por el equipo del SIOSE-INNOVA.

Las métricas marcadas como pendientes de aceptación en la tabla 3.1 devuelven el resultado correcto con el paisaje de ejemplo descrito en este trabajo, por lo que **es de esperar que pronto pasen a formar parte del repositorio oficial de *pg_landmetrics***.

En relación con los objetivos de este trabajo y del proyecto SIOSE-INNOVA, resulta muy significativo el modo en que se han simplificado todas las tareas de desarrollo y cálculo de las métricas del paisaje. Los miembros del equipo de trabajo pueden disponer, en cuestión de minutos, de una versión actualizada de la extensión *pg_landmetrics* con los últimos cambios realizados por otro compañero. Esta versión actualizada se obtiene mediante la ejecución de un sencillo comando (*docker-compose up*) y viene acompañada de todo el software necesario para trabajar, incluyendo las mismas opciones de configuración utilizadas por todo el equipo. Igualmente, resulta también sencillo trabajar en laboratorio desde un servidor o desde un portatil en casa. Más aún, la tecnología utilizada funciona en los sistemas operativos más utilizados (Linux GNU, Windows o Mac).

Los usuarios finales de *pg_landmetrics* o de la base de datos del SIOSE, también verán facilitado su trabajo, ya que las 25 métricas implementadas simplifican en gran medida las consultas necesarias para realizar este tipo de análisis. Por ejemplo, una métrica en apariencia tan sencilla como Total Core Area (TCA) pasa a calcularse en una única línea frente a las más de 20 líneas que serían necesarias en SQL (ver ejemplos de código 2.5 y 2.7).

3.2. Caso de uso sobre el SIOSE-2011

La puesta a prueba de la extensión *pg_landmetrics* ha permitido obtener una experiencia computacional al adquirir resultados de un caso de uso a partir del conjunto de datos del SIOSE 2011, descrito en el capítulo 2. Además, se ha mostrado que en una única consulta SQL ha servido para obtener todos los resultados necesarios. Es cierto que en un principio se desarrolló otra consulta que tenía 55 líneas frente a la consulta final que tiene 21 líneas como se muestra en el ejemplo de código 2.5.

Al aplicar este experimento a partir de la consulta SQL, se han extraído los resultados de las métricas *Patch Area (AREA)*, *Total (Class) Area (CA)* y *Total Area (TCA)* en diferentes escalas de referencia. Exponer la tabla de los valores obtenidos puede tener gran complejidad a la hora de su interpretación. Por ello se han representado en una serie de figuras/mapas para mostrarlos de forma más sencilla y visual.

Tabla 3.1 Listado de las métricas de paisaje disponibles en la extensión, según su nivel de complejidad y si han sido aceptadas en el repositorio oficial.

Métrica	Manual/QGIS	Consulta SQL	<i>pg_landmetrics</i>
AREA	●	●	●
PERIM	●	●	●
PARA	●	●	●
SHAPE	●	●	●
CORE	●	●	●
NCORE	●	●	●
CAI	●	●	●
ENN	○	●	○
CA	●	●	●
PLAND	●	●	●
TE	●	●	○
ED	●	●	○
TCA	●	●	○
CPLAND	●	●	○
NP	●	●	○
PD	●	●	○
TA	●	●	●
TE	●	●	●
ED	●	●	●
NP	●	●	●
PD	●	●	●
PR	●	●	○
PRD	●	●	○
SHDI	●	●	○
SHIDI	●	●	○

Función simple

Función agrég.

- Aceptada

- Pendiente

Los resultados están representados mediante el método de clasificación de desviación estándar que muestra los valores medios en rangos iguales; aquellos valores que se encuentren alejados de la media están representados de color naranja (aquellas celdas que presentan un área menor) y morado (aquellas celdas con un área mayor).

En la figura 3.3, calculada la métrica Patch Area (AREA), se puede realizar una comparación entre las distintas zonas de estudio. Este ejemplo presenta una semejanza entre los distintos paisajes. Sin embargo, en la parte derecha superior de la zona de Zaragoza se presenta una anomalía. Si se observa con más detalle, la celda de color morado más oscuro presenta unos valores más altos por encima de la media. Esto se debe a que la intersección de la celda con los polígonos haga que se calculen todos los polígonos que se encuentren dentro más aquellos polígonos donde alguna de sus partes se encuentre dentro de la respectiva celda. Por este motivo, esta celda presenta un alto valor ya que dentro de esta celda se encuentran 6 polígonos con una gran extensión (ver figura 3.4).

Este sería uno de los ejemplos para el cálculo de AREA a nivel de polígono, donde aquellas anomalías que distingan de un paisaje a otro sea por casos como el que una celda le pertenezcan muchos polígonos cuyos valores de sus áreas son altos. Este tipo de fenómenos son los que caracterizan y diferencian unos paisajes de otros.

Por otro lado, se ha querido mostrar el resultado del cálculo del área a nivel de categoría. Para ello se han seleccionado algunas que pertenezcan a la clase de forestal como: prados (290), pastizal (300), frondosas caducifolias (312), frondosas perennifolias (313), coníferas (316) y matorral (320). Cabe mencionar que la elaboración de esta clasificación de ocupación del suelo queda explicada en el capítulo 2. En la figura 3.5, calculada la métrica Total (Class) Area (CA), a raíz de la selección de las categorías anteriores se puede apreciar aproximadamente la disposición del relieve donde en las zonas más interiores se encuentran los polígonos con mayor área.

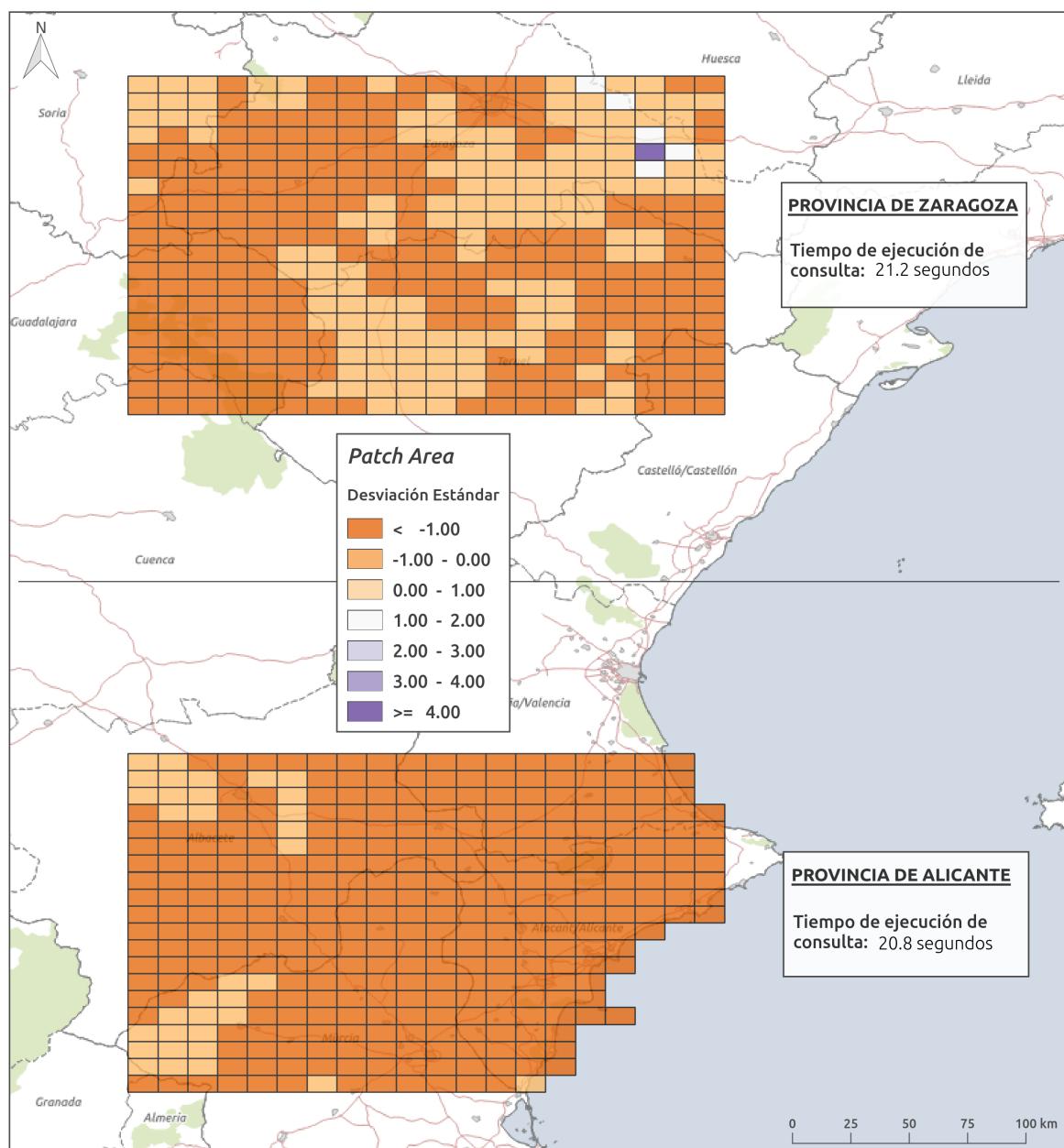


Figura 3.3 Calendario de actividad y contribuciones de este trabajo al proyecto *pg_landmetrics* (*heatmap calendar*).

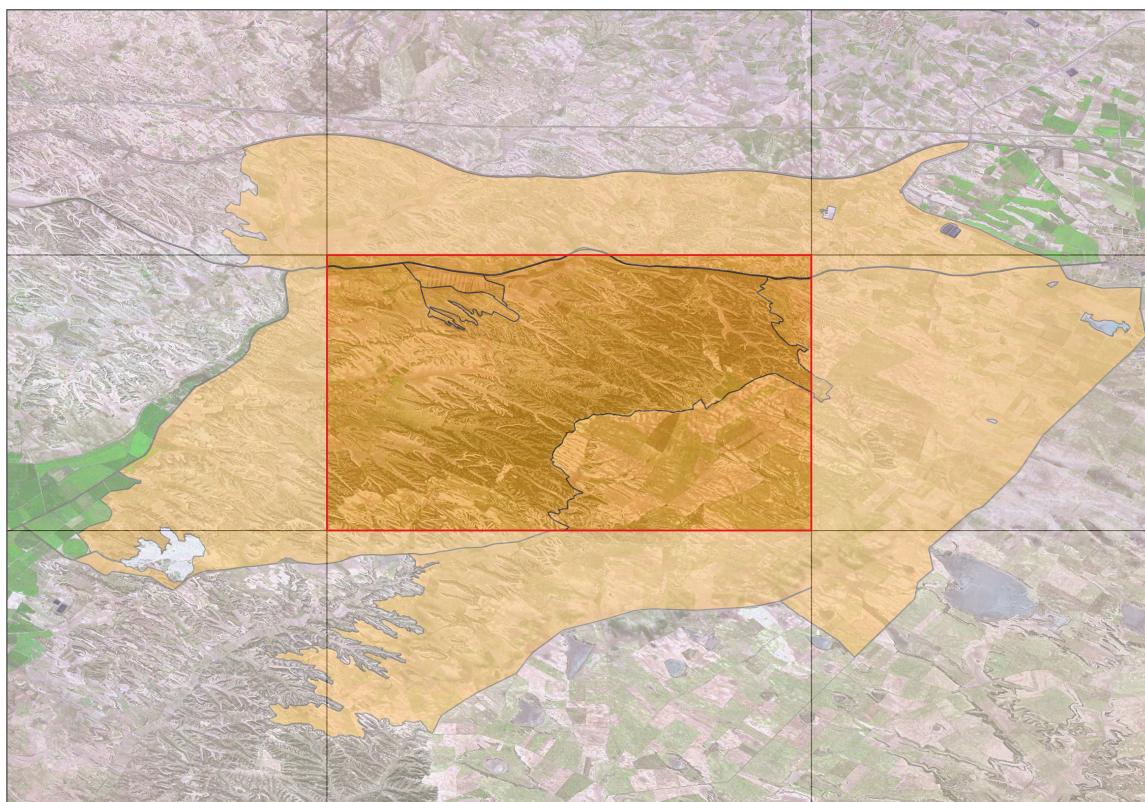


Figura 3.4 Calendario de actividad y contribuciones de este trabajo al proyecto *pg_landmetrics* (*heatmap calendar*).

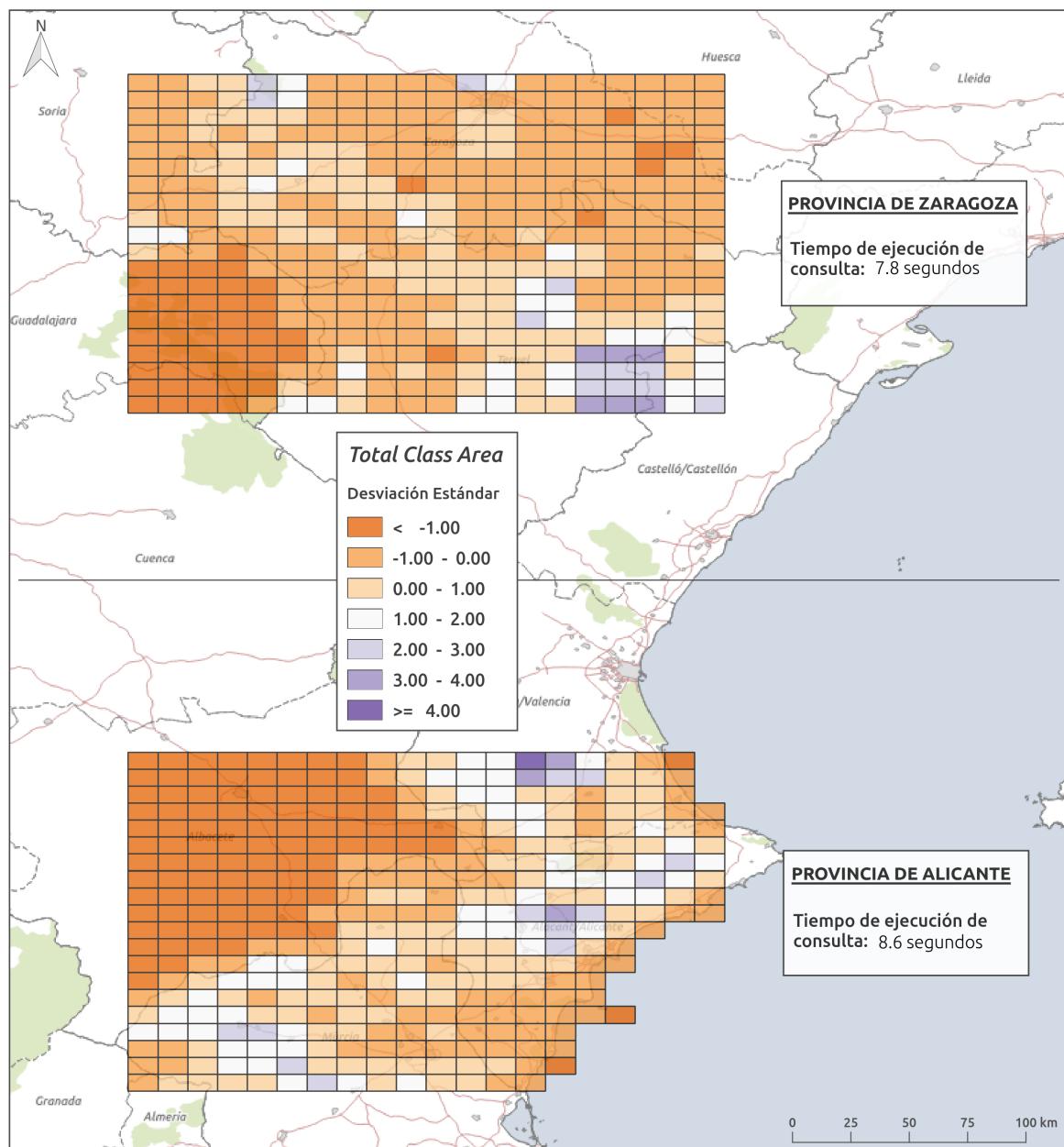


Figura 3.5 Calendario de actividad y contribuciones de este trabajo al proyecto *pg_landmetrics* (*heatmap calendar*).

4. CONCLUSIONES Y TRABAJO FUTURO

Puntos de interés:

- Se ha desarrollado una extensión que simplifica complejas consultas SQL a consultas de una o pocas líneas.
- El rendimiento de las consultas resulta prometedor, lo que permitirá crear servicios web de consulta directa sobre el SIOSE y bases de datos similares.
- El trabajo con *dockers* facilita la reproducibilidad de la investigación y un despliegue escalable en Internet.
- Los conocimientos adquiridos en el Máster han servido como introducción a un campo profesional muy complejo.

Los objetivos planteados en la [Introducción](#) comprendían aspectos de trabajo colaborativo, cuestiones tecnológicas y había una gran preocupación por mejorar la usabilidad de bases de datos voluminosas y complejas como la del SIOSE. En este sentido, el objetivo principal se ha conseguido al contribuir significativamente en el desarrollo de una extensión de Postgres/PostGIS realmente potente, utilizando las tecnologías y plataformas más actuales.

El desarrollo de nuevas métricas se puede sistematizar en gran medida, lo que permite repartir el trabajo en equipos multidisciplinares, como lo es el del Laboratorio de Geomática de la Universidad de Alicante.

Este laboratorio trabaja con las mismas herramientas que utilizan algunas empresas de referencia (CARTO, Geographica, etc)

En este trabajo se ha conseguido desarrollar un prototipo de una aplicación que, según los objetivos del Proyecto SIOSE-INNOVA, se debería desarrollar en unos tres años. El desarrollo de una extensión *en producción* llevará más tiempo, es un trabajo complejo que requiere de todo un equipo de expertos y meses de trabajo. **El trabajo en equipo es esencial en este tipo de proyectos.**

Tras redactar este trabajo es posible valorar aún más los contenidos del "Máster en Tecnologías de la Información Geográfica para la Ordenación del Territorio: SIG y Teledetección". En las asignaturas de *fundamentos teóricos sobre bases de datos, diseño e implementación de bases de datos, lenguajes de programación, software libre, análisis espacial básico, análisis visual de imágenes y desarrollo e implementación de la información geográfica en aplicaciones infográficas* se adquirieron conocimientos básicos para empezar a trabajar en un proyecto sobre geodatabases como este. Esta experiencia será similar en otro tipo de proyectos sobre teledetección, cartografía, etc, con lo cual el aprendizaje será constante en cualquier rama de las TIG.

A. FUNCIONES TIPO

<——Mover texto de los anexos a resultados——>

En el anexo A se presentan los ejemplos más básicos de funciones para calcular los distintos tipos de métricas. Se pueden distinguir métricas que requieren un único registro (*funciones SQL*), frente a aquellas funciones que necesitan considerar más de un registro (*funciones agregadas*).

La programación puede variar en complejidad. Por ejemplo, hay funciones agregadas que aplican cálculos posteriores a la agregación de valores o funciones que realizan más de un cálculo simultáneo. Además, las métricas no siempre se calculan igual si son proyectadas (*geometry*) o en coordenadas esféricas (*geography*).

Los ejemplos más sencillos de los distintos tipos de métricas serían las que se calculan únicamente a partir de las áreas de uno o más polígonos: *Patch Area*, *Class Area* y *Landscape Area*. La implementación del resto de métricas puede consultarse en el repositorio del proyecto en GitHub.

Ejemplo de código A.1 Función para calcular AREA de un polígono de PostGIS de tipo *geography*

```
1 CREATE OR REPLACE FUNCTION p_area(geom geography)
2 RETURNS metric AS
3 $$
4
5 SELECT (1, St_Area(geom)/10000)::metric;
6
7 $$
8 LANGUAGE SQL
9 IMMUTABLE
10 RETURNS NULL ON NULL INPUT;
```

```
11
12 COMMENT ON FUNCTION p_area(geom geography) IS 'Divide el área en
      metros cuadrados de un polígono por 10.000 para devolver un
      valor en Hectáreas.';
```

Ejemplo de código A.2 Función para calcular CA por categorías.

```

1  /*
2   Total (Class) Area - devuelve la suma de las áreas (m2) de todos
      los polígonos correspondientes al tipo de polígono, dividido
      por 10.000 (unidades: Hectáreas).
3 */
4 -- SAMPLE USAGE
5 /*
6 SELECT c_totalarea_state(ARRAY[('Agricultural area', ('Total
      Class Area'::text, 10, 'Ha.'::text))::labeled_metric], geom,
      category) FROM sample_patches_25830;
7 SELECT c_totalarea_state(ARRAY[('Agricultural area', ('Total
      Class Area'::text, 10, 'Ha.'::text))::labeled_metric], geom,
      category) FROM sample_patches_4326;
8 */
9
10 CREATE OR REPLACE FUNCTION c_totalarea_state(
11     current_state metric_labeled,
12     geom geometry,
13     category text)
14 RETURNS metric_labeled
15 LANGUAGE 'sql'
16
17 AS
18 $BODY$
19
20 WITH inputs AS (
21     SELECT current_state AS cstate
22 ), melt AS (
23     SELECT unnest((cstate).pairs) AS m2 FROM inputs
24     UNION
25     SELECT (category, (p_area(geom)).value)::(
26         metric_labeled_pair AS m2
27 ), summarize AS (
28     SELECT (m2).label, SUM((m2).value) AS value FROM melt
          GROUP BY (m2).label
)

```

```
29  SELECT (9, ARRAY(SELECT (label, value)::metric_labeled_pair FROM
30    summarize))::metric_labeled;
31
32
33
34 -- SAMPLE USAGE
35 -- SELECT c_totalarea(geom,category) FROM sample_patches_25830;
36
37 CREATE AGGREGATE c_totalarea(geometry, text)(
38   SFUNC=c_totalarea_state,
39   STYPE=metric_labeled,
40   INITCOND='(9,{} )'
41 );
42
43 COMMENT ON AGGREGATE c_totalarea(geom geometry, category text)
44   IS 'Calcula la suma de las áreas de los polígonos de la misma
45     categoría dividido por 10.000 para devolver un valor en
46     Hectáreas.';
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
```

```
63 ), summarize AS (
64     SELECT (m2).label, SUM((m2).value) AS value FROM melt
65     GROUP BY (m2).label
66 )
67 SELECT (9, ARRAY(SELECT (label, value)::metric_labeled_pair FROM
68     summarize))::metric_labeled;
69
70
71 -- SAMPLE USAGE
72 -- SELECT c_totalarea(geom,category) FROM sample_patches_4326;
73
74 CREATE AGGREGATE c_totalarea(geography, text)(
75     SFUNC=c_totalarea_state,
76     STYPE=metric_labeled,
77     INITCOND='(9,{}',
78 );
79
80 COMMENT ON AGGREGATE c_totalarea(geom geography, category text)
81 IS 'Calcula la suma de las áreas de los polígonos de la misma
82 categoría dividido por 10.000 para devolver un valor en
83 Hectáreas.';
```

Ejemplo de código A.3 Función para calcular TA del paisaje.

```
1  /*
2   Total Area - devuelve el total del área (m2) del paisaje
3   dividido por 10.000 (unidades: Hectáreas).
4 */
5 -- SAMPLE USAGE:
6 /*
7 SELECT (l_totalarea(geom)).value FROM sample_patches_25830;
8 SELECT (l_totalarea(geom)).value FROM sample_patches_4326;
9 */
10 CREATE OR REPLACE FUNCTION l_totalarea_state(metric,geometry)
11     RETURNS metric AS
12 $$
13     SELECT $1 + (p_area($2)).value;
14 $$
15 LANGUAGE 'sql' IMMUTABLE;
16
17
18 CREATE AGGREGATE l_totalarea(geometry)(
19     SFUNC=l_totalarea_state,
20     STYPE=metric,
21     INITCOND='(0,0)'
22 );
23
24 COMMENT ON AGGREGATE l_totalarea(geometry) IS 'Calcula el área
25     total del paisaje dividida por 10.000 para devolver un valor
26     en Hectáreas.';
27
28 CREATE OR REPLACE FUNCTION l_totalarea_state(metric,geography)
29     RETURNS metric AS
30 $$
31     SELECT $1 + (p_area($2)).value;
32 $$
33 LANGUAGE 'sql' IMMUTABLE;
34
```

```
35 CREATE AGGREGATE l_totalarea(geography) (
36     SFUNC=l_totalarea_state ,
37     STYPE=metric ,
38     INITCOND='(0,0)'
39 );
40
41 COMMENT ON AGGREGATE l_totalarea(geography) IS 'Calcula el área
      total del paisaje dividida por 10.000 para devolver un valor
      en Hectáreas.';
```


B. RESULTADOS FINALES

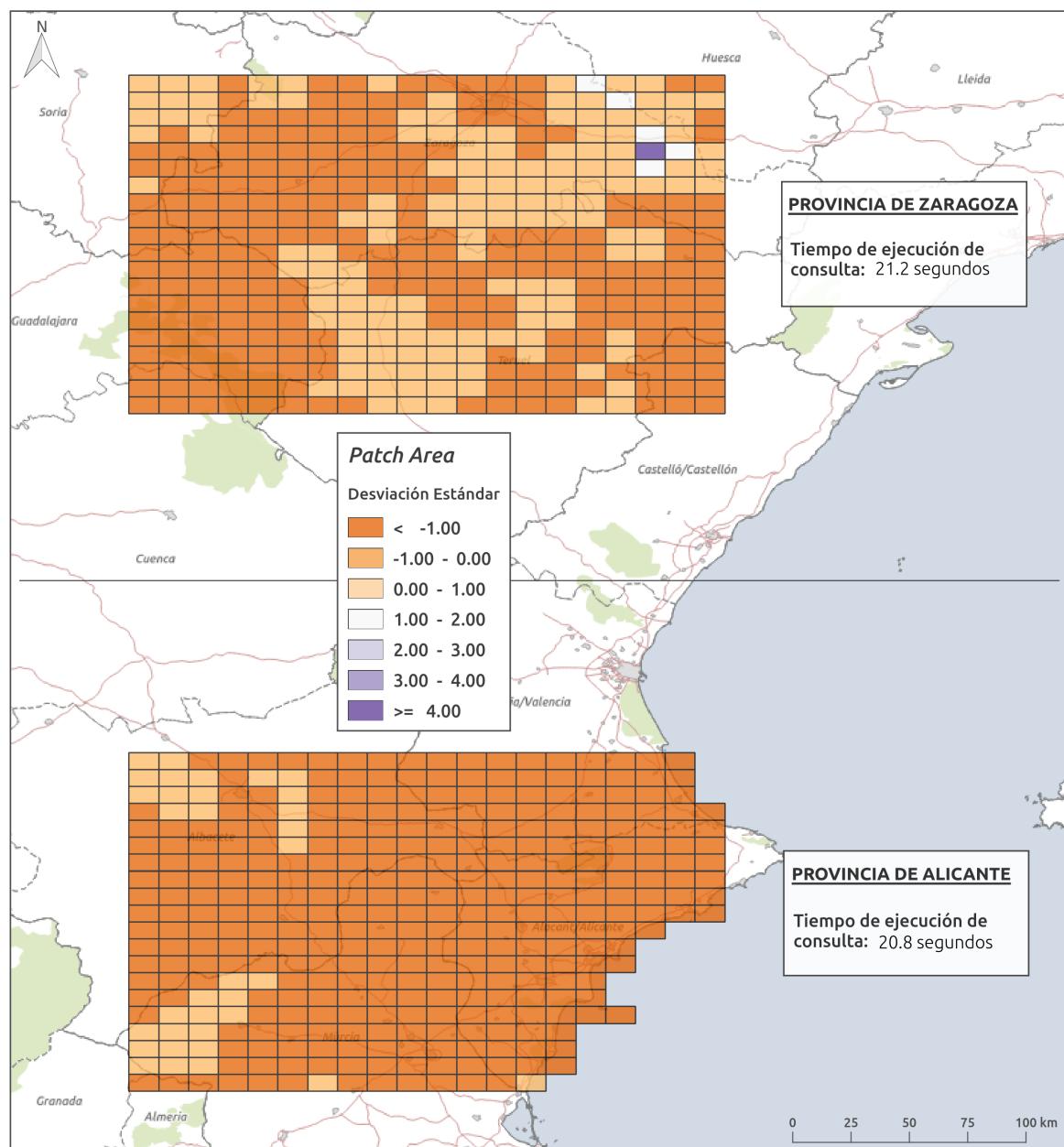


Figura B.1 Título.

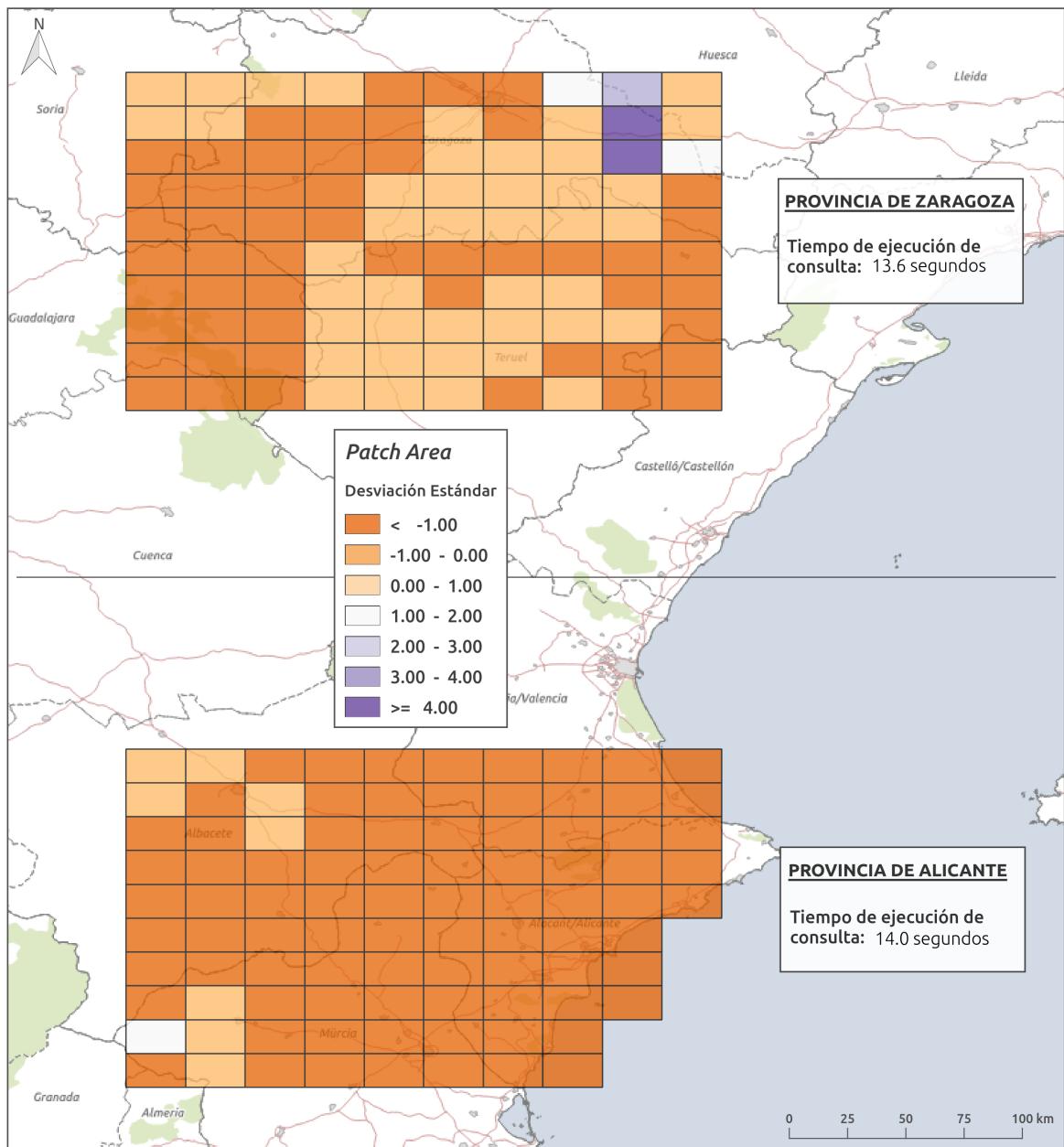


Figura B.2 Título.

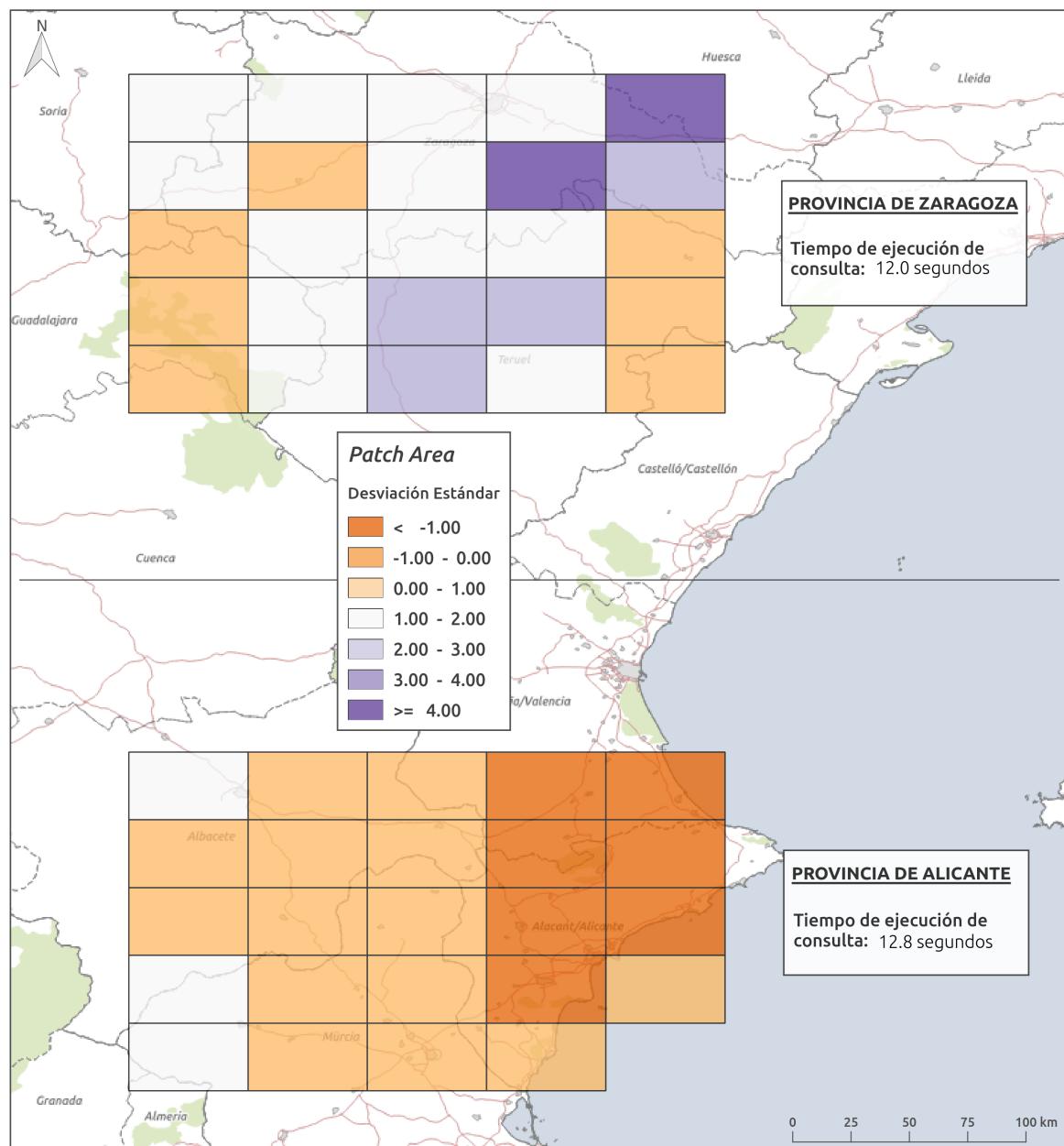


Figura B.3 Título.

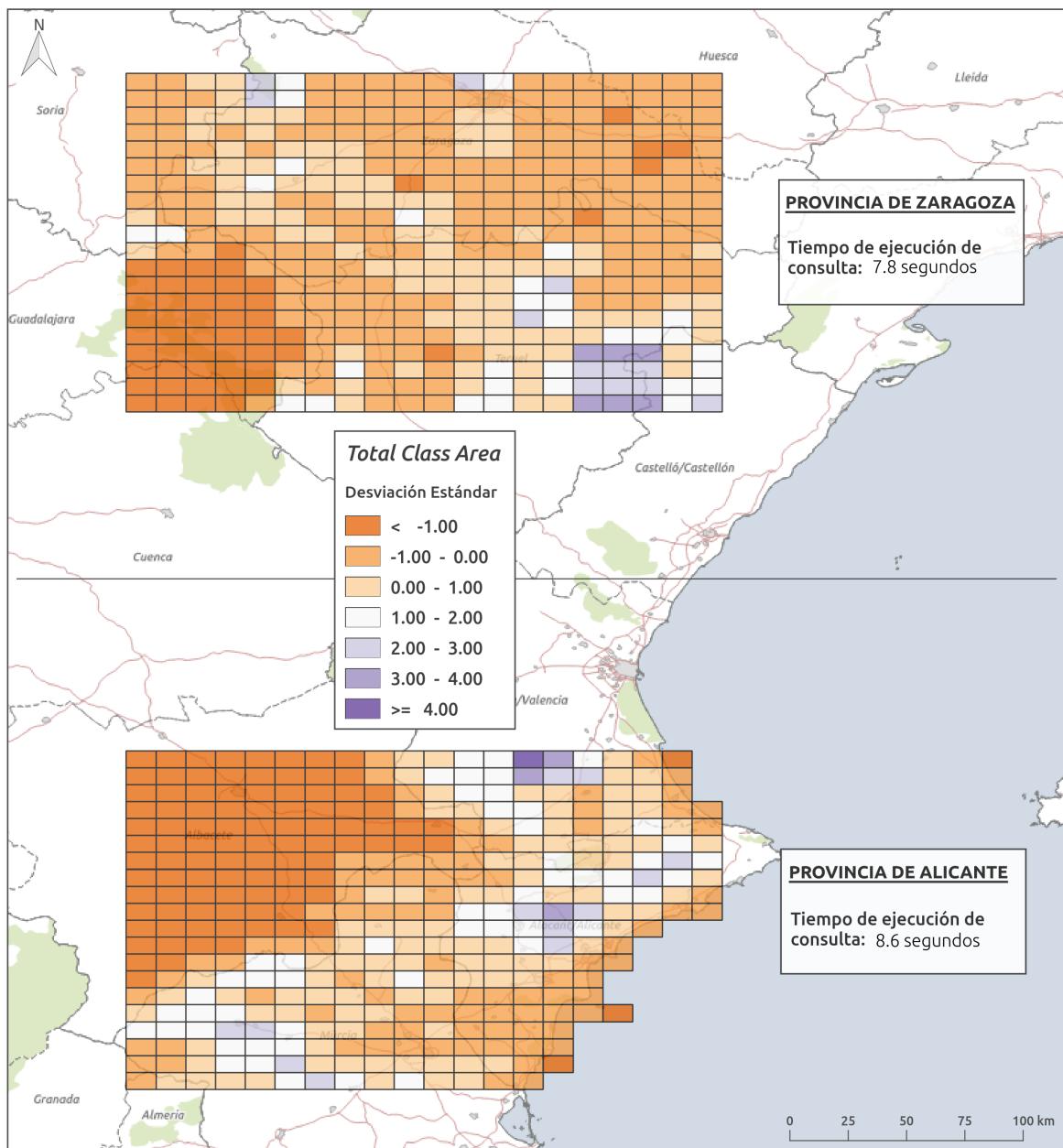


Figura B.4 Título.

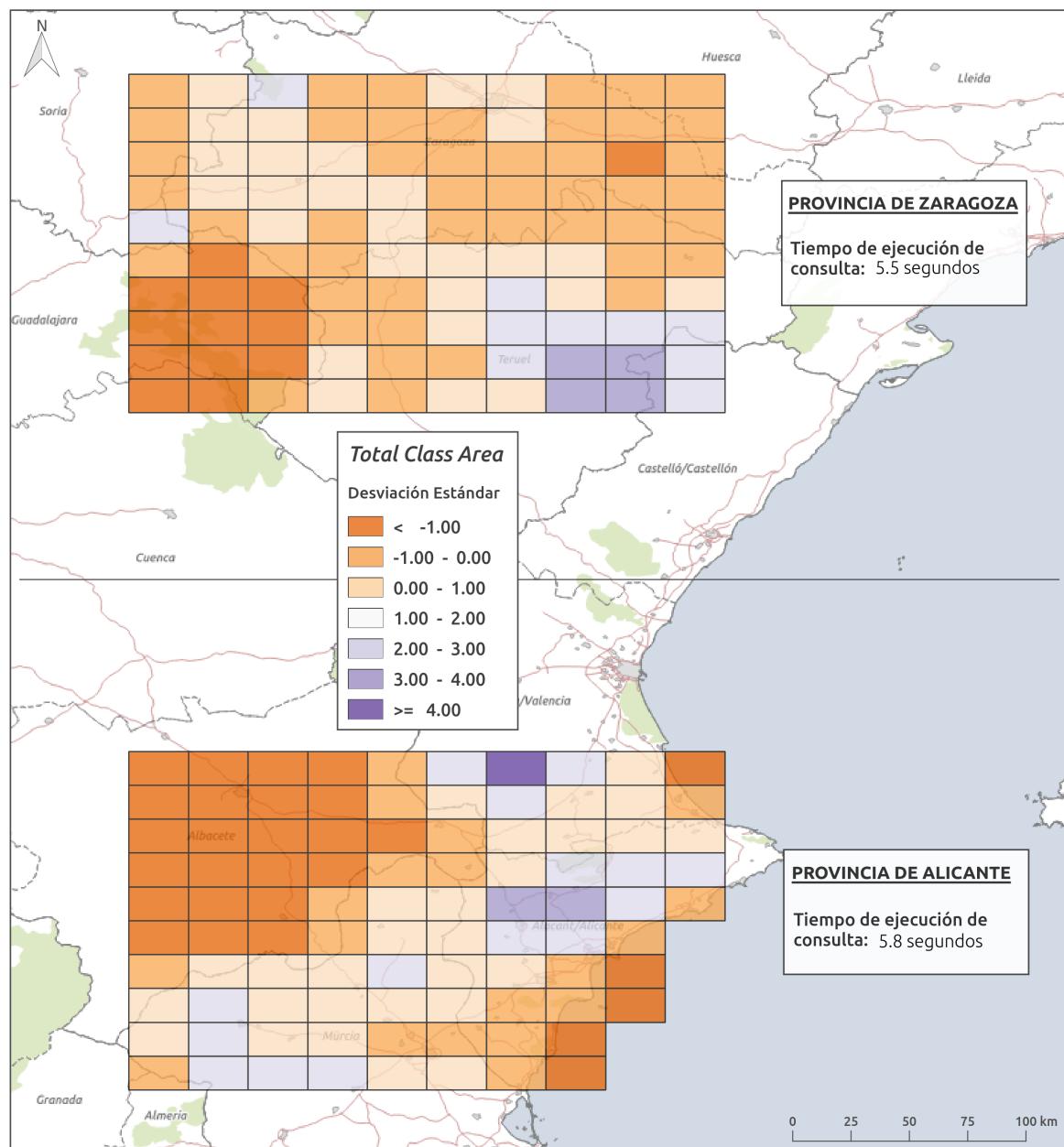


Figura B.5 Título.

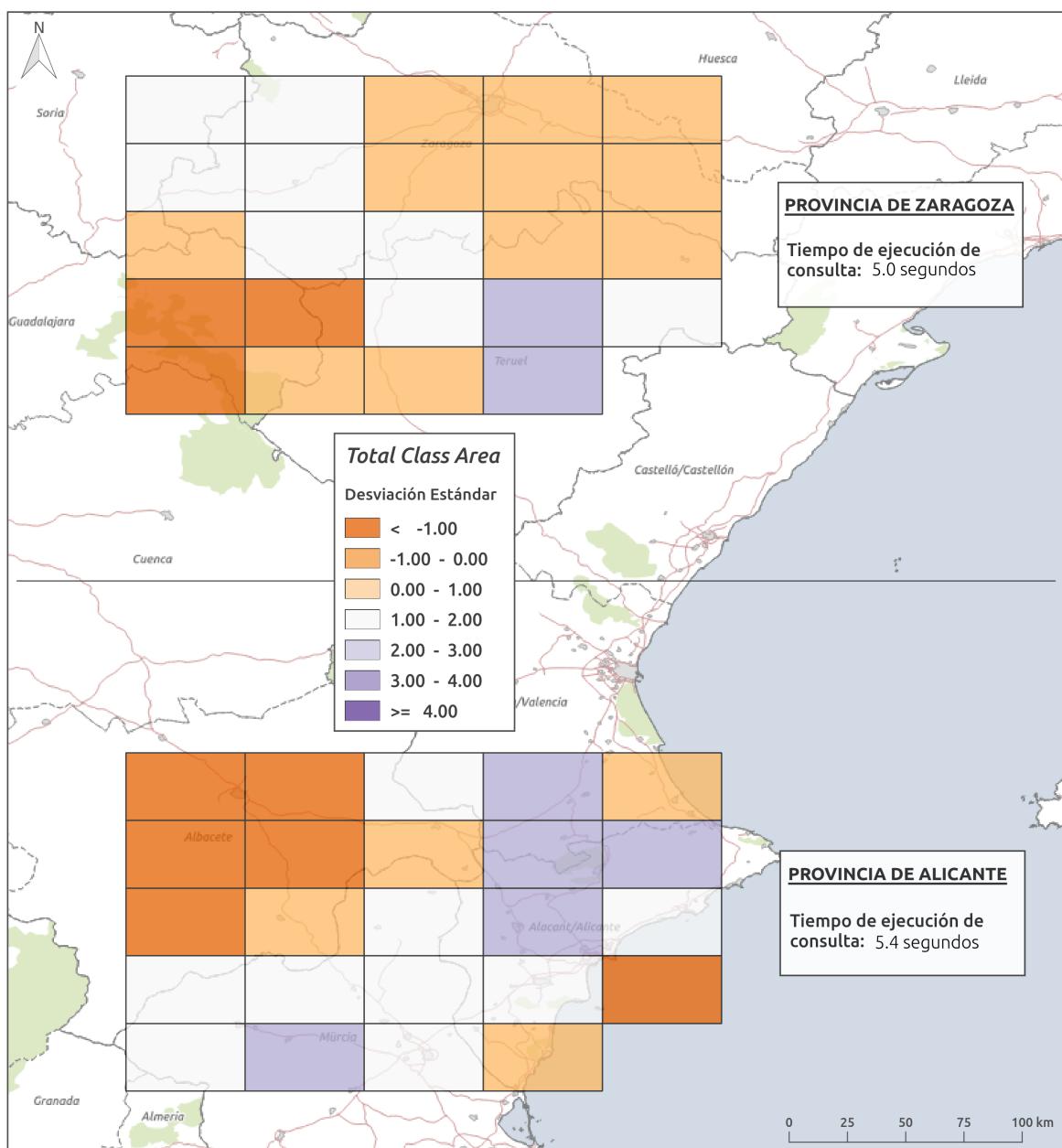


Figura B.6 Título.

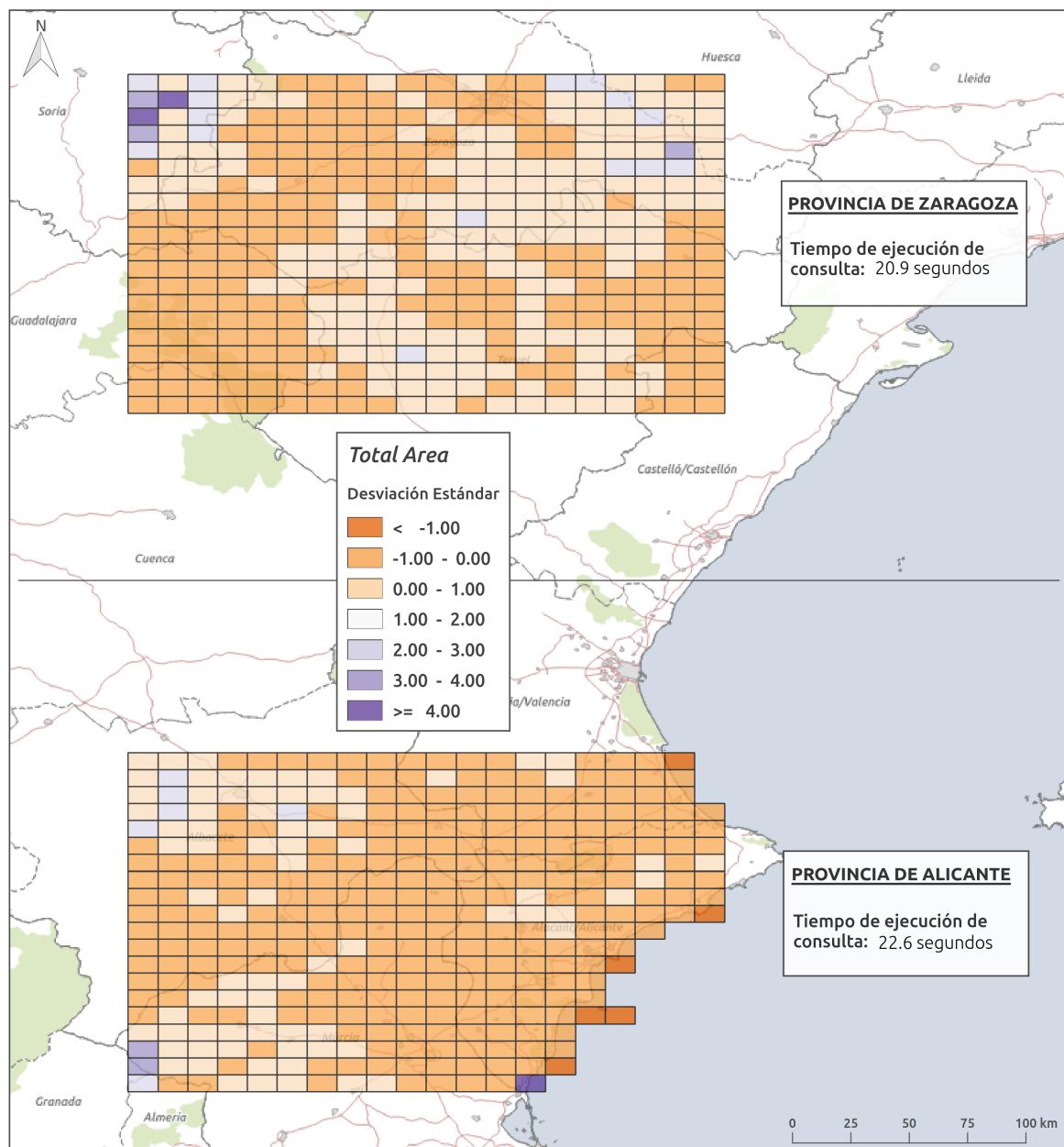


Figura B.7 Título.

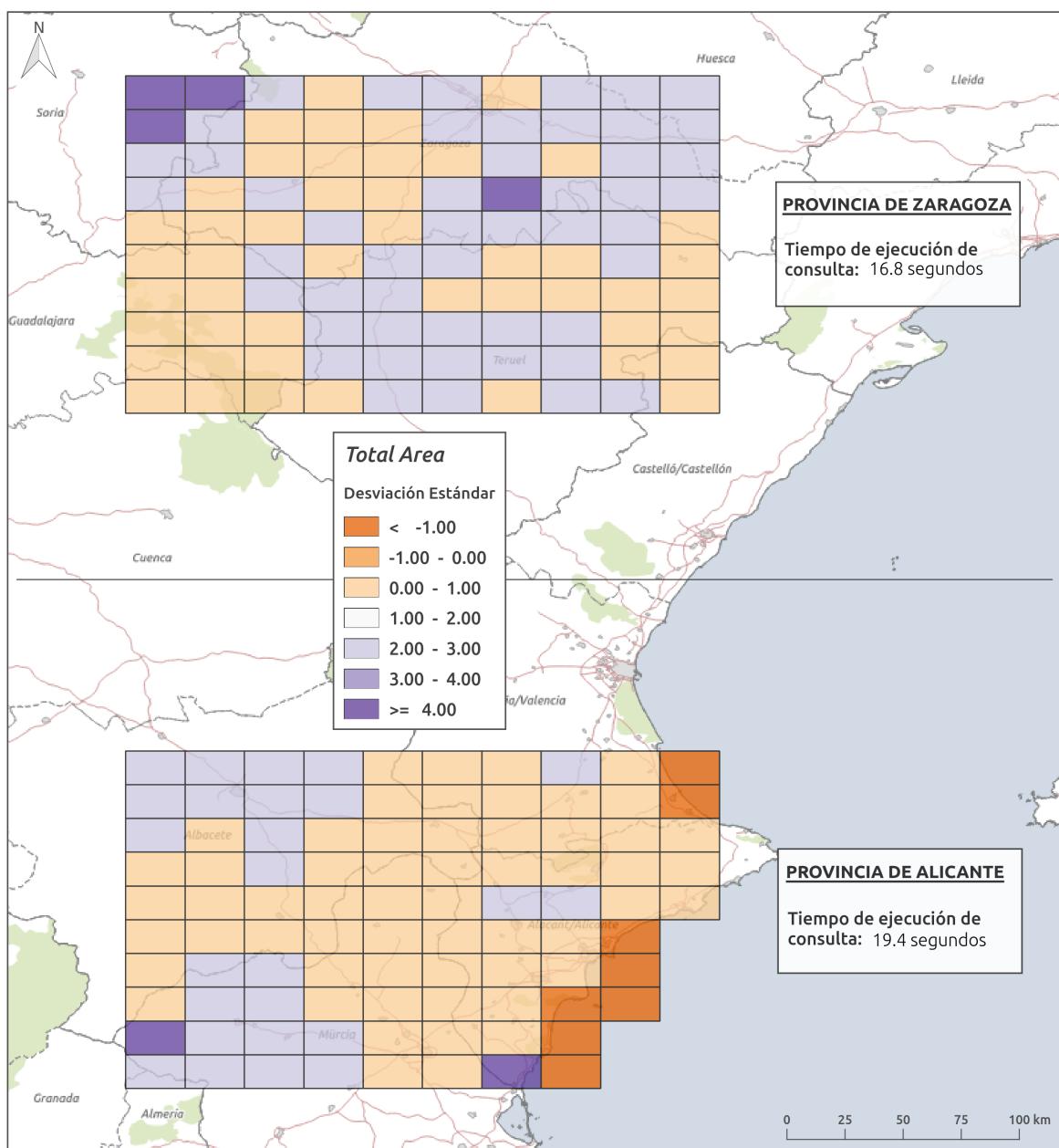


Figura B.8 Título.

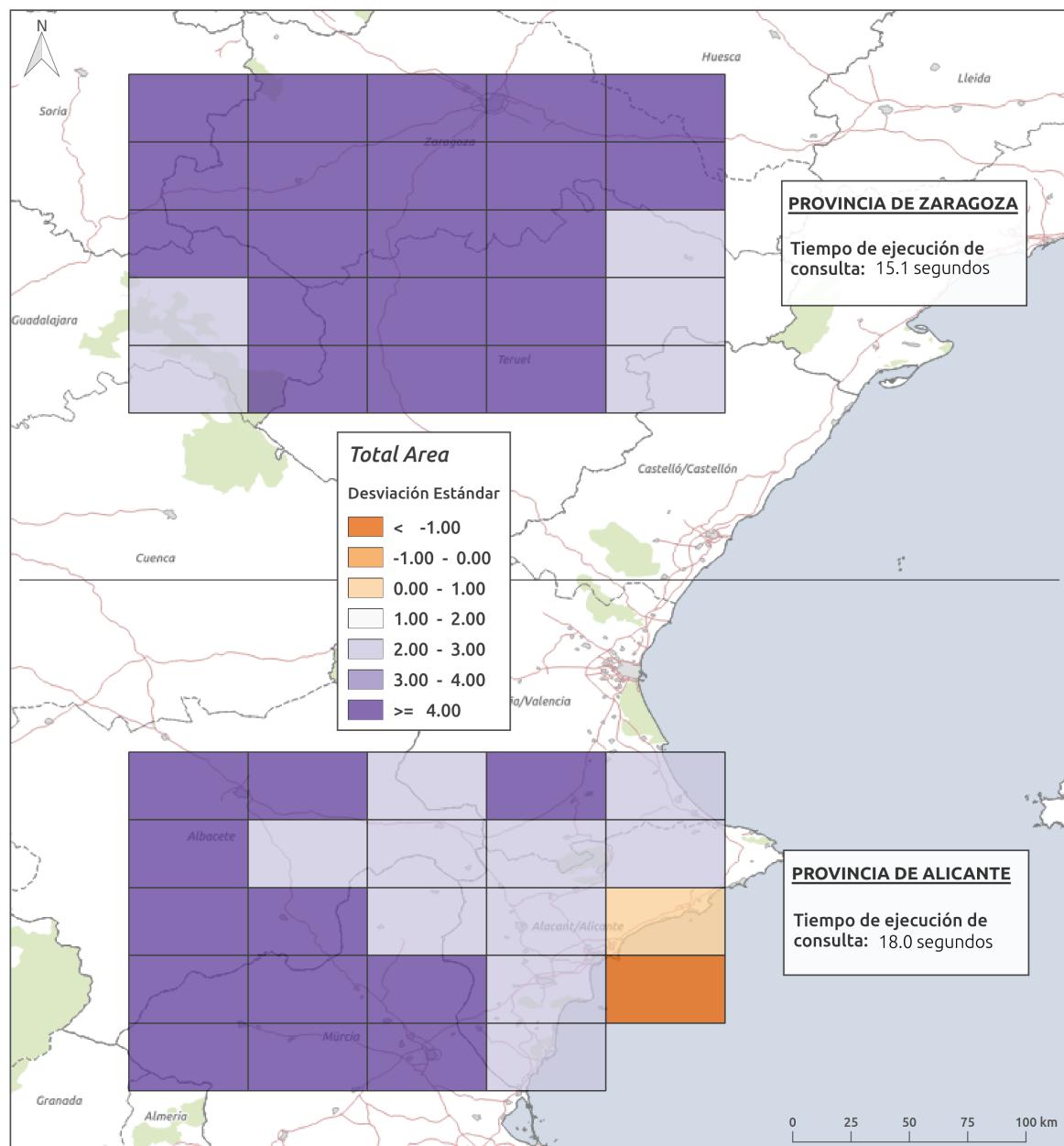


Figura B.9 Título.

C. DOCUMENTACIÓN SVG

Entre los distintas técnicas utilizadas para documentar el desarrollo realizado, se ha creado una función SQL que crea un fichero de gráficos vectoriales (SVG) para el resultado del cálculo de una métrica.

El código de esta función es relativamente complejo, tal y como se muestra en el anexo C, pero su uso sería realmente sencillo como se demuestra en el ejemplo de código ??.

Esta función será útil también para los usuarios de la extensión que quieran elaborar salidas gráficas similares a las de la figura ???. Además, es una función que puede ser reutilizada en otros proyectos que requieran este tipo de salidas gráficas.

Ejemplo de código C.1 Función para pintar gráficos vectoriales escalables (SVG).

```
1 -- SELECT patches_toSVG(id::integer, geom, category, svg_color,
2   p_area(geom)) FROM sample_patches_25830;
3 -- SELECT patches_toSVG(id::integer, geom, category, svg_color,
4   p_area(geom)) FROM sample_patches_4326;
5
6 CREATE OR REPLACE FUNCTION patches_toSVG(gid integer, geom
7   geometry, category text, svg_color text, metric metric)
8 RETURNS TEXT AS
9 $func$
10
11 WITH sp(gid,geom,category, svg_color) AS (VALUES
12   (gid,geom,category,
13   svg_color)),
14 paths (svg) AS (
15   SELECT array_to_string(
16     array_agg(
17       concat(
```

```
14          '<path d="', ST_AsSVG(geom,0), '" , " stroke="'
15              black" stroke-width="2" fill="', svg_color, '
16              "/>',
17          )) , ')
18      FROM sp
19 ),
20 texts (svg) AS(
21     SELECT array_to_string(
22         array_agg(
23             concat(
24                 '<text x="', st_x(st_pointonsurface(geom)), '" y="'
25                 , -st_y(st_pointonsurface(geom)), '" font-size'
26                 ="100px" fill="black" text-anchor="middle">',
27                 metric.value , ' ', metric.units , '</text>'
28             )) , ')
29     FROM sp
30 ),
31 env AS(
32     SELECT st_extent(geom) AS extent
33     FROM sp
34 ),
35 dims AS (
36     SELECT st_ymin(extent) AS bottom, st_ymax(extent) AS top,
37             st_xmin(extent) AS lefthand, st_xmax(extent) AS righthand
38     FROM env
39 )
40
41     SELECT concat(
42         '<svg width="100%"'
43         preserveAspectRatio="xMinYMin meetOrSlice"
44         style="border: 1px solid #cccccc;">',
45         '<g transform= "translate(0, 0) scale (0.1, 0.1) translate('
46             , - dims.lefthand , ' , ' , dims.top , ')">',
47         paths.svg , texts.svg ,
48         '</g> ,
49         '</svg>')
50
51     FROM dims , paths , texts;
52
53
54 $func$
55 LANGUAGE sql;
```

[REDACTED]

