

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 ha sido posible gracias al apoyo y ayuda de muchas personas a las que me gustaría agradecer y también por todo el conocimiento que he obtenido gracias a ellos a lo largo de esta etapa.

En primer lugar, he de dar las gracias a mis directores Raquel Montorio Llovería y Daniel Borini Alves de la Universidad de Zaragoza quienes han dirigido este trabajo, y 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.

También me gustaría agradecer especialmente a los compañeros del laboratorio por su colaboración y paciencia a lo largo de las prácticas como también por su apoyo y ayuda cuando lo necesitaba: B.M. Zaragozaí Zaragozaí. J. Torres Prieto y J.T. Navarro Carrión. Gracias por hacerme sentir como si fuera una compañera más.

Finalmente, y no menos importante, a mi familia por su apoyo y comprensión, como también a todos mis amigos y compañeros del máster de la Universidad de Zaragoza.

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 3.2).

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,

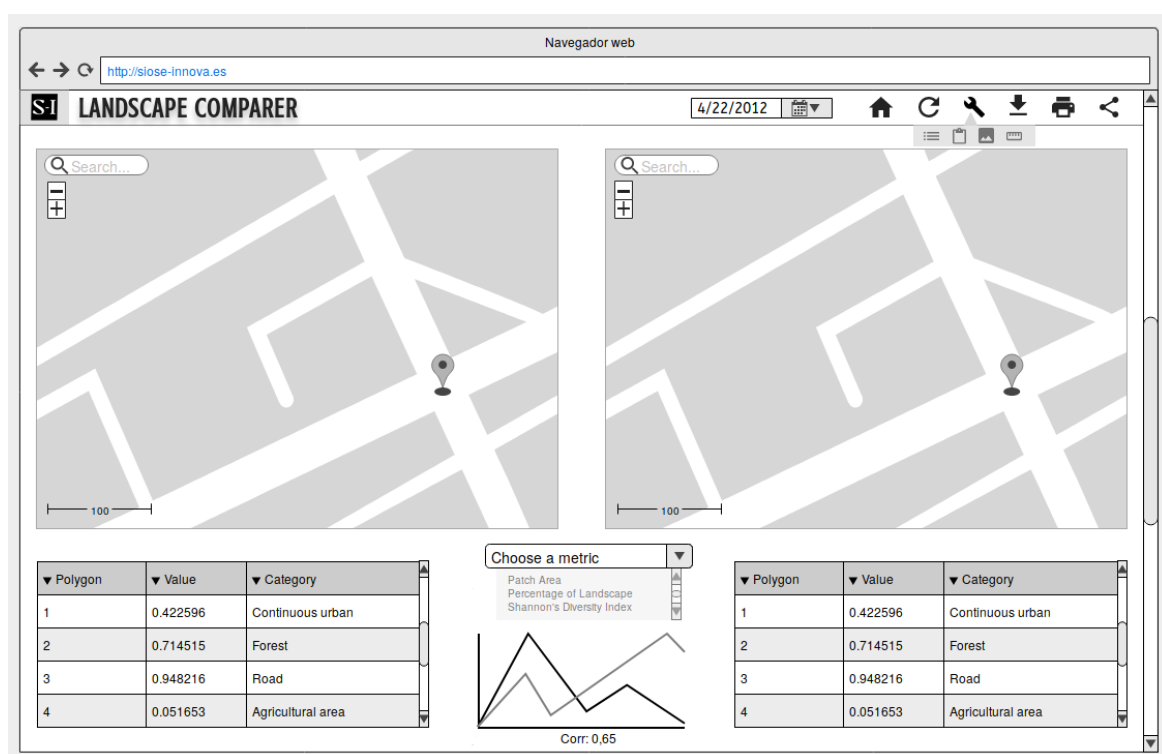


Figura 1 Prototipo de un visor cartográfico para el análisis (comparación) de la estructura del paisaje a partir del SIOSE. La extensión sobre PostgreSQL/PostGIS desarrollada en este trabajo (*pg_landmetrics*) es la base para este tipo de aplicaciones.

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 útiles 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 analyze 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	XIX
Ejemplos de código	XX
1. Introducción	1
1.1. Estudio de la estructura del paisaje utilizando el SIOSE	2
1.2. Métricas de paisaje y software que las calcula	5
1.3. Objetivos	6
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	14
2.1.3. Extensibilidad en PostgreSQL	17
2.1.4. Otras aplicaciones	18
2.2. Conjuntos de datos	20
2.3. Selección de métricas	23
2.4. Implementación/desarrollo de funciones en PostgreSQL	23

2.5. Documentación de la extensión	30
3. Resultados y Discusión	33
3.1. pg_landmetrics	33
3.2. Caso de uso sobre el SIOSE-2011	36
4. Conclusiones y trabajo futuro	37
Bibliografía	39
Apéndice A. Funciones tipo	43
Apéndice B. Documentación SVG	51

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
2.1.	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>	13
2.2.	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.	15
2.3.	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.	17
2.4.	Vista de la interfaz web de PgAdmin4.	19
2.5.	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.	21
3.1.	Título.	34

3.2. Título.	34
3.3. Título.	34

Índice de tablas

2.1. Métricas de paisaje disponibles en la extensión.	24
3.1. Listado de las métricas de paisaje disponibles en la extensión.	35
3.2. Características de los conjuntos de datos utilizados.	36

Ejemplos de código

2.1. Consulta SQL para la métrica AREA.	26
2.2. Función simple para calcular la métrica AREA por polígonos.	26
2.3. Ejemplo de uso para calcular la métrica AREA.	27
2.4. Consulta SQL para la métrica TCA.	27
2.5. Función de agregado para calcular TCA por categorías.	27
2.6. Ejemplo de uso para calcular la métrica TCA.	29
2.7. Extracción de los resultados de las métricas en diferentes escalas de referencia a partir de grids.	29
A.1. Función para calcular AREA de un polígono de PostGIS de tipo <i>geography</i>	43
A.2. Función para calcular CA por categorías.	45
A.3. Función para calcular TA del paisaje.	48
B.1. Función para pintar gráficos vectoriales escalables (SVG).	51

1. INTRODUCCIÓN

La interacción de los factores naturales y antrópicos es el origen de la estructura espacial compleja y heterogénea que presenta el paisaje (Forman and Godron, 1986; Turner et al., 2001). 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** (Aguilera, 2010). Hoy en día, para estudiar la estructura del paisaje se dispone de **gran cantidad de información y herramientas**.

En primer lugar, 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. Estudio de la estructura del paisaje utilizando el SIOSE

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 (p.ej. desde aplicaciones SIG de escritorio).

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. El SIOSE es un modelo orientado a objetos (entidad-relación) que describe los objetos,

¹<http://www.ign.es/web/ign/portal>

atributos y relaciones, y que 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* ([Equipo Técnico Nacional SIOSE, 2015](#)). 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)*.

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 ([Arnold et al., 2013](#)). Gracias a la iniciativa de EAGLE, el Instituto Geográfico Nacional de España (IGN) crea el SIOSE que se ha convertido en un repositorio de ocupación del suelo de referencia a nivel europeo ([Equipo Técnico Nacional SIOSE, 2015](#)). Así pues, uno de los objetivos futuros del SIOSE es obtener una base de datos de ocupación del suelo europea para que resulte más fácil trabajar entre fronteras.

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 ([Giné, 2014](#); [Hamilton and Mather, 2009](#); [Hebeisen et al., 2008](#)), ([Brennan and Schnell, 2005](#); [Giménez Font, 2010](#); [Lin et al., 2014](#)).
- Demografía, urbanismo y planificación del territorio ([Aguilera et al., 2011](#)), ([Aguilera, 2010](#); [Blaschke and Petch, 1999](#); [Jacquin et al., 2008](#); [Prastacos et al., 2017](#); [Tudor et al., 2014](#)).
- Infraestructuras, energía y transporte.
- Dinámica de la ocupación del suelo ([Dunk et al., 2011](#); [Herold et al., 2002](#); [Van der Kwast et al., 2011](#)), ([Aguilera and Botequilha-Leitão, 2012](#); [Liu et al., 2016](#); [Roces-Díaz et al., 2014](#); [Rodríguez-Rodríguez and Martínez-Vega, 2017](#)).

También se han realizado estudios sobre abandono agrícola ([Zaragozí et al., 2011](#)) y otros sobre el riesgo de incendio asociado a nuevas formas de ocupación del suelo [Vázquez \(2017\)](#).

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) (Fernández Villarino et al., 2012). 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 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 (Navarro-Carrión et al., 2016).

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 Fernández Villarino et al. (2012). 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 ([Giné, 2014](#)).

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 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 ([Uuemaa et al., 2009](#)). Por ejemplo, en [Uuemaa \(2017\)](#) 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 [Zaragozí et al. \(2012\)](#) se establece una comparativa entre **programas específicos** para el cálculo de métricas del paisaje, destacando entre ellos FRAGSTATS ([McGarigal, 2015](#); [McGarigal and Marks, 1994](#)). Por otro lado, hay otros programas como Conefor Sensinode ([Saura and Torné, 2009](#)), Patch Analyst, varios módulos de GRASS GIS, LecoS ([Jung, 2016](#)), ZonalMetrics ([Adamczyk and Tiede, 2017](#)), 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 capaz de realizar análisis sobre bases de datos de ocupación del suelo tan voluminosas y complejas como lo es la del SIOSE.

1.3. Objetivos

El principal objetivo de este trabajo es **desarrollar una extensión (PostgreSQL/Post-GIS) 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 [Zaragozí et al. \(2017\)](#) 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 *orquestración 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 periódicamente 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/>).

- **GitHub**⁶ 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:

- 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 2.1). 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

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

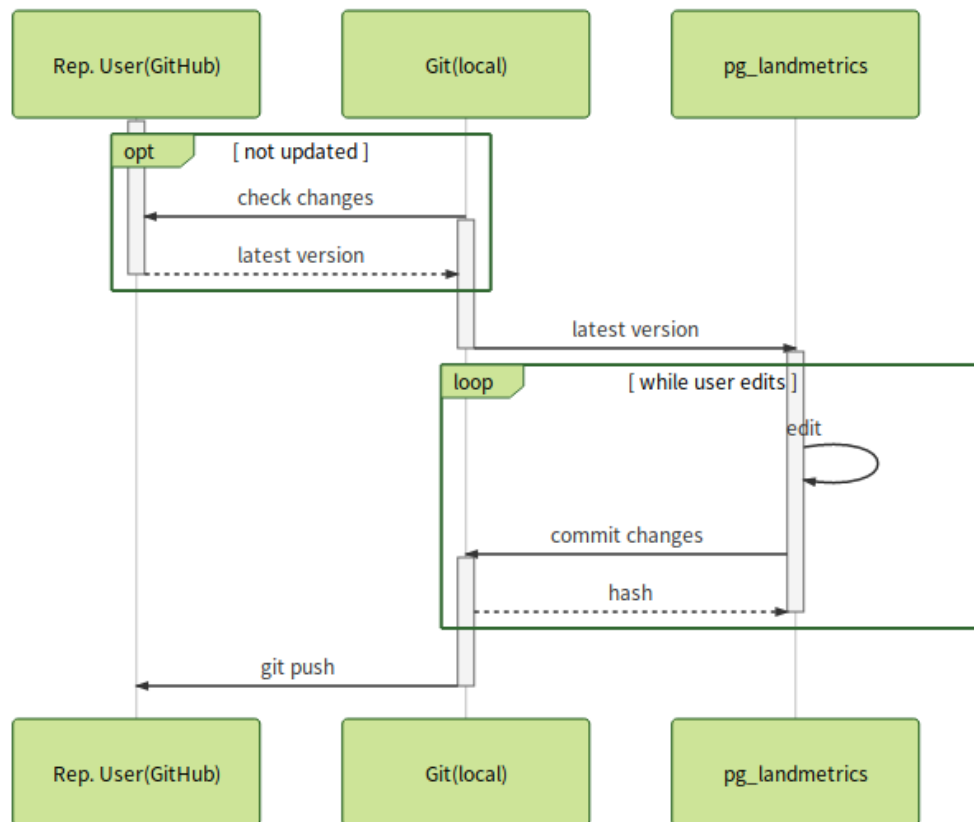


Figura 2.1 Diagrama de secuencia en el que se describe el flujo de integración continua más habitual en este trabajo (varias veces al día). Cada vez que se realizaban cambios importantes en la extensión *pg_landmetrics*, se añadían al control de versiones local y al de *La Nube de GitHub*.

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 2.2). 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 contra-tiempo, 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 2.1. Una vez subidos los cambios al *fork* de GitHub, es posible enviar una petición al repositorio oficial (*upstream*) para sincronizar ambos repositorios. Esta 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.

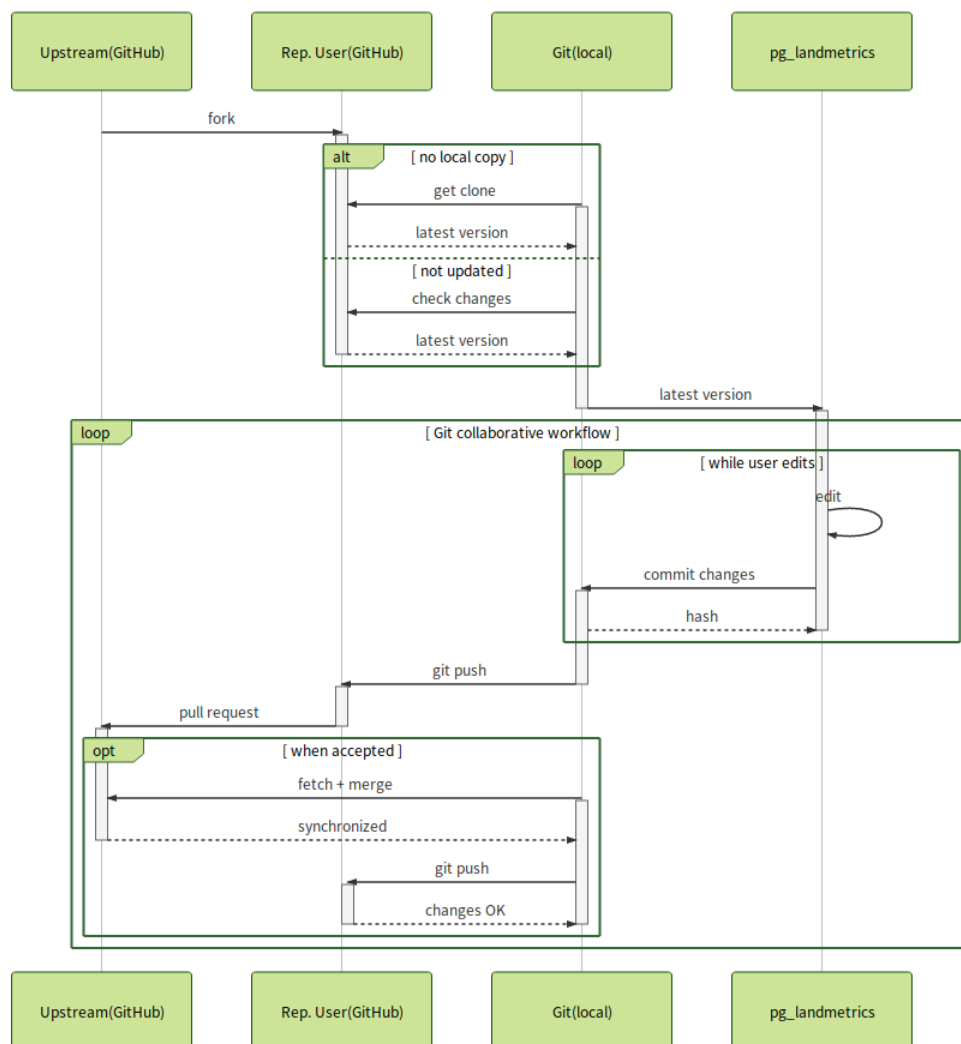


Figura 2.2 Diagrama de secuencia sobre el desarrollo colaborativo entre repositorios. Aproximadamente una vez al día, se proponen una serie de cambios al repositorio oficial (*pull request*), si estos cambios son aceptados después es posible sincronizar los dos repositorios.

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.
- **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 2.3). 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 intrucción**. 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,

¹¹<https://kubernetes.io/>

¹²<https://www.docker.com/>

¹³<https://hub.docker.com/>

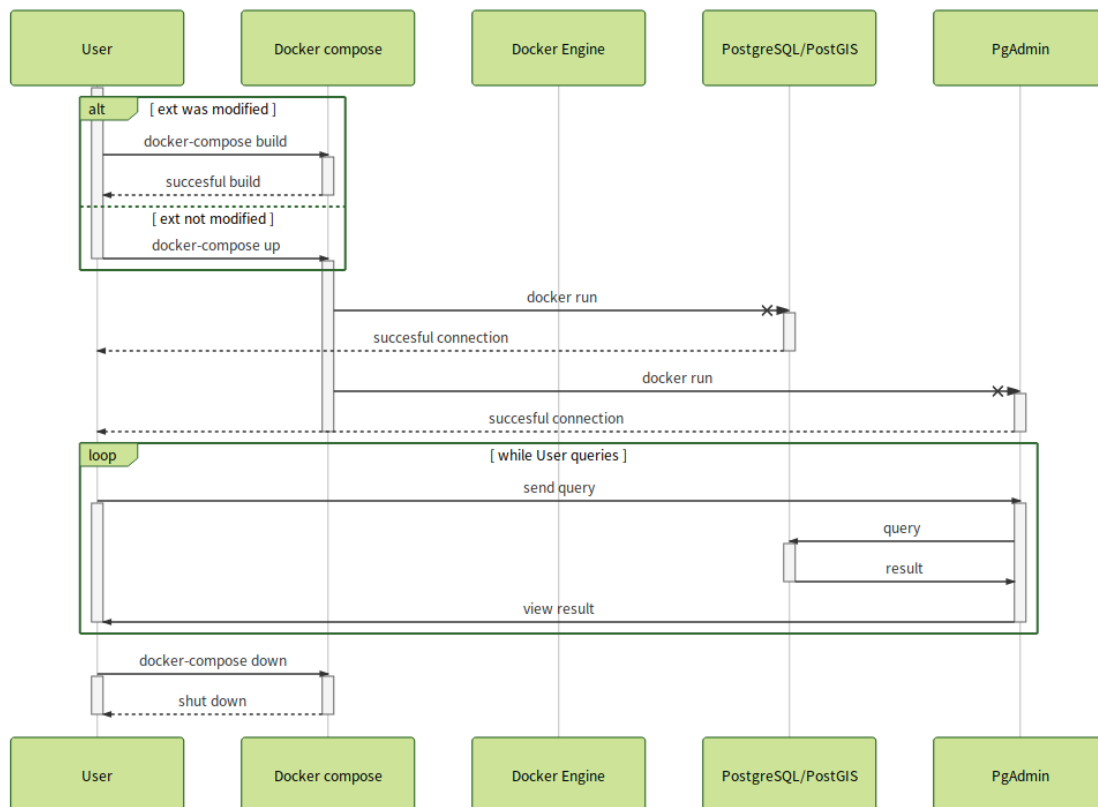


Figura 2.3 Diagrama de secuencia. Integración continua en la implementación y desarrollo de funciones SQL con *dockers*. Cada vez que una nueva función es añadida a la base de datos, se elimina cualquier rastro de la versión anterior y se vuelve a lanzar un nuevo contenedor para evaluar si los cambios son válidos.

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.

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.

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

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>

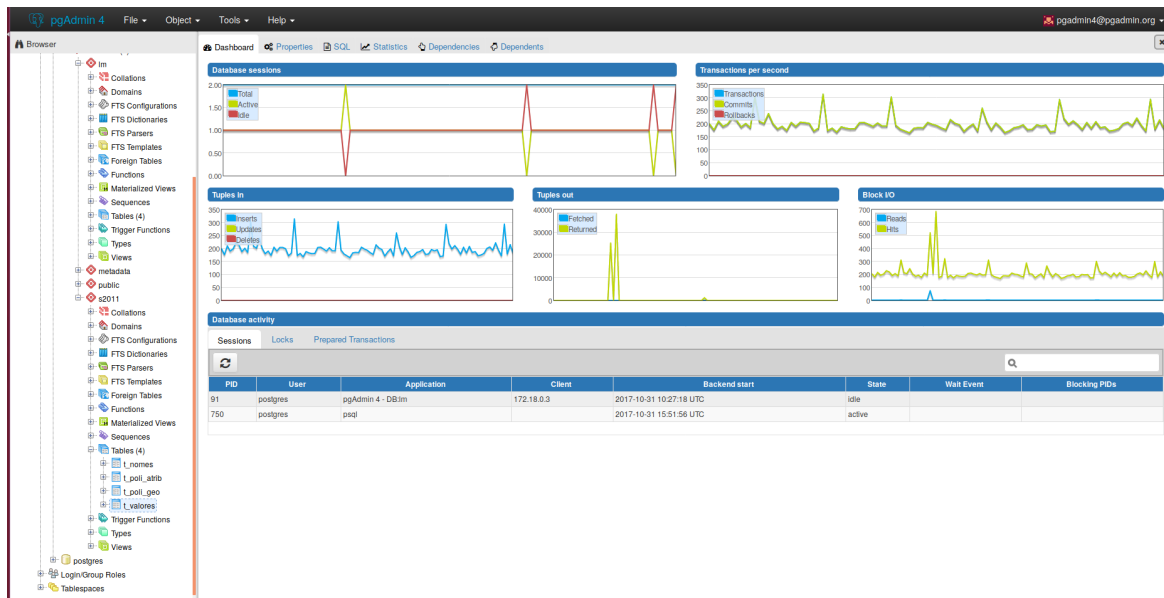


Figura 2.4 Vista de la interfaz web de PgAdmin4.

- **PgAdmin4**¹⁶: es una plataforma que administra, gestiona y desarrolla código abierto en bases de datos PostgreSQL. En la Figura 2.4 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 trabajó 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 2.18**¹⁷: 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 geoprocso 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 2.5). 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/colorsvg.html>

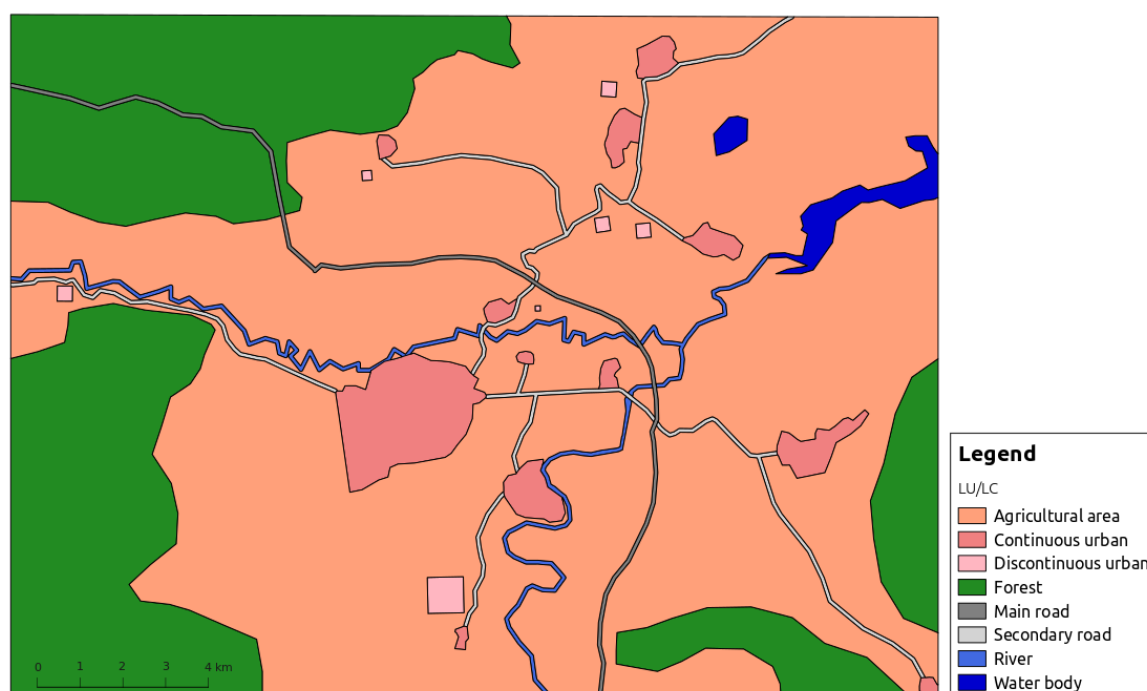


Figura 2.5 Coberturas del suelo del paisaje de ejemplo. Estas geometrías se distribuyen junto con la extensión *pg_landmetrics* para que los usuarios realicen pruebas antes de importar sus propios datos.

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, toponimia, *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 que consiste la simulación o experimento que se analiza en el capítulo 3.

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*.

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 2.1 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 vecino más próximo o la densidad.

2.4. Implementación/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 language SQL es el más sencillo para programar en PostgreSQL y está mejor integrado.

Tabla 2.1 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

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 2.1). 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.

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.

Ejemplo de código 2.1 Consulta 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;
```

A partir de ésta, se ha desarrollado la función la cual sigue una estructura sencilla. 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, 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*.

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 desarrollado la función, se ha ejecutado un ejemplo de uso (ver ejemplo de código 2.4) para obtener el cálculo de la métrica AREA a partir de un conjunto de datos (ver en la figura ref polig area).

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;
```

Sin embargo, las funciones de agregado son distintas a las funciones simples. Éstas corresponden a las métricas de paisaje tanto a nivel de categoría como a nivel de paisaje. En este caso, como ejemplo a nivel de categoría, la métrica de paisaje *Total Core Area (TCA)* devuelve la suma de los núcleos de las áreas (m2) de cada polígono correspondiente al tipo de polígono, dividido por 10.000 (unidades: Hectáreas). 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.

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, 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 CREATE AGGREGATE c_totalcorearea(geometry, text)(
28     SFUNC=c_totalcorearea_state,
29     STYPE=metric_labeled,
30     INITCOND='(13,{})'
31 );
32
33 COMMENT ON AGGREGATE c_totalcorearea(geom geometry, category
34     text) IS 'Suma las áreas de los núcleos de cada polígono de
35     la misma categoría dividido por 10.000 para devolver un valor
36     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 ref polig tca).

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

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

En cuanto a las métricas de paisaje a nivel de paisaje también se han desarrollado de tipo agregado para obtener un solo resultado en 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.

Finalmente, con todas las funciones implementadas, desarrolladas y testeadas, el conjunto de datos del SIOSE 2011 preparado y todos los grids dispuestos en sus diferentes escalas, se ha elaborado una última función para obtener todos los resultados de este trabajo que se encuentran en el capítulo 3 (ver ejemplo de código 2.4).

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(' ||  
    quote_literal(table_schema || '.' || table_name) || ')', '  
    UNION ALL ') into script  
16  FROM information_schema.tables where table_schema = 'test' and  
    table_name like 'grid_%';  
17  
18  RETURN QUERY EXECUTE script;  
19  
20 END  
21 $$;
```

El que realiza esta función concretamente es construir una tabla virtual con los resultados organizados por columnas según la zona de estudio a la que corresponden, el grid sobre el que está intersectado, y los valores de cada métrica en su columna correcta. Luego se ha ejecutado un *script* el cual extrae todos los valores calculados en todas las escalas de referencia en una sola tabla que más adelante se ha extraído como csv para manejar los resultados.

2.5. 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, **sobretudo 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 facilita la aplicación de todas las medidas necesarias para llevar a cabo el funcionamiento de la extensión a cualquier usuario, sobretudo a aquellos menos expertos en la materia. Así pues, se han aplicado los siguientes lenguajes de marcado:

- **Markdown**²¹ 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.
- **TeX**²² 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 y se ha utilizado para escribir este trabajo.
- **Scalable Vector Graphics (SVG)**²³ 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 B).
- **Mermaid**²⁴ es un lenguaje que genera gráficos a partir de texto mediante JavaScript. Se han generado desde diagramas de flujo hasta diagramas de secuencia y de Gantt.

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).

Intro a los resultados... código/commits y caso de uso

En la subsección 3.1 se describen cuantitativamente las aportaciones realizadas en este proyecto 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 pone a prueba la extensión desarrollada. 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 que en otras aplicaciones no es posible, como las citadas en el capítulo de [Introducción](#).

3.1. `pg_landmetrics`

Cuantitativamente, detalles de mi aportación frente al total de la extensión... y para cada una un comentario o valoración personal.

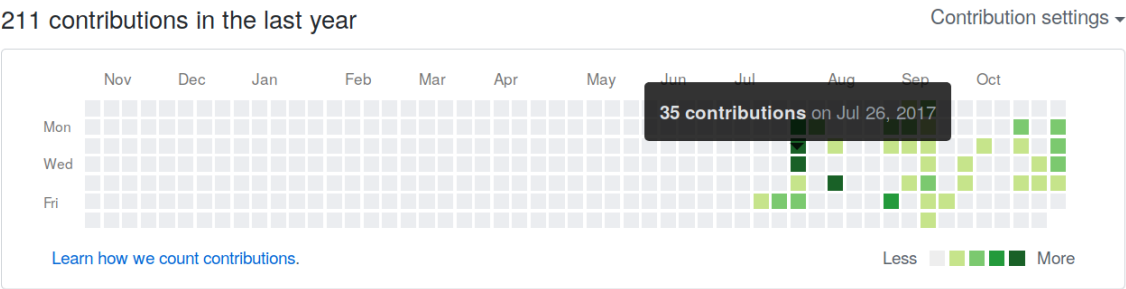


Figura 3.1 Título.

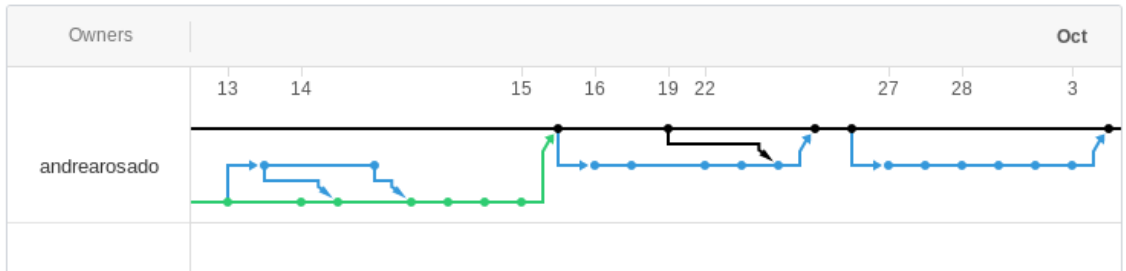


Figura 3.2 Título.

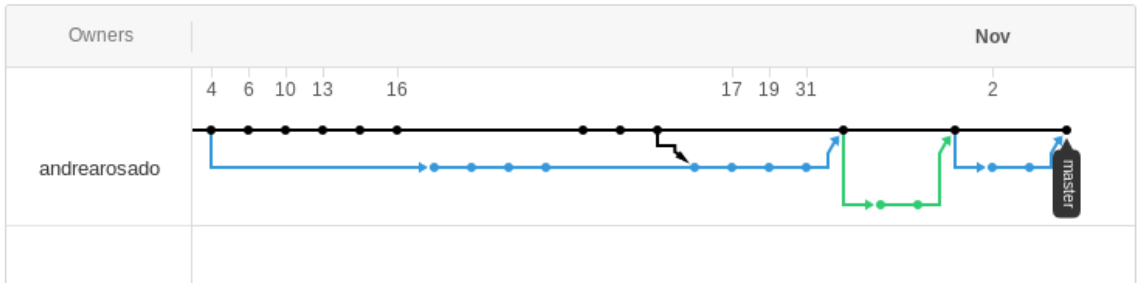


Figura 3.3 Título.

Tabla 3.1 Listado de las métricas de paisaje disponibles en la extensión.

Métrica	Consulta QGIS	Consulta SQL	Extensión
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 de agregado
●	Disponible
○	No disponible

Tabla 3.2 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

3.2. Caso de uso sobre el SIOSE-2011

Listing con la consulta final, una única línea frente a las n líneas que serían necesarias en una única consulta SQL.

Figuras correlación entre dos zonas de estudio grandes...

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

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.

Bibliografía

- Adamczyk, J. and Tiede, D. (2017). ZonalMetrics - a Python toolbox for zonal landscape structure analysis. *Computers and Geosciences*, 99(October 2016):91–99.
- Aguilera, F. (2010). Aplicación de métricas de ecología del paisaje para el análisis de patrones de ocupación urbana en el Área Metropolitana de Granada. *Anales de geografía de la Universidad Complutense*, 30:9–29.
- Aguilera, F. and Botequilha-Leitão, A. (2012). Selección de métricas de paisaje mediante análisis de componentes principales para la descripción de los cambios de uso y cobertura del suelo del Algarve, Portugal. *GeoFocus*, 12:93–121.
- Aguilera, F., Valenzuela, L. M., and Botequilha-Leitão, A. (2011). Landscape metrics in the analysis of urban land use patterns: A case study in a Spanish metropolitan area. *Landscape and Urban Planning*, 99(3-4):226–238.
- Arnold, S., Kosztra, B., Banko, G., Smith, G., Hazeu, G., and Bock, M. (2013). The EAGLE concept A vision of a future European Land Monitoring Framework. In *EARSeL Symposium proceedings 2013, Towards Horizon 2020*, pages 551–568.
- Blaschke, T. and Petch, J. (1999). Landscape structure and scale: comparative studies on some landscape indices in Germany and the UK. *Heterogeneity in Landscape Ecology: Pattern and Scale*, pages 75–84.
- Brennan, S. P. and Schnell, G. D. (2005). Relationship between bird abundances and landscape characteristics: The influence of scale. *Environmental Monitoring and Assessment*, 105(1-3):209–228.
- Dunk, A. V. D., Grêt-regamey, A., Dalang, T., and Hersperger, A. M. (2011). Defining a typology of peri-urban land-use conflicts - A case study from Switzerland. *Landscape and Urban Planning*, 101(2):149–156.
- Equipo Técnico Nacional SIOSE (2015). Documento Técnico SIOSE 2011. Technical report, D.G. Instituto Geográfico Nacional. Servicio de Ocupación del Suelo. S.G de Cartografía, Madrid.
- Fernández Villarino, X., Delgado Hernández, J., Valcárcel Sanz, N., Caballero, M. E., Benito Saz, M. A., and Porcuna Fernández-Monasterio, A. (2012). Geoservicios web SIOSE: un sistema de información como servicio público. In *VI Jornadas de SIG Libre*, page 8, Girona. Servei de Sistemes d’Informació Geogràfica i Teledetecció (SIGTE). Universitat de Girona.

- Forman, R. T. and Godron, M. (1986). *Landscape Ecology*. New York, New York, USA, John Wiley edition.
- Giménez Font, P. (2010). Sobre las repercusiones ambientales de las transformaciones del paisaje: oportunidades para la geografía histórica. *Investigaciones Geográficas*, 53:219–230.
- Giné, D. S. (2014). Unidades De Paisaje Naturales Y Unidades De Paisaje Artificiales. Comparación Mediante Sig Y Métricas De Paisaje. *Geofocus. Revista Internacional de Ciencia y Tecnología de la Información Geográfica*, 14:23–54.
- Hamilton, G. S. and Mather, P. B. (2009). Differential impacts of habitat heterogeneity on male and female connectivity in a spatially structured pest system. *Austral Ecology*, 34(1):41–49.
- Hebeisen, C., Fattebert, J., Baubet, E., and Fischer, C. (2008). Estimating wild boar (*Sus scrofa*) abundance and density using capture-resights in Canton of Geneva, Switzerland. *European Journal of Wildlife Research*, 54(3):391–401.
- Herold, M., Scepan, J., and Clarke, K. C. (2002). The use of remote sensing and landscape metrics to describe structures and changes in urban land uses. *Environment and Planning A*, 34(8):1443–1458.
- Jacquin, A., Misakova, L., and Gay, M. (2008). A hybrid object-based classification approach for mapping urban sprawl in periurban environment. *Landscape and Urban Planning*, 84(2):152–165.
- Jung, M. (2016). LecoS - A python plugin for automated landscape ecology analysis. *Ecological Informatics*, 31:18–21.
- Lin, Y. P., Huang, C. W., Ding, T. S., Wang, Y. C., Hsiao, W. T., Crossman, N. D., Lengyel, S., Lin, W. C., and Schmeller, D. S. (2014). Conservation planning to zone protected areas under optimal landscape management for bird conservation. *Environmental Modelling and Software*, 60:121–133.
- Liu, Y., Wei, X., Li, P., and Li, Q. (2016). Sensitivity of correlation structure of class- and landscape-level metrics in three diverse regions. *Ecological Indicators*, 64:9–19.
- McGarigal, K. (2015). Fragstats.Help.4.2. Technical Report April.
- McGarigal, K. and Marks, B. J. (1994). FRAGSTATS: spatial pattern analysis program for quantifying landscapesStructure. *General Technical Report PNW-GTR-351*. U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station. Portland, OR, 97331(503):134.
- Navarro-Carrión, J., Zaragoza, B., Ramón-Morte, A., and Valcárcel-Sanz, N. (2016). Should EU land use and land cover data be managed with a NOSQL document store? *International Journal of Design & Nature and Ecodynamics*, 11(3):438–446.
- Prastacos, P., Lagarias, A., and Chrysoulakis, N. (2017). Using the Urban Atlas dataset for estimating spatial metrics. Methodology and application in urban areas of Greece. *Cybergeo*, (May):1–27.

- Roces-Díaz, J. V., Díaz-Varela, E. R., and Álvarez-Álvarez, P. (2014). Analysis of spatial scales for ecosystem services: Application of the lacunarity concept at landscape level in Galicia (NW Spain). *Ecological Indicators*, 36:495–507.
- Rodríguez-Rodríguez, D. and Martínez-Vega, J. (2017). Assessing recent environmental sustainability in the Spanish network of National Parks and their statutory peripheral areas. *Applied Geography*, 89(September):22–31.
- Saura, S. and Torné, J. (2009). Conefor Sensinode 2.2: A software package for quantifying the importance of habitat patches for landscape connectivity. *Environmental Modelling and Software*, 24(1):135–139.
- Tudor, C. A., Ioj, I. C., Ptru-Stupariu, I., Nit, M. R., and Hersperger, A. M. (2014). How successful is the resolution of land-use conflicts? A comparison of cases from Switzerland and Romania. *Applied Geography*, 47:125–136.
- Turner, M. G., Gardner, R. H., and O'Neill, R. (2001). *Landscape Ecology in Theory and Practice*.
- Uuemaa, E. (2017). *Mapping Forest Landscape Patterns*.
- Uuemaa, E., Antrop, M., Roosaare, J., and Marja, R. (2009). Landscape Metrics and Indices: An Overview of Their Use in Landscape Research Living Reviews in Landscape Research. *Living Reviews in Landscape Research*, 3:1–28.
- Van der Kwast, J., Van de Voorde, T., Canters, F., Uljee, I., Van Looy, S., and Engelen, G. (2011). Inferring urban land use using the optimised spatial reclassification kernel. *Environmental Modelling and Software*, 26(11):1279–1288.
- Vázquez, A. (2017). *Las TIG para la delimitación de las ZIUF de Náquera (Camp de Turia)*. Master, Universidad de Alicante.
- Zaragozí, B., Belda, A., Linares, J., Martínez-Pérez, J., Navarro, J., and Esparza, J. (2012). A free and open source programming library for landscape metrics calculations. *Environmental Modelling & Software*, 31:131–140.
- Zaragozí, B. M., Navarro, J. T., Ramón, A., and Rodríguez-Sala, J. J. (2011). A study of drivers for agricultural land abandonment using GIS and Data Mining techniques. 144:363–374.
- Zaragozí, B. M., Navarro-Carrión, J. T., Torres-Prieto, J., Belda-Antolí, A., and Ramón-Morte, A. (2017). Investigación reproducible en ciencias de la información geográfica: una propuesta metodológica. In *XXV Congreso de la AGE*, pages 2659–2668, Madrid. Asociación de Geógrafos Españoles (AGE).

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 (mš) 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)::
        metric_labeled_pair AS m2
26 ), summarize AS (
27     SELECT (m2).label, SUM((m2).value) AS value FROM melt
        GROUP BY (m2).label
28 )

```

```
29 SELECT (9, ARRAY(SELECT (label, value)::metric_labeled_pair FROM
    summarize))::metric_labeled;
30
31 $BODY$;
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 AS
55 $BODY$
56
57 WITH inputs AS (
58     SELECT current_state AS cstate
59 ), melt AS (
60     SELECT unnest((cstate).pairs) AS m2 FROM inputs
61     UNION
62     SELECT (category, (p_area(geom)).value)::
        metric_labeled_pair AS m2
```

```
63 ), summarize AS (  
64     SELECT (m2).label, SUM((m2).value) AS value FROM melt  
        GROUP BY (m2).label  
65 )  
66 SELECT (9, ARRAY(SELECT (label, value)::metric_labeled_pair FROM  
        summarize))::metric_labeled;  
67  
68 $BODY$;  
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)  
    IS 'Calcula la suma de las áreas de los polígonos de la misma  
        categoría dividido por 10.000 para devolver un valor en  
        Hectáreas.';
```

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

```

1  /*
2  Total Area - devuelve el total del área (mš) del paisaje
      dividido por 10.000 (unidades: Hectáreas).
3  */
4  -- SAMPLE USAGE:
5  /*
6  SELECT (l_totalarea(geom)).value FROM sample_patches_25830;
7  SELECT (l_totalarea(geom)).value FROM sample_patches_4326;
8  */
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
      total del paisaje dividida por 10.000 para devolver un valor
      en Hectáreas.';
25
26
27 CREATE OR REPLACE FUNCTION l_totalarea_state(metric,geography)
28     RETURNS metric AS
29 $$
30     SELECT $1 + (p_area($2)).value;
31 $$
32 LANGUAGE 'sql' IMMUTABLE;
33
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. DOCUMENTACIÓN SVG

Entre las 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 [B](#), 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 B.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
4  -- SELECT patches_toSVG(id::integer, geom, category, svg_color,
5  p_area(geom)) FROM sample_patches_4326;
6
7
8  CREATE OR REPLACE FUNCTION patches_toSVG(gid integer, geom
9  geometry, category text, svg_color text, metric metric)
10 RETURNS TEXT AS
11 $func$
12
13 WITH sp(gid,geom,category, svg_color) AS (VALUES
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606

```

```

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

```
