

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

## Presentación

El presente Trabajo Fin de Máster en Tecnologías de la Información Geográfica para la Ordenación del Territorio: SIG y Teledetección, expone todas las tareas que se han realizado durante la colaboración 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 Alfredo Ramón Morte, que cuenta con la participación de los miembros del Laboratorio de Geomática y otros profesionales, en colaboración con el equipo de investigación responsable de la base de datos del SIOSE.

Todo este trabajo se ha realizado mediante el marco de 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, de un total de 440 horas (ver en el ??).

El Laboratorio de Geomática se encarga de administrar el Sistema de Información Geográfica de la Universidad de Alicante (SIGUA), o como era conocido al principio, Laboratorio de SIG y Cartografía Automatizada. Este cambio de nombre fue razón por la orientación del laboratorio al desarrollo de soluciones basadas en la geomática o informática aplicada a la Geografía.

Su origen data de 1997, año en el que además, se pone en marcha el servicio SIGUA, que gran parte de los empleados se centran en el mantenimiento del sistema, como también de la creación de nuevas utilidades adaptadas a las necesidades de la Universidad de Alicante y, compartir recursos e interconectar sistemas de información por otras unidades. Cabe mencionar que, está formado por licenciados en Geografía y en Informática que desarrollan su trabajo en las Tecnologías de la Información Geográfica basado en software libre.

## Proyecto SIOSE-INNOVA

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 marco Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016. Los objetivos principales 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, 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.

A partir de estos objetivos, el proyecto SIOSE-INNOVA tiene como objetivo final crear un visor cartográfico donde se pueda consultar y comparar resultados entre distintos paisajes (ver figura 1). Por este motivo, se desarrolla una nueva extensión denominada *pg\_landmetrics* capaz de calcular métricas del paisaje, papel fundamental para este trabajo.

Las métricas de paisaje son importantes para el estudio del paisaje en su estructura, comportamiento o modificación temporal, ya sea por factores naturales como artificiales. Es por este motivo la necesidad de crear una extensión que permita aplicarlas y calcularlas sobre el paisaje.



Figura 1 Prototipo del visor cartográfico.

## Estructura del trabajo

Este trabajo se organiza en cuatro capítulos, a parte las referencias bibliográficas y los anexos, y su estructura es la siguiente:

- En el capítulo 1 se analiza la estructura del paisaje a partir del uso y cobertura del suelo en relación con las métricas de paisaje. Además, se desarrollan los objetivos que se quieren alcanzar en el trabajo.
- En el capítulo 2 se describen todos los procesos necesarios para implementar y desarrollar la nueva extensión PostgreSQL/PostGIS, desde el conjunto de datos hasta las tareas diarias, la incorporación de funciones y la documentación de la extensión.
- En el capítulo 3 se detallan todos los resultados que se obtienen a partir de la extensión en casos reales.
- Por último, en el capítulo 4 se realiza un análisis conclusivo para Además, se describen las aportaciones interesantes para trabajos futuros.



## Resumen

Las métricas de paisaje se utilizan para analizar la estructura y comportamiento del paisaje como también las modificaciones temporales, ya sea por factores naturales o humanos. Dada la utilidad para una variedad de aplicaciones, existen muchos softwares diseñados para ofrecer cálculos y análisis de patrones de paisaje. El objetivo principal de este trabajo es crear una extensión PostgreSQL/PostGIS reproducible y extensible capaz de calcular métricas de paisaje para datos de entrada vectoriales. Más adelante, esta extensión debería permitir añadir nuevas métricas e investigar nuevos estudios relacionados con la estructura del paisaje. Finalmente, se pone en valor la implementación de la extensión ya que requiere de una metodología de trabajo colaborativo basado en una serie de herramientas de contenerización y orquestación.

**Palabras clave:** métricas de paisaje, extensión, reproducibilidad, contenerización, orquestación.



## **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

<b>Índice de figuras</b>	<b>XV</b>
<b>Índice de tablas</b>	<b>XVII</b>
<b>Ejemplos de código</b>	<b>XVIII</b>
<b>1. Introducción</b>	<b>1</b>
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
<b>2. Metodología</b>	<b>9</b>
2.1. Integración continua y desarrollo colaborativo . . . . .	9
2.1.1. Control de versiones . . . . .	10
2.1.2. Contenerización y orquestación de servicios . . . . .	13
2.1.3. Extensibilidad . . . . .	16
2.1.4. Aplicaciones . . . . .	16
2.2. Conjunto de datos . . . . .	17
2.3. Selección de métricas . . . . .	20
2.4. Implementación/desarrollo de funciones en PostgreSQL . . . . .	20

2.5. Documentación de la extensión . . . . .	27
<b>3. Resultados y Discusión</b>	<b>29</b>
3.1. pg_landmetrics . . . . .	29
3.2. Caso de uso sobre el SIOSE-2011 . . . . .	32
<b>4. Conclusiones y trabajo futuro</b>	<b>33</b>
<b>Bibliografía</b>	<b>35</b>
<b>Apéndice A. Funciones tipo</b>	<b>39</b>
<b>Apéndice B. Documentación SVG</b>	<b>47</b>

# Índice de figuras

1.	Prototipo del visor cartográfico. . . . .	VII
2.1.	Integración continua de la actualización del sistema de ficheros de la extensión <i>pg_landmetrics</i> . . . . .	12
2.2.	Desarrollo colaborativo entre repositorios. . . . .	13
2.3.	Integración continua en la implementación y desarrollo de funciones en PostgreSQL. . . . .	15
2.4.	Interfaz web de la plataforma de PgAdmin. . . . .	17
2.5.	Cobertura del suelo del paisaje de ejemplo. . . . .	18
3.1.	Título. . . . .	30
3.2.	Título. . . . .	30
3.3.	Título. . . . .	30





# Índice de tablas

2.1. Métricas de paisaje disponibles en la extensión. . . . .	21
3.1. Listado de las métricas de paisaje disponibles en la extensión. . . . .	31
3.2. Características de los conjuntos de datos utilizados. . . . .	32



## Ejemplos de código

2.1. Consulta SQL para la métrica AREA. . . . .	22
2.2. Función simple para calcular la métrica AREA por polígonos. . . . .	23
2.3. Ejemplo de uso para calcular la métrica AREA. . . . .	23
2.4. Consulta SQL para la métrica TCA. . . . .	24
2.5. Función de agregado para calcular TCA por categorías. . . . .	24
2.6. Ejemplo de uso para calcular la métrica TCA. . . . .	25
2.7. Extracción de los resultados de las métricas en diferentes escalas de referencia a partir de grids. . . . .	26
A.1. Función para calcular AREA de un polígono de PostGIS de tipo <i>geography</i>	39
A.2. Función para calcular CA por categorías. . . . .	41
A.3. Función para calcular TA del paisaje. . . . .	44
B.1. Función para pintar gráficos vectoriales escalables (SVG). . . . .	47



# 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)<sup>1</sup> 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,

<sup>1</sup><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 del SIOSE *actual*.

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. Así 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 presenta en este capítulo.

### 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 decenas de proyectos ESRI, entre muchos otros).
- La **contenerización o *dockers*** es una novedosa tecnología para la virtualización de software/servicios, frente a la virtualización de sistemas operativos (p.ej. máquinas virtuales). La orquestación de *dockers* permite organizar complejos sistemas de información con muchas facilidades.
- PostgreSQL/PostGIS es la *geodatabase libre* más potente del mercado, destacando por sus opciones de **extensibilidad** (p.ej. PostGIS en sí misma es una extensión de PostgreSQL).

El trabajo colaborativo ha sido uno de los aspectos más fundamentales para la coordinación del desarrollo de la extensión *pg\_landmetrics* en este trabajo donde se ha aplicado un procedimiento de integración continua y desarrollo colaborativo.

En la subsección 2.1.1 se describe el sistema de control de versiones más popular de los últimos años entre usuarios y que se ha aplicado en este trabajo. A continuación, en la subsección 2.1.2 se presentan los servicios de contenerización o *dockers* que se han utilizado para la virtualización del software/servicios necesarios para el desarrollo de la extensión, y la orquestación de éstos. En la subsección 2.1.3 se muestra la geodatabase más potente destacada por su extensibilidad que se ha aplicado para el manejo y gestión de la base de datos del SIOSE. Finalmente, en la subsección 2.1.4 se describen las aplicaciones que se han utilizado para diversos procesos de la metodología.

Es importante destacar que en esta sección se presenta el procedimiento que se ha seguido de forma periódica y diaria en este trabajo desde la integración continua y el desarrollo colaborativo.

### 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.

En este caso, se trabaja de forma colaborativa donde cada usuario modifica desde su máquina local, comparte los cambios y luego el sistema de control combina las modificaciones. En este trabajo se ha utilizado los siguientes sistemas de control de versiones:

- **Git**<sup>1</sup> es un sistema de control para el mantenimiento de versiones de código fuente de archivos. Es el más popular de los últimos años y es utilizado por muchos proyectos como por ejemplo PostGIS, QGIS, CARTO y decenas de proyectos de ESRI, entre otros más. Para la actualización de archivos desde la máquina local se utilizan formas verbales como *pull*, *push*, *add*, *commit*, *clone*, entre otros.
- **GitHub**<sup>2</sup> es una plataforma de desarrollo colaborativo que alberga proyectos y almacena de forma pública el código fuente utilizando el sistema de control de versiones de *Git*.

---

<sup>1</sup><https://git-scm.com/>

<sup>2</sup><https://github.com/>

Todos los archivos que se han generado para la extensión *pg\_landmetrics* se encuentran en el repositorio de este trabajo<sup>3</sup> y en el repositorio proyecto SIOSE-INNOVA<sup>4</sup>.

Uno de los procesos de integración continua y desarrollo colaborativo que se ha realizado es la actualización de ficheros de la extensión *pg\_landmetrics* al repositorio del trabajo (ver la figura 2.1). El procedimiento consistía en que el usuario ejecutaba el comando *pull* desde la máquina local cuando el sistema de ficheros de la extensión no estaba actualizado y éste se encargaba de obtener la última versión. Este paso es opcional si el sistema de ficheros tenía la última versión y no era necesario ejecutar la anterior instrucción. Una vez el usuario recibía un mensaje de notificación de que su sistema de fichero estaba actualizado, se procedía a editar los archivos. Cuando la edición finalizaba, el usuario se encargaba de transmitir todas las modificaciones realizadas a partir de los comandos *add* y *commit*, y a continuación recibía un mensaje de notificación con un *hash* (identificador único) por cada cambio que se había realizado en el proceso. Finalmente, el usuario desde la máquina local subía esta última versión del sistema de ficheros de la extensión al repositorio del trabajo a partir del comando *pull* y así obtener una nueva versión actualizada.

Otro de los procesos de integración continua y desarrollo colaborativo que se ha realizado es la actualización de la extensión entre el repositorio del trabajo y el repositorio *upstream* del proyecto SIOSE-INNOVA (ver la figura 2.2). Una de las primeras tareas era realizar un *fork*, es decir, una copia de todos los ficheros del repositorio *upstream* a un nuevo repositorio de este trabajo. La ventaja es que si en cualquier momento el repositorio del usuario sufría algún contratiempo, se podía volver a realizar un *fork* del repositorio *upstream*. Lo mismo ocurre cuando en nuestra máquina local se tenía el sistema de ficheros de la extensión a partir del comando *clone*. Éste último comando era ejecutado solamente cuando en la máquina local no existía el sistema de ficheros de *pg\_landmetrics*. A continuación se realizaba o no un chequeo de la última versión de la extensión, como se realizaba en el anterior proceso. A partir de aquí, se inicia el trabajo colaborativo entre usuarios y repositorios. En este caso, se vuelve a repetir las mismas acciones de edición y actualización de archivos como en el anterior proceso. Una vez se tenían todos los ficheros de la extensión con una nueva última versión, el usuario enviaba una petición al repositorio *upstream* para emparejar ambos repositorios a través de la función *pull request* que se encuentra en la plataforma de *GitHub*. Si la petición era aceptada por el personal encargado del repositorio *upstream*, se ejecutaba el comando *merge* (unión) y se recibía una notificación de sincronización completada. Finalmente los repositorios quedaban sincronizados y los ficheros con la última versión. Se volvían a subir

<sup>3</sup>[https://github.com/andrearosado/pg\\_landmetrics](https://github.com/andrearosado/pg_landmetrics)

<sup>4</sup>[https://github.com/siose-innova/pg\\_landmetrics](https://github.com/siose-innova/pg_landmetrics)

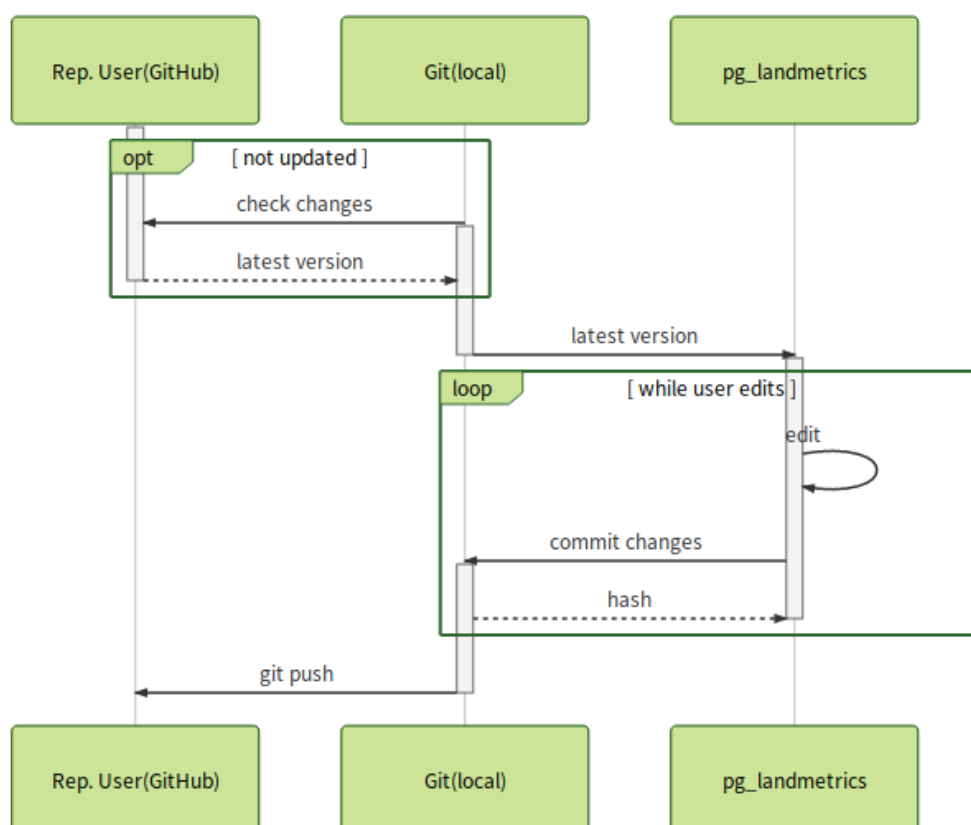


Figura 2.1 Integración continua de la actualización del sistema de ficheros de la extensión *pg\_landmetrics*.



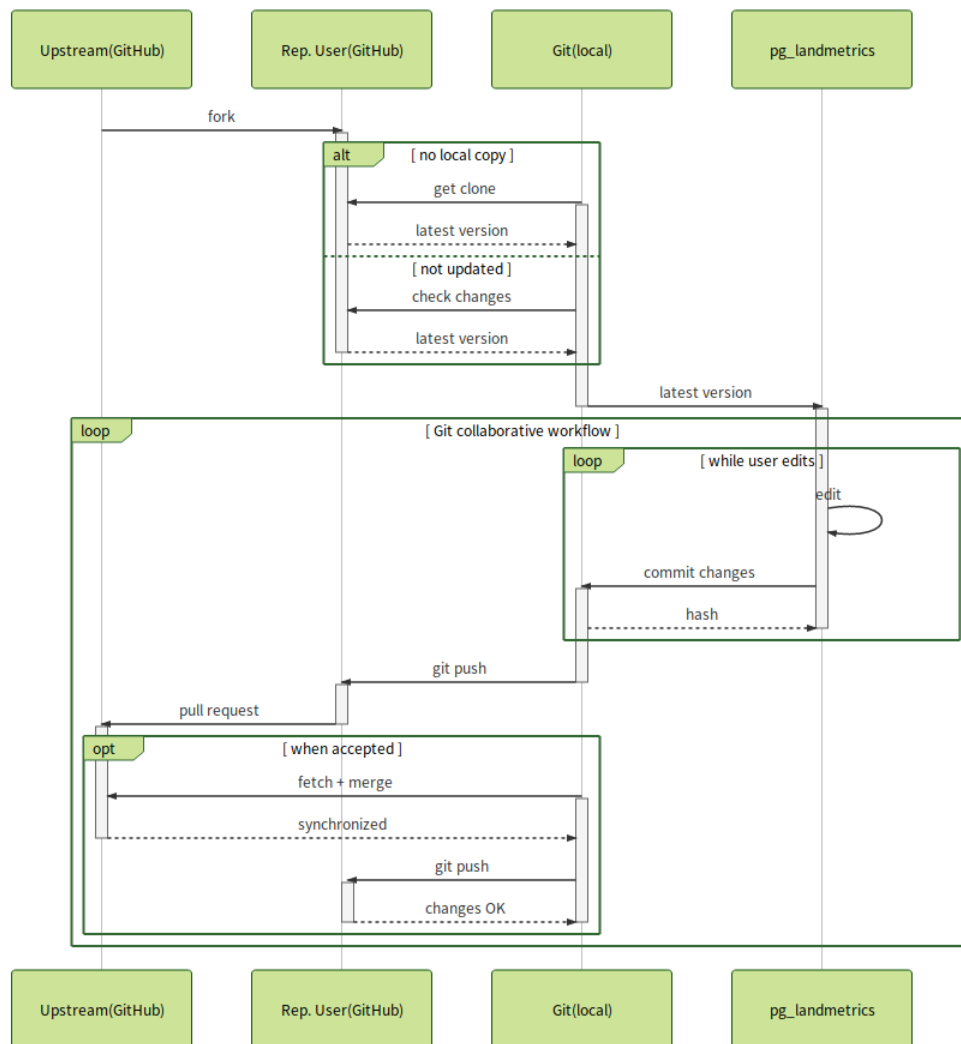


Figura 2.2 Desarrollo colaborativo entre repositorios.

los cambios al repositorio del trabajo y se recibía una notificación de que los cambios habían sido correctos.

### 2.1.2. Contenerización y orquestación de servicios

La contenerización o *docker* es una novedosa tecnología que aporta portabilidad y que virtualiza softwares y servicios frente a la virtualización de sistemas operativos completos como son por ejemplo las máquinas virtuales. Las herramientas de *docker* ofrecen las funciones de construir (*build*), subir o iniciar (*up*) y bajar o detener (*down*) los contenedores

que son como "cajas" donde se almacena lo esencial para ejecutar un determinado software o servicio. En cuanto a la orquestación de contenedores, éste permite organizar complejos sistemas de información con muchas facilidades. En este trabajo se ha utilizado las siguientes herramientas:

- **Docker**<sup>5</sup>: es un proyecto de código abierto que proporciona el despliegue y movilidad de contenedores a través de la nube. Permite autosuficiencia entre las aplicaciones y la infraestructura y los desarrolladores y operaciones para crear un modelo colaborativo e innovador.
- **Docker Hub**<sup>6</sup>: 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 automatización en el proceso de desarrollo.
- **Docker-compose**: es un fichero de configuración que despliega todos los servicios que sean necesarios durante el desarrollo de manera rápida y eficaz.

Finalmente, otro de los procesos de integración continua y desarrollo colaborativo que se ha realizado es el funcionamiento de los *dockers* y extensiones a la hora de implementar y desarrollar las funciones en la extensión *pg\_landmetrics* (ver la figura 2.3). Desde el punto de vista del usuario, se ejecutaba el comando *docker-compose build* si la extensión había sido modificada. En el caso de que no lo estuviera y no fuera necesario volver a construir el *docker-compose*, solamente era necesario ejecutar el comando *docker-compose up* para iniciarlo. A partir de aquí, el *docker-compose* orquestaba y se ocupaba de lanzar dos contenedores: PostgreSQL/PostGIS y PgAdmin. Una vez han sido lanzados dichos contenedores y se han recibido las notificaciones de que la conexión había sido satisfactoria, el usuario realizaba consultas a los contenedores. A continuación, los contenedores realizaban consultas entre ambos y luego se enviaban al usuario los resultados de las consultas acompañado de una vista previa de ellos. Cuando ya eran adquiridos todos los resultados deseados, se ejecutaba el comando *docker-compose down* para que el *docker-compose* dejase de funcionar y el usuario recibiese una notificación por parte de este actor de que había dejado de "funcionar" sin ningún problema.

---

<sup>5</sup><https://www.docker.com/>

<sup>6</sup><https://hub.docker.com/>

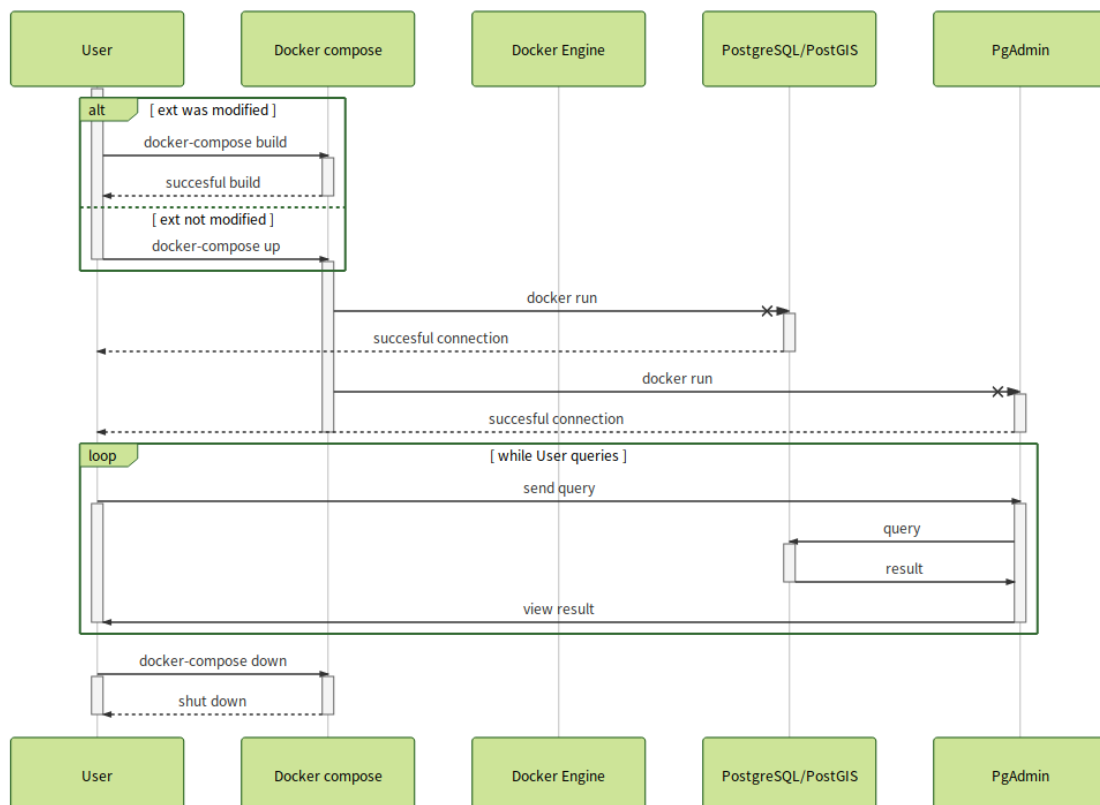


Figura 2.3 Integración continua en la implementación y desarrollo de funciones en PostgreSQL.

### 2.1.3. Extensibilidad

La extensibilidad es la fácil adaptación del software a los nuevos cambios realizados en su especificación. Para ello se utiliza un *makefile* que es un fichero que se encarga de organizar todo el código compilado de todos los softwares necesarios para el desarrollo. Este fichero tiene la capacidad de ampliar las funcionalidades de cualquier aplicación. En este trabajo se ha utilizado la siguiente extensión:

- **PostgreSQL/PostGIS:** es la geodatabase libre más potente del mercado diseñada para ser extensible. PostGIS, como extensión, habilita y agrega soporte a objetos espaciales a la base de datos relacional de PostgreSQL.

### 2.1.4. Aplicaciones

Para la elaboración del conjunto de datos que se han utilizado en este trabajo y las consultas y funciones que se han desarrollado para la extensión *pg\_landmetrics*, se han utilizado las siguientes aplicaciones:

- **PgAdmin4<sup>7</sup>:** es una plataforma que administra, gestiona y desarrolla código abierto en bases de datos PostgreSQL. En la figura 2.4 se muestra el interfaz en el cual se ha trabajado.
- **QGIS 2.18<sup>8</sup>:** 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.

---

<sup>7</sup><https://www.pgadmin.org/>

<sup>8</sup><https://www.qgis.org/es/site/>

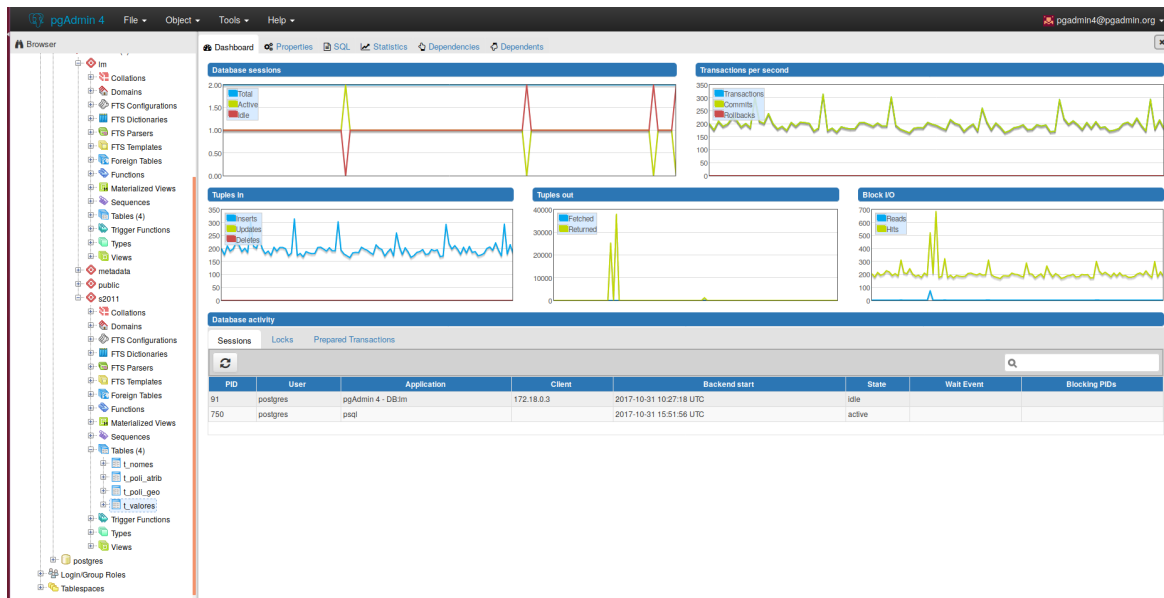


Figura 2.4 Interfaz web de la plataforma de PgAdmin.

## 2.2. Conjunto de datos

### Puntos de interés:

- En este trabajo se han utilizado dos conjuntos de datos, **un paisaje de ejemplo y el SIOSE de 2011 completo**, para poner a prueba la extensión *pg\_landmetrics* propuesta en los objetivos.
- Para demostrar que la **extensión es escalable**, se han utilizado distintas escalas de referencia a partir de *grids*.

Para someter a prueba la extensión *pg\_landmetrics* se han utilizados dos conjuntos de datos: 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 desde PgAdmin como también en QGIS. La escala de referencia es 1:50.000 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 son proyectadas (*geometry*) o son coordenadas esféricas (*geography*).

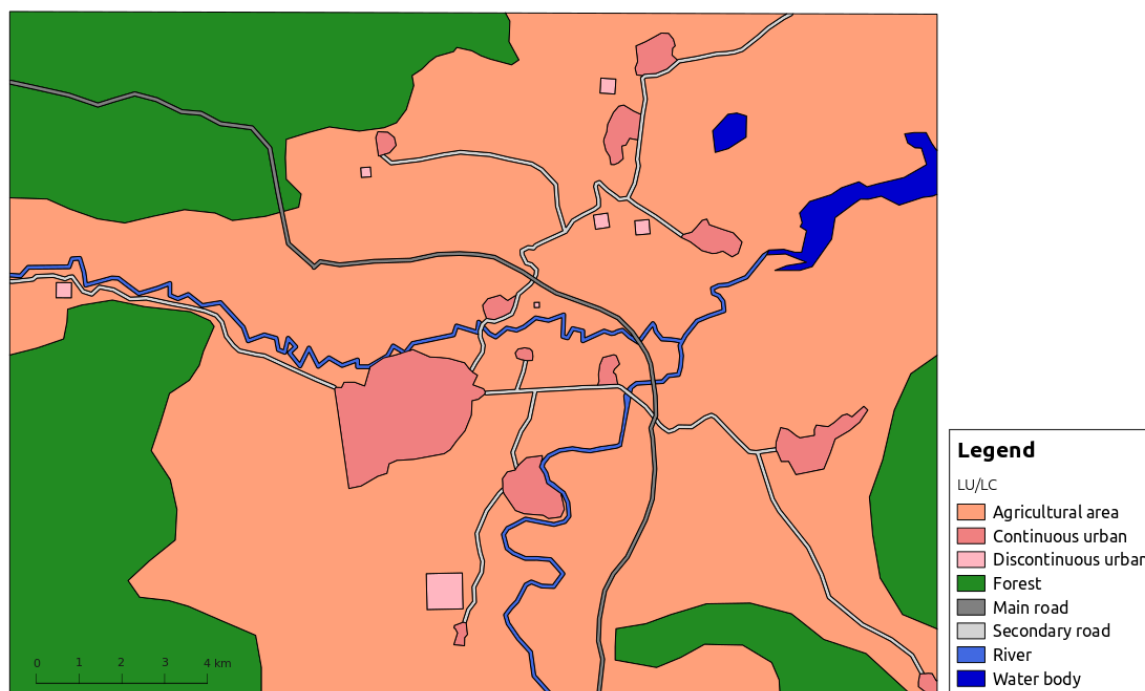


Figura 2.5 Cobertura del suelo del paisaje de ejemplo.

En este conjunto de datos hay un total de 51 polígonos que se han digitalizado a partir de las herramientas de geoprocreso y edición avanzada mediante la aplicación de QGIS. En total hay 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 presenta Scalable Vector Graphics (SVG) Specification<sup>9</sup> además del apoyo de la clasificación de color que especifica el *Corine Land Cover (CLC)*.

Una vez se han comprobado que las métricas de paisaje funcionan correctamente, el segundo conjuntos de datos que se ha utilizado para obtener los resultados es el **SIOSE 2011**, datos que se han obtenido a partir del equipo técnico del SIOSE que colabora en el proyecto SIOSE-INNOVA.

Esta base de datos tan voluminosa se componen de 4 tablas. En la tabla de datos `t_nomes` se encuentran todas las nomenclaturas de cada polígono, en `t_poly_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_poly_geo` ofrece la geometría de cada uno de los

<sup>9</sup><http://www.december.com/html/spec/colorsvg.html>

2.562.800 de polígonos en total y finalmente, en la tabla `t_valores` calculos estadísticos como por ejemplo la superficie en hectáreas.

Como se menciona en la sección 1.1, los polígonos del SIOSE presentan una *cobertura simple* o una *cobertura compuesta*. Dentro de esta última cobertura 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.

Una vez se han reclasificado todos los polígonos del SIOSE 2011, se han obtenido **grids** (cuadrículas cartográficas) para el cálculo de métricas del paisaje en distintas escalas de referencia, a través del Centro de Descargas del Instituto Geográfico Nacional (CNIG)<sup>10</sup>.

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 comprenden en su mayor parte a la provincia de Alicante y a la provincia de Zaragoza, y los alrededores de cada una. El motivo de por qué se han escogido estas dos zonas es por hacer referencia al lugar donde se han realizado las prácticas de empresa y dónde se ha impartido el *máster*.

Dado que ya se han escogido las zonas de estudio, 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 éstas se han realizado todos los resultados que pueden consultarse en el capítulo 3.

---

<sup>10</sup><http://centrodedescargas.cnig.es/CentroDescargas/index.jsp>

## 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 dependen de muchos factores, como por ejemplo los objetivos del estudio, el modelo de datos, la escala, nivel de agregación etc. Por este mismo motivo y según los objetivos especificados en este trabajo y en el proyecto SIOSE-INNOVA, se han seleccionado unas determinadas métricas de paisaje para esta primera propuesta de extensión.

En la tabla 2.1 se aprecian las métricas, acompañadas por su abreviatura, que se han escogido divididas en tres niveles de agregación: a nivel de polígono (*patch*), nivel de categoría (*class*) y nivel de paisaje (*landscape*). Para ello se ha seleccionado un número equitativo entre los niveles de agregación. Además, se han querido escoger algunas métricas que calculen operaciones simples como por ejemplo el área o perímetro, y por otro lado, métricas cuyos cálculos sean más complejos como por ejemplo la distancia del vecino más próximo o la densidad, entre otros.

## 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**.

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



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

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. 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. Por otra parte están los lenguajes procedurales que están incorporados en PostgreSQL para implementar funciones y disparadores como pueden ser PL/pgSQL, PL/R, PL/Python, entre muchos otros.

Dado que se ha trabajado sobre una base de datos relacional y orientada a objetos como es el SIOSE y se ha utilizado tanto PgAdmin 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 por si en un futuro se desconfiguraran y ocasionasen errores. Sin embargo, es cierto que en algunos casos se ha tenido que aplicar PL/pgSQL como lenguaje procedural cuando no era posible elaborar funciones en SQL.

Una vez se ha decidido sobre qué lenguaje de programación se ha utilizado para implementar y desarrollar las funciones, con las métricas de paisaje que se han seleccionado, se ha construido su consulta SQL a partir de las fórmulas matemáticas que acompañan a cada métrica en FRAGSTATS, como referencia. A continuación, se han comprobado estas consultas tanto en QGIS como en PgAdmin a partir del conjunto de datos del paisaje de ejemplo.

Cuando estas últimas comprobaciones han resultado satisfactorias, se ha procedido a desarrollar la función de cada una de las métricas. Cabe mencionar que todas las funciones se han desarrollado para calcular las métricas proyectadas (*geometry*) o en coordenadas esféricas (*geography*) para poder trabajar con conjuntos de datos con cualquier sistema geodésico de referencia. Dicho esto, se han elaborado dos tipos de funciones: simples y de agregado.

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 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 2.2 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)::
           metric_labeled_pair AS m2
17     ), summarize AS (
18         SELECT (m2).label, SUM((m2).value) AS value FROM melt
           GROUP BY (m2).label
19     )
20     SELECT (13, ARRAY(SELECT (label, value)::metric_labeled_pair
           FROM summarize))::metric_labeled;
21
22 $BODY$;
23
24 CREATE AGGREGATE c_totalcorearea(geometry, text)(
25     SFUNC=c_totalcorearea_state,
26     STYPE=metric_labeled,
27     INITCOND='(13,{})'
28 );
29
30 COMMENT ON AGGREGATE c_totalcorearea(geom geometry, category
           text) IS 'Suma las áreas de los núcleos de cada polígono de
           la misma categoría dividido por 10.000 para devolver un valor
           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 $$
11 DECLARE
12     script text;
13
14 BEGIN
15     SELECT DISTINCT string_agg('SELECT * from test.runtests(' ||
16         quote_literal(table_schema || '.' || table_name) || ')', '
17         UNION ALL ') into script
18     FROM information_schema.tables where table_schema = 'test' and
19         table_name like 'grid_%';
20
21     RETURN QUERY EXECUTE script;
22
23 END
24 $$;
```

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**<sup>11</sup> 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**<sup>12</sup> 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)**<sup>13</sup> 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**<sup>14</sup> 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.

<sup>11</sup><https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

<sup>12</sup><https://www.latex-project.org/>

<sup>13</sup>[https://www.w3schools.com/graphics/svg\\_intro.asp](https://www.w3schools.com/graphics/svg_intro.asp)

<sup>14</sup><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 (ref al mockup) 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.

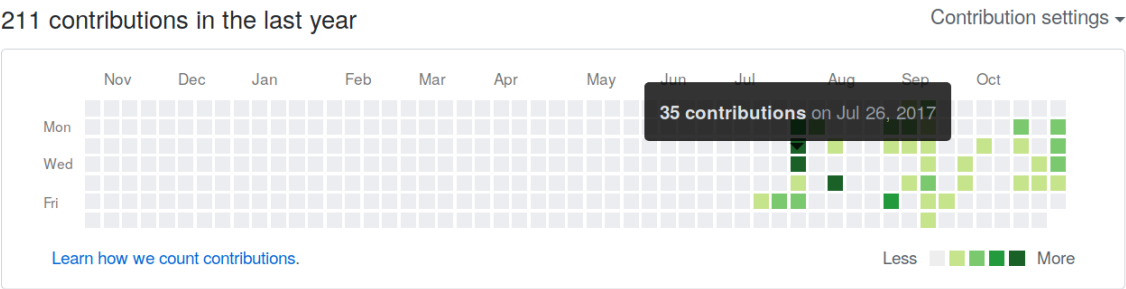


Figura 3.1 Título.

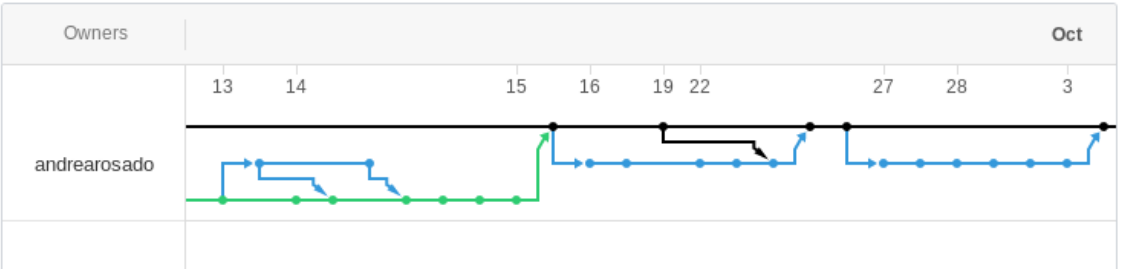


Figura 3.2 Título.

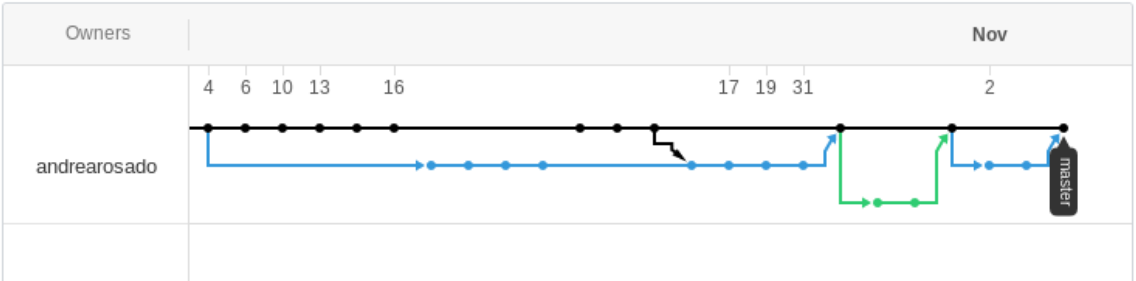


Figura 3.3 Título.

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.
- Baker, W. L. and Cai, Y. (1992). The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology*, 7(4):291–302.
- 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.
- Jacquín, 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.
- Zaragoza, 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.
- Zaragoza, 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.
- Zaragoza, 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)
    IS 'Calcula la suma de las áreas de los polígonos de la misma
        categoría dividido por 10.000 para devolver un valor en
        Hectáreas.';
44
45
46
47 CREATE OR REPLACE FUNCTION c_totalarea_state(
48     current_state metric_labeled,
49     geom geography,
50     category text)
51 RETURNS metric_labeled
52 LANGUAGE 'sql'
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 (gid,geom,category,
16     svg_color)),
17 paths (svg) AS (
18     SELECT array_to_string(
19         array_agg(
20             concat(
```

```

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;

```

---

