

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 [Anexo IV](#)).

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

Índice de figuras	XV
Índice de tablas	XVII
Ejemplos de código	XVIII
1. Introducción	1
1.1. Análisis de la estructura del paisaje a través del SIOSE	1
1.2. Métricas de paisaje y software que las calcula	4
1.3. Objetivos	5
2. Metodología	7
2.1. Integración continua y desarrollo colaborativo	7
2.1.1. Control de versiones	7
2.1.2. Contenerización y orquestación de servicios	7
2.1.3. Extensibilidad	10
2.1.4. Aplicaciones	10
2.2. Conjunto de datos	11
2.3. Selección de métricas	13
2.4. Implementación/desarrollo de funciones en PostgreSQL	15

2.5. Documentación de la extensión	17
3. Resultados y Discusión	19
3.1. pg_landmetrics	19
3.2. Caso de uso sobre el SIOSE-2011	19
4. Conclusiones y trabajo futuro	23
Apéndice A. Anexo I	25
Apéndice B. Anexo II	27
Apéndice C. Anexo III	37
Apéndice D. Anexo IV	39

Índice de figuras

1.	Prototipo del visor cartográfico.	VII
2.1.	Flujo de proceso de actualización de ficheros.	8
2.2.	Flujo de proceso de trabajo colaborativo entre repositorios.	9
2.3.	Flujo de proceso de integración continua.	10
2.4.	Flujo de proceso de integración continua.	11
2.5.	Uso y cobertura del suelo del paisaje de ejemplo.	12
3.1.	Título.	20
3.2.	Título.	20
3.3.	Título.	20

Índice de tablas

2.1. Atributos del primer conjunto de datos.	12
2.2. Color SVG según el tipo de categoría al que corresponde.	12
2.3. Tablas usadas para <i>pg_landmetrics</i>	13
2.4. Listado de métricas de paisaje disponibles en la extensión.	14
2.5. My caption	16
2.6. My caption	16

Ejemplos de código

2.1. Crear una función para calcular el IDW (I)	15
2.2. Crear una función para calcular el IDW (I)	15
2.3. Crear una función para calcular el IDW (I)	15
2.4. Crear una función para calcular el IDW (I)	15
2.5. Crear una función para calcular el IDW (I)	17
2.6. Crear una función para calcular el IDW (I)	17
2.7. Crear una función para calcular el IDW (I)	17
2.8. Crear una función para calcular el IDW (I)	17
B.1. Función para calcular AREA de un polígono.	27
B.2. Función para calcular CA por categorías.	29
B.3. Función para calcular TA del paisaje.	32
B.4. Listado de métricas de paisaje y unidad disponibles en la extensión.	34
C.1. Función para pintar gráficos vectoriales escalables (SVG).	37

1. INTRODUCCIÓN

La interacción de los factores naturales y/o naturales son el origen de la estructura espacial compleja y heterogénea que presenta el paisaje [?] [?]. Desde las últimas décadas, la ecología del paisaje centra sus estudios en la configuración, tamaño y forma que estructuran el territorio a partir de métricas de paisaje [?]. Hoy en día, para conocer la estructura se utilizan bases de datos de ocupación del suelo como por ejemplo el Sistema de Información de Ocupación del Suelo en España (SIOSE). Existen aplicaciones de escritorio que calculan métricas de paisaje pero ninguna de ellas es escalable y extensible para analizar geodatabases con gran volumen y complejidad. Es por ello, que en este trabajo se va a desarrollar una extensión PostgreSQL/PostGIS capaz de calcular métricas a partir de geodatabases tan complejas como es la del SIOSE.

1.1. Análisis de la estructura del paisaje a través del 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

¹<http://www.ign.es/web/ign/portal>

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, 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* [?]. 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 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 [?]. 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 [?]. 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 implica la aplicación por parte de diversos campos de la ciencia, como por ejemplo:

- Medio Ambiente, estudios de hábitat, etc., [?], [?], [?], [?], [?], [?], entre otros.
- Agricultura, por ejemplo [?].

- Demografía, urbanismo y planificación del territorio [?], [?], [?], [?], [?], [?], entre otros.
- Infraestructuras, energía y transporte.
- Riesgos naturales, por ejemplo [?].
- Dinámica de la ocupación del suelo [?], [?], [?], [?], [?], [?], [?], entre otros.

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 usuarios particulares.

El SIOSE presenta dos dificultades de *usabilidad*: 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 adjunto. Este volumen de datos también influye en la velocidad de publicarlos o manejarlos por parte de los usuarios. La complejidad del modelo de datos es mayor ya que se compone de 85 clases, un total de 820.632 casos de coberturas de suelo diferentes (simples y compuestas) [?]. Debido a que la base de datos es un modelo orientado a objetos, hace que haya escasa *usabilidad* por parte de usuarios no expertos. Toda esta información que se almacena para una gran cantidad de geometrías, gestionarla a través de aplicaciones SIG (Sistemas de Información Geográfica) puede llegar a superar la capacidad de éstas, por lo que es necesario estudiar otras tecnologías [?].

El proyecto SIOSE-INNOVA pretende investigar y proponer soluciones para los problemas de *usabilidad* descritos en [?]. 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.

1.2. Métricas de paisaje y software que las calcula

Puntos de interés:

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

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

Hay cientos de métricas de paisaje, correlacionadas entre sí, pero no todas las métricas tendrán significado en todos los contextos y/o estudios. Algunas de las investigaciones que utilizan las métricas de paisaje son aquellas relacionadas con biodiversidad, hábitats, aplicaciones de agua, cambios de suelo, estructura urbana, infraestructura vial, riesgos naturales, estética del paisaje, planificación territorial, entre otros [?].

Por ejemplo, en [?] se investigan aquellas métricas que parecen estar más relacionadas con los estudios forestales para explicar las relaciones entre los procesos ecológicos y los patrones espaciales. Como dice la misma autora, hay cientos de métricas de paisaje que se han desarrollado para cuantificar la heterogeneidad del espacio y las proporciones de uso y cobertura del suelo.

Las métricas de paisaje se pueden calcular a partir de aplicaciones de escritorio. En [?] se establece una comparativa entre programas específicos para el cálculo de métricas del paisaje, destacando entre ellos FRAGSTATS [?], [?]. Por otro lado, hay otros programas como Conefor Sensinode [?], Patch Analyst [?], r.le de GRASS [?], LecoS [?], ZonalMetrics [?], entre otros. Evidentemente, esta lista no puede estar completa.

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 determinadas métricas con herramientas típicas de un SIG como por ejemplo QGIS o ArcGIS.

No obstante, no hay ninguna aplicación escalable y extensible capaz de relaizar análisis sobre bases de datos de ocupación del suelo tan complejas como es la del SIOSE. Por este motivo, se quiere llevar a cabo el objetivo principal de este trabajo (ver [1.3](#)).

1.3. Objetivos

Puntos de interés:

- El principal objetivo de este trabajo es **desarrollar una extensión** (PostgreSQL/PostGIS) capaz de calcular métricas de paisaje a partir de geodatabases tan voluminosas o más como la del SIOSE *actual* (2014).

Enlazando con el objetivo principal de este trabajo de desarrollar una extensión escalable para el cálculo de métricas del paisaje, surgen una serie de objetivos más específicos relacionados con la metodología planteada en el proyecto SIOSE-INNOVA y con conseguir adaptar este trabajo a una dinámica más amplia desarrollada desde hace pocos años en el Laboratorio de Geomática de la Universidad de Alicante. Esta metodología de trabajo es también aplicada en algunas de las principales empresas del sector de Tecnologías de la Información.

Para alcanzar este objetivo, 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 para la implementación de *pg_landmetrics*.

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).

2.1.1. Control de versiones

- **Git:**
- **GitHub:**

2.1.2. Contenerización y orquestación de servicios

Se han utilizado los siguientes elementos:

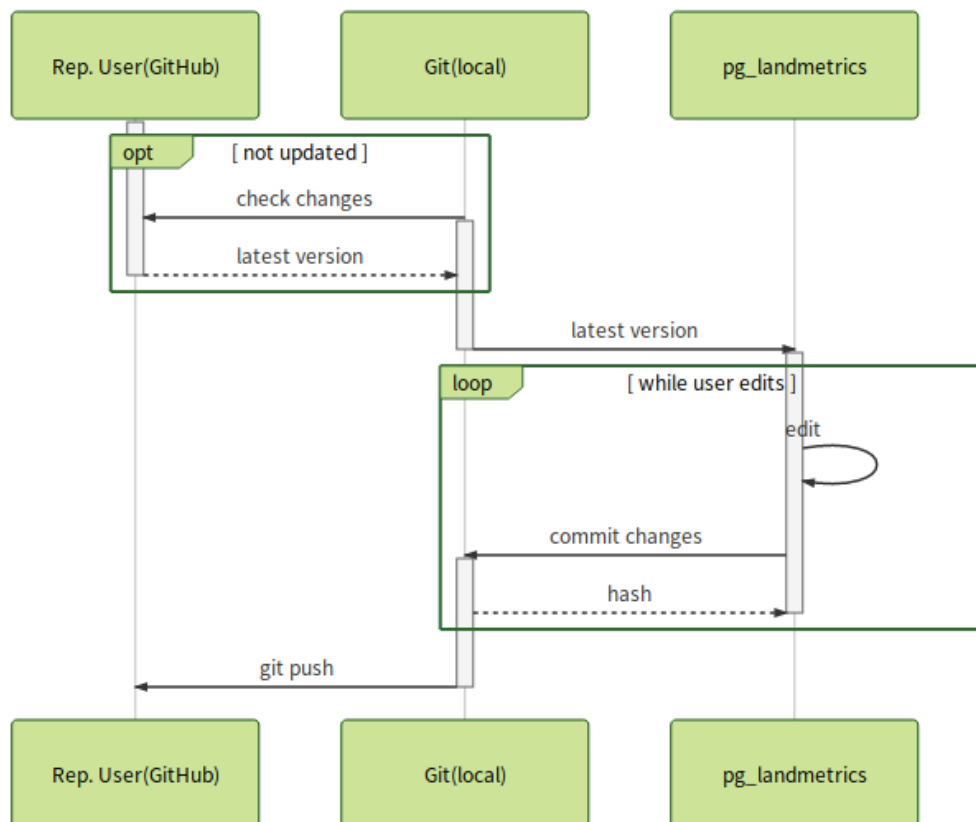


Figura 2.1 Flujo de proceso de actualización de ficheros.

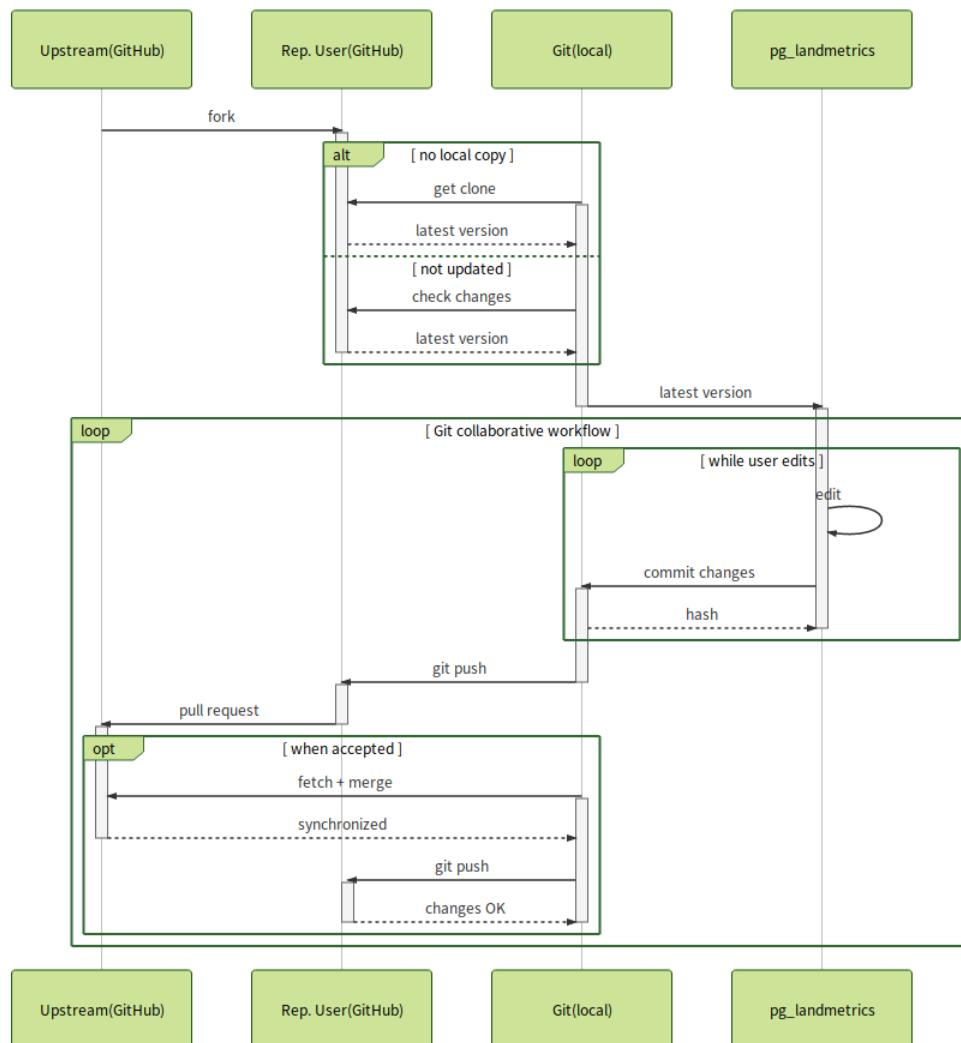


Figura 2.2 Flujo de proceso de trabajo colaborativo entre repositorios.

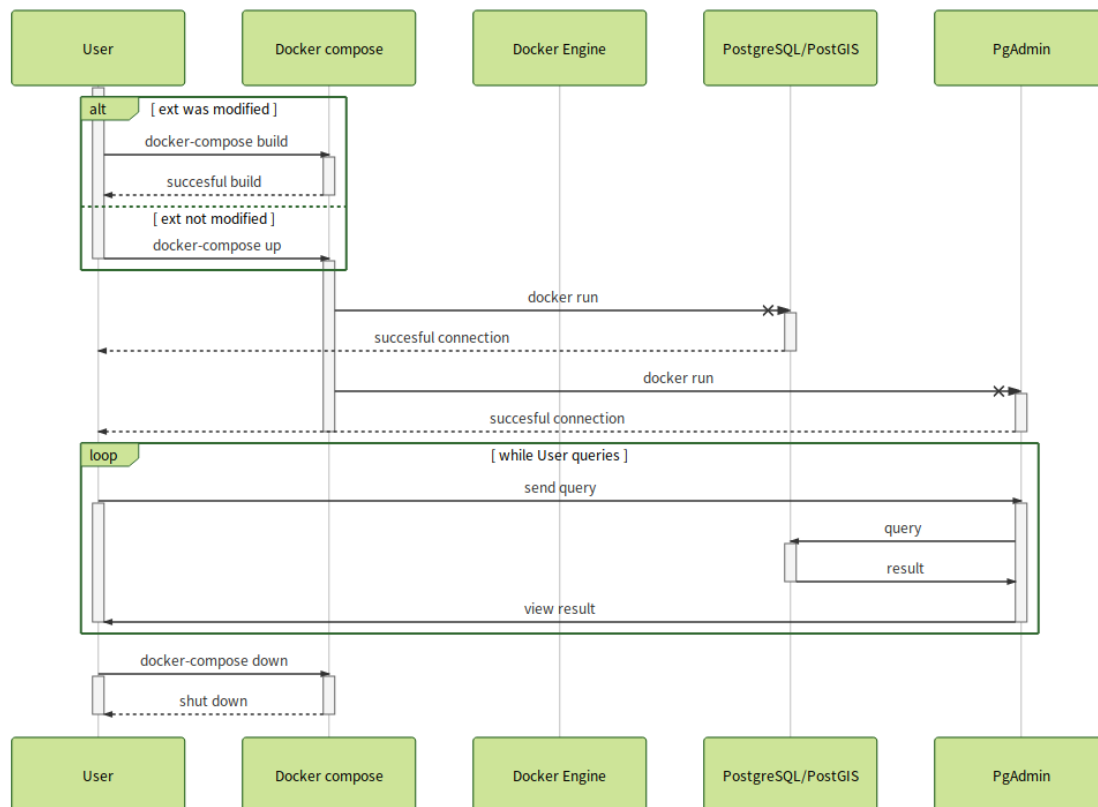


Figura 2.3 Flujo de proceso de integración continua.

- Docker:
- Docker Hub:
- Docker-compose:

2.1.3. Extensibilidad

- PostgreSQL/PostGIS:

2.1.4. Aplicaciones

- PgAdmin4:
- QGIS 2.18:

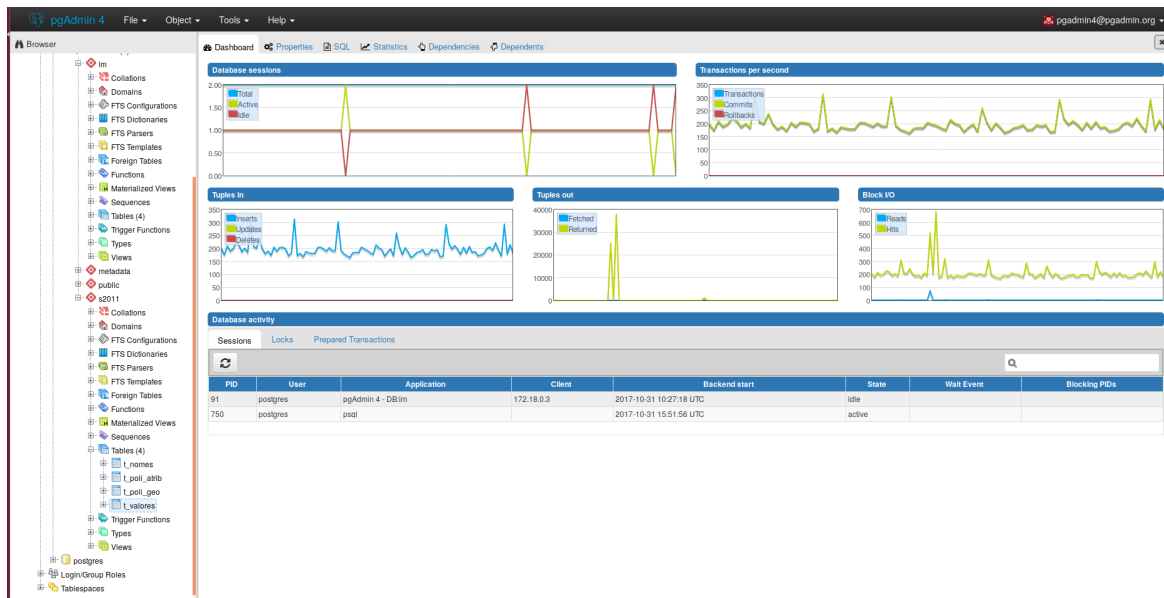


Figura 2.4 Flujo de proceso de integración continua.

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.

Una vez se han escogido las métricas de paisaje adecuadas, se han elaborado dos conjuntos de datos, un paisaje de ejemplo y el SIOSE 2011, para someter a prueba la extensión *pg_landmetrics*.

En el paisaje de ejemplo se han digitalizado todos los polígonos que lo comprenden desde el principio. Se ha querido obtener un paisaje lo menos complejo posible para comprobar de manera más fácil si las métricas de paisaje funcionan correctamente. Para la elaboración de este paisaje se han utilizado las herramientas de geoproceso y edición de la aplicación de escritorio QGIS 2.18.

Este paisaje se ha elaborado con escala de referencia 1:50.000 en dos sistemas geodésicos de referencia: European Terrestrial Reference System 1989 (ETRS89) y World Geodetic System 84 (WGS84). En cuanto a sus características técnicas,

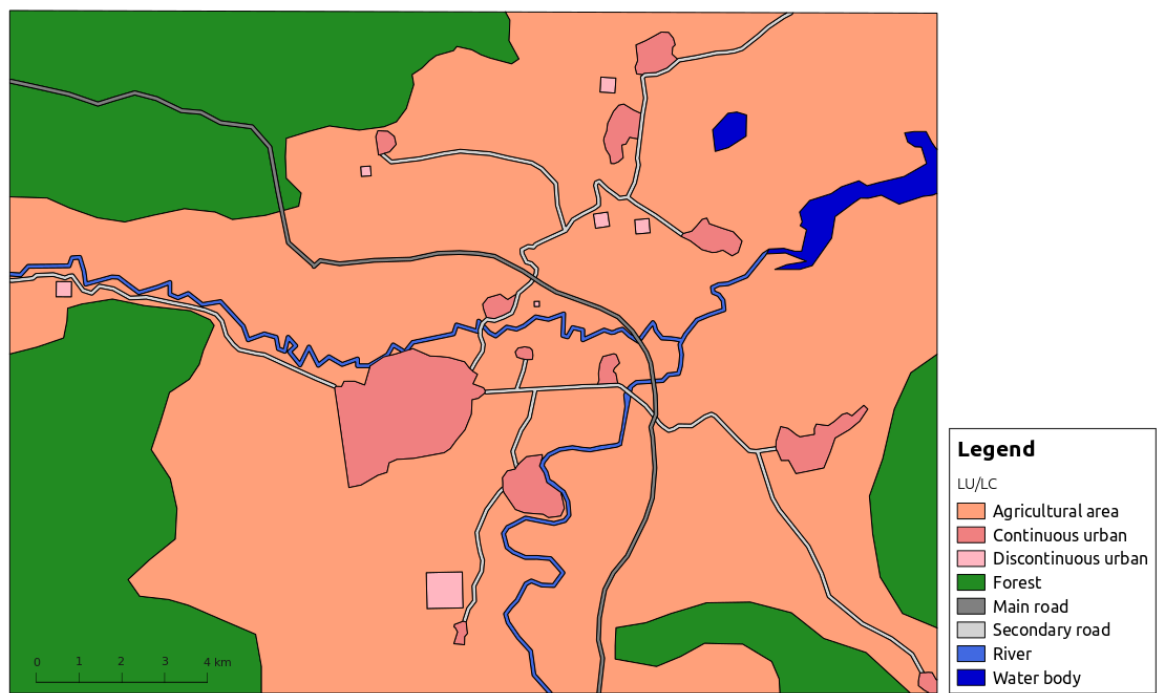


Figura 2.5 Uso y cobertura del suelo del paisaje de ejemplo.

Tabla 2.1 Atributos del primer conjunto de datos.

Nombre	Tipo de campo	Descripción
gid	integer	Identificador único de cada polígono
geom	geometry	Geometría del polígono
category	character varying (text)	Clasificación de la cobertura del suelo
svg_color	character varying (text)	Color según tipo de categoría

Tabla 2.2 Color SVG según el tipo de categoría al que corresponde.

Category	svg_color
Agricultural area	lightsalmon
Continuous urban	lightcoral
Discontinuous urban	lightpink
Forest	forestgreen
Main road	gray
Secondary road	lightgrey
River	royalblue
Water body	mediumblue

Tabla 2.3 Tablas usadas para *pg_landmetrics*.

Tipo	Tablas	Filas	Tamaño total
Relational	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	g25k	756	232,3 kB
	g50k	192	57,8 kB
	g100k	48	13,8 kB
	g500k	2	677bytes

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.

En la tabla 2.4 se aprecian las métricas, acompañadas por su abreviatura, que se han escogido divididas en tres niveles de agregación: a nivel polígono (*patch*), nivel categoría (*class*) y nivel 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.

Tabla 2.4 Listado de 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

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. PLpgSQL, PL/R, PL/Python, entre muchos otros), **dependiendo de las necesidades**.

Ejemplo de código 2.1 Crear una función para calcular el IDW (I)

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

Ejemplo de código 2.2 Crear una función para calcular el IDW (I)

```
1 SELECT SUM(St_Area(geom))/10000, category FROM  
   sample_patches_25830 GROUP BY category;  
2 SELECT SUM(St_Area(geom))/10000, category FROM  
   sample_patches_4326 GROUP BY category;
```

Ejemplo de código 2.3 Crear una función para calcular el IDW (I)

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

Ejemplo de código 2.4 Crear una función para calcular el IDW (I)

```
1 SELECT St_Distance(p1.geom, p2.geom)  
2 FROM sample_patches_25830 AS p1, sample_patches_25830 AS p2  
3 WHERE p1.id = 1 AND p1.id <> p2.id AND p2.category= "category"  
4 ORDER BY St_Distance (p1.geom, p2.geom)  
5 LIMIT 1;  
6  
7 SELECT St_Distance(p1.geom, p2.geom)  
8 FROM sample_patches_4326 AS p1, sample_patches_4326 AS p2  
9 WHERE p1.id = 1 AND p1.id <> p2.id AND p2.category= "category"  
10 ORDER BY St_Distance (p1.geom, p2.geom)
```

Tabla 2.5 My caption

id	level	s_name	l_name	unit_id
1	Patch	AREA	Patch Area	1
2		PERIM	Patch Perimeter	2
3		PARA	Perimeter Area Ratio	3
4		SHAPE	Shape Index	5
5		CORE	Core Area	1
6		NCORE	Number of Core Area	5
7		CAI	Core Area Index	4
8		ENN	Euclidean Nearest Neighbour Distance	2
9	Class	CA	Total (Class) Area	1
10		PLAND	Percentage of Landscape	4
11		TE	Total Edge	2
12		ED	Edge Density	6
13		TCA	Total Core Area	1
14		CPLAND	Core Area Percentage of Landscape	4
15		NP	Number of Patches	5
16		PD	Patch Density	7
17	Landscape	TA	Total Area	1
18		TE	Total Edge	5
19		ED	Edge Density	6
20		NP	Number of Patches	5
21		PD	Patch Density	7
22		PR	Patch Richness	5
23		PRD	Patch Richness Density	7
24		SHDI	Shannon's Diversity Index	5
25		SHIDI	Simpson's Diversity Index	5
26	Proportion	PC	Proportion Class	4
27		PL	Proportion Landscape	4

Tabla 2.6 My caption

id_unit	s_unit	l_unit
1	Ha.	Hectáreas
2	m.	Metros
3	m ²	Metros cuadrados
4	%	Porcentaje
5	-	Ninguno
6	m/Ha	Metros por Hectárea
7	num/100 Ha	Número por 100 Hectáreas

```
11 | LIMIT 1;
```

Ejemplo de código 2.5 Crear una función para calcular el IDW (I)

```
1 | SELECT SUM(St_Area(St_Buffer(geom, -100)))/10000 FROM  
   | sample_patches_25830 GROUP BY category;  
2 | SELECT SUM(St_Area(St_Buffer(geom, -100)))/10000 FROM  
   | sample_patches_4326 GROUP BY category;
```

Ejemplo de código 2.6 Crear una función para calcular el IDW (I)

```
1 | SELECT SUM(St_Perimeter(geom)/St_Area(geom))*10000 FROM  
   | sample_patches_25830;  
2 | SELECT SUM(St_Perimeter(geom)/St_Area(geom))*10000 FROM  
   | sample_patches_4326;
```

Ejemplo de código 2.7 Crear una función para calcular el IDW (I)

```
1 | SELECT (p_corearea(geom, 50)).value FROM sample_patches_25830;  
2 | SELECT (p_corearea(geom, 50)).value FROM sample_patches_4326;
```

Ejemplo de código 2.8 Crear una función para calcular el IDW (I)

```
1 | SELECT c_totalarea(geom,category) FROM sample_patches_25830;  
2 | SELECT c_totalarea(geom,category) FROM sample_patches_4326;
```

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, sobretodo a aquellos menos expertos en la materia. Así pues, se han aplicado los siguientes lenguajes de marcado:

- **Markdown**¹ es un lenguaje ligero que permite una escritura sencilla y de fácil lectura usando texto plano. Se ha utilizado para documentar el usp de la extensión.
- **TeX**² 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)**³ 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 III](#)).
- **Mermaid**⁴ 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.

¹<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

²<https://www.latex-project.org/>

³https://www.w3schools.com/graphics/svg_intro.asp

⁴<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, 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.

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.

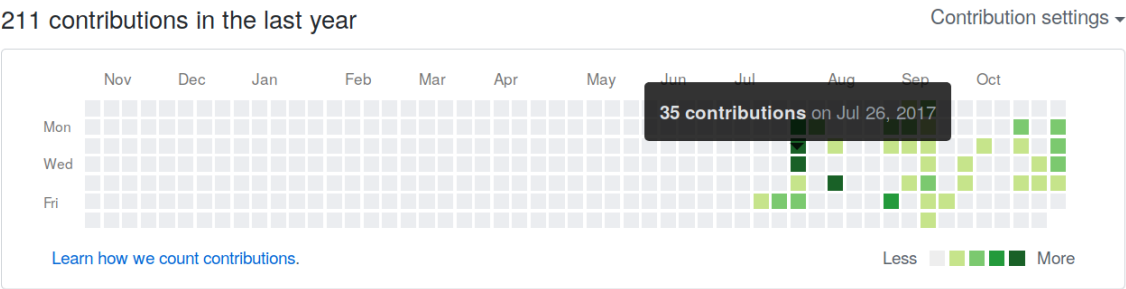


Figura 3.1 Título.

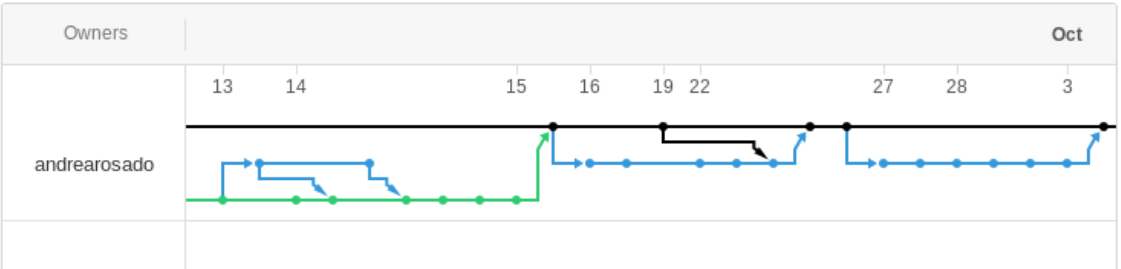


Figura 3.2 Título.

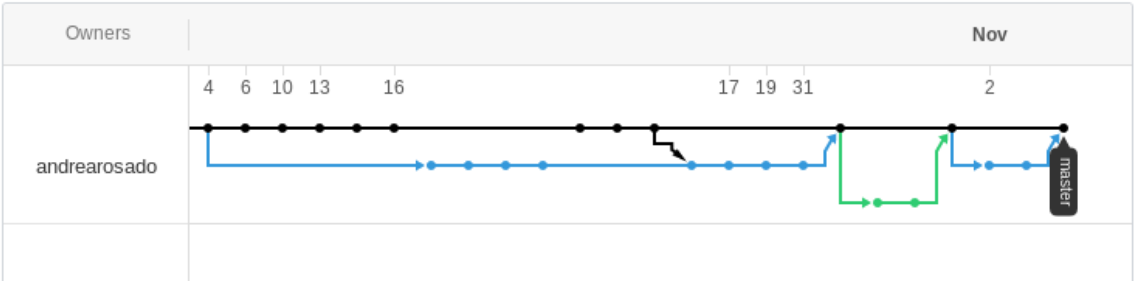


Figura 3.3 Título.

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:

- What?
- How?
- When?
- Who?
- Where?
- why or for what?

Conclusiones

Trabajo

A. ANEXO I

Métricas de paisaje disponibles en la extensión

B. ANEXO II

Funciones en PostgreSQL

Ejemplo de código B.1 Función para calcular AREA de un polígono.

```
1  /*
2  Patch Area - devuelve la suma del área del polígono dividido por
3              10.000 (unidades: Hectáreas).
4  */
5  -- SAMPLE USAGE:
6  /*
7  SELECT (p_area(geom)).value As p_area FROM sample_patches_25830;
8  SELECT (p_area(geom)).value As p_area FROM sample_patches_4326;
9  */
10 CREATE OR REPLACE FUNCTION p_area(geom geometry)
11 RETURNS metric AS
12 $$
13
14 SELECT (1, St_Area(geom)/10000)::metric;
15
16 $$
17 LANGUAGE SQL
18 IMMUTABLE
19 RETURNS NULL ON NULL INPUT;
20
21 COMMENT ON FUNCTION p_area(geom geometry) IS 'Divide el área en
22         metros cuadrados de un polígono por 10.000 para devolver un
23         valor en Hectáreas.';
24
25 CREATE OR REPLACE FUNCTION p_area(geom geography)
26 RETURNS metric AS
27 $$
```

```
28 SELECT (1, St_Area(geom)/10000)::metric;
29
30 $$
31 LANGUAGE SQL
32 IMMUTABLE
33 RETURNS NULL ON NULL INPUT;
34
35 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 B.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 B.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.';
```

Ejemplo de código B.4 Listado de métricas de paisaje y unidad disponibles en la extensión.

```

1 CREATE TABLE metric_meta_unit (
2     id serial primary key,
3     sunit text NOT NULL,
4     lunit text NOT NULL
5 );
6
7 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('Ha.', '
8     Hectáreas');
9 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('m.', 'Metros'
10 );
11 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('mš.', 'Metros
12 ');
13 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('%', '
14     Porcentaje');
15 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('', 'None');
16 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('m/Ha.', '
17     Metros por Hectárea');
18 INSERT INTO metric_meta_unit(sunit,lunit) VALUES ('num/100 Ha.',
19     'Número por 100 Hectáreas');
20
21
22
23 CREATE TABLE metric_meta (
24     id serial primary key,
25     sname text NOT NULL,
26     lname text NOT NULL,
27     unit_id integer NOT NULL
28 );
29
30 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('AREA', '
31     Patch Area',1);
32 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PERIM', '
33     Patch Perimeter',2);
34 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PARA', '
35     Perimeter Area Ratio',3);
36 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('SHAPE', '
37     Shape Index',5);

```

```

27| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('CORE', '
    Core Area',1);
28| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('NCORE', '
    Number of Core Areas',5);
29| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('CAI', '
    Core Area Index',4);
30| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('ENN', '
    Euclidean Nearest Neighbour Distance',2);
31| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('CA', '
    Total (Class) Area',1);
32| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PLAND', '
    Percentage of Landscape',4);
33| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('TE', '
    Total Edge',2);
34| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('ED', 'Edge
    Density',6);
35| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('TCA', '
    Total Core Area',1);
36| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('CPLAND', '
    Core Area Percentage of Landscape',4);
37| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('NP', '
    Number of Patches',5);
38| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PD', '
    Patch Density',7);
39| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('TA','Total
    Area',1);
40| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('TE','Total
    Edge',5);
41| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('ED','Edge
    Density',6);
42| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('NP','
    Number of Patches',5);
43| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PD','Patch
    Density',7);
44| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PR','Patch
    Richness',5);
45| INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PRD','
    Patch Richness Density',7);

```

```
46 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('SHDI','
    Shannon Diversity Index',5);
47 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('SHIDI','
    Simpson Diversity Index',5);
48 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PC','
    Proportion Class',4);
49 INSERT INTO metric_meta(sname,lname,unit_id) VALUES ('PL','
    Proportion Landscape',4);
50
51
52
53 -- id: metric_meta id
54 CREATE TYPE metric AS (
55     id integer,
56     value numeric
57 );
58
59
60 CREATE TYPE metric_labeled_pair AS
61 (
62     label text,
63     value numeric
64 );
65
66 -- id: metric_meta id
67 CREATE TYPE metric_labeled AS
68 (
69     id integer,
70     pairs metric_labeled_pair[]
71 );
```


C. ANEXO III

Patch to SVG

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

```

1  -- SELECT patches_toSVG(id::integer, geom, category, svg_color,
    p_area(geom)) FROM sample_patches_25830;
2  -- SELECT patches_toSVG(id::integer, geom, category, svg_color,
    p_area(geom)) FROM sample_patches_4326;
3
4  CREATE OR REPLACE FUNCTION patches_toSVG(gid integer, geom
    geometry, category text, svg_color text, metric metric)
5  RETURNS TEXT AS
6  $func$
7
8  WITH sp(gid,geom,category, svg_color) AS (VALUES
9
    (gid,geom,category,
        svg_color)),
10 paths (svg) AS (
11     SELECT array_to_string(
12         array_agg(
13             concat(
14                 '<path d= "', ST_AsSVG(geom,0), '" ', ' stroke='
15                 'black" stroke-width="2" fill=" ', svg_color, '
16                 " />'
17             ),','))
18     FROM sp
19 ),
20 texts (svg) AS(
21     SELECT array_to_string(
22         array_agg(
23             concat(
24                 '<text x=" ',st_x(st_pointonsurface(geom)),'" y="
25                 ',-st_y(st_pointonsurface(geom)),'" font-size

```

```

                                ="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;

```

D. ANEXO IV

Cronograma

