



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS *VERSUS*
ARCHIVOS *RASTER* EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)**

Mario Enrique Cojón Tomas

Asesorado por el Ing. Byron Emmanuel Bobadilla Chin

Guatemala, marzo de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS *VERSUS*
ARCHIVOS *RASTER* EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

MARIO ENRIQUE COJÓN TOMAS

ASESORADO POR EL ING. BYRON EMMANUEL BOBADILLA CHIN

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MARZO 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

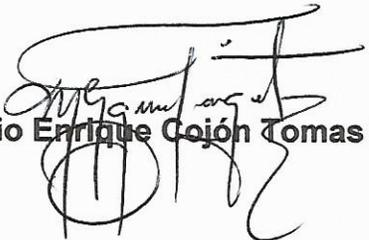
DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramirez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS *VERSUS* ARCHIVOS *RASTER* EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 5 de marzo de 2013.


Mario Enrique Cojón Tomás

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS

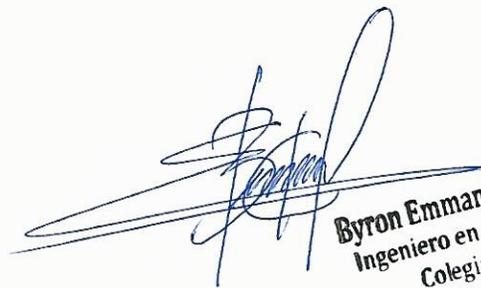
Guatemala, 30 de Septiembre de 2013

Ing. Carlos Azurdia:

Por este medio le informo que he realizado la revisión y evaluación del trabajo de graduación del estudiante Mario Enrique Cojón Tomas que se identifica con número de carne: 2000-10819 cuyo tema es “Estudio de factibilidad usando base de datos vrs archivos raster en Sistemas de Información Geográfica SIG” y estoy conforme con el trabajo realizado.

Agradeciendo la atención a la presente y quedando a sus órdenes para cualquier información adicional.

Atentamente,



Byron Emmanuel Bobadilla Chin
Ingeniero en Ciencias y Sistemas
Colegiado No. 10,977

Ing. Byron Emmanuel Bobadilla Chin
Ingeniero en Ciencias y Sistemas
Colegiado No. 10977
Tel: 4136-8156
Email: bynuel685@gmail.com



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 9 de Octubre de 2013

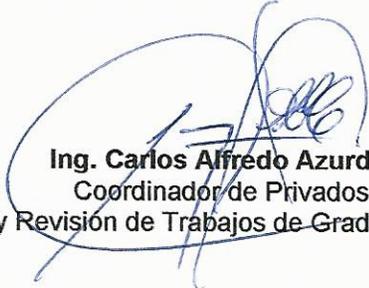
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **MARIO ENRIQUE COJÓN TOMAS** carné 2000-10819, titulado: "ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS VRS ARCHIVOS RASTER EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



Universidad de San
Carlos de Guatemala



Facultad de Ingeniería
Unidad de Lingüística

Guatemala, 17 de marzo de 2015
Ling.7/15

Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería, USAC

Señor director:

Por este medio hago de su conocimiento que la Unidad de Lingüística hace una modificación al título del trabajo de graduación del estudiante **Mario Enrique Cojón Tomas**, con número de carné: **2000-10819** el cual fue aprobado de acuerdo al protocolo como: **ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS VRS ARCHIVOS RASTER EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG.**

La Unidad modifica el título del trabajo en virtud de que el mismo no está bien redactado y propone la siguiente forma: **ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS VERSUS ARCHIVOS RASTER EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG).**



Licenciada Rosa Amelia González Domínguez
Coordinadora de la Unidad de Lingüística

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

E
S
C
U
E
L
A

D El Director de la Escuela de Ingeniería en Ciencias y
E Sistemas de la Facultad de Ingeniería de la Universidad de
C San Carlos de Guatemala, luego de conocer el dictamen del
I asesor con el visto bueno del revisor y del Licenciado en
E Letras, del trabajo de graduación **“ESTUDIO DE
N FACTIBILIDAD USANDO BASE DE DATOS VERSUS
C ARCHIVOS RASTER EN SISTEMAS DE
I INFORMACIÓN GEOGRÁFICA (SIG).”**, realizado por el
A estudiante MARIO ENRIQUE COJÓN TOMAS, aprueba el
S presente trabajo y solicita la autorización del mismo.

Y

“ID Y ENSEÑAD A TODOS”

S
I
S
T
E
M
A
S



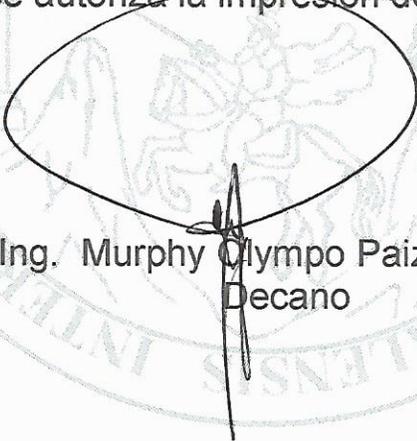
Ing. Marlon Antonio Pérez Türk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 18 de marzo 2015



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ESTUDIO DE FACTIBILIDAD USANDO BASE DE DATOS VERSUS ARCHIVOS RASTER EN SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)**, presentado por el estudiante universitario: **Mario Enrique Cojón Tomas**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.



Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, marzo de 2015



/cc

ACTO QUE DEDICO A:

- Dios** Por guiar mi camino, dándome fuerzas y sabiduría, para alcanzar una meta más en mi vida.
- Mis padres** Todo mi amor y admiración. Por su apoyo y toda la confianza depositada en mí para realizar esta meta. Para ustedes esto es uno de los frutos de la semilla que han sembrado en mí.
- Mi esposa** Miriam Griselda López, todo mi amor y admiración. Siempre estuviste conmigo, gracias por el apoyo en la realización de esta meta en mi vida.
- Mi hijo** Kenneth Enrique Cojón López, espero ser para ti un ejemplo digno de seguir.
- Mis hermanas** Claudia Hortensia, Gloria Marina y Clementina Dolores Cojón Tomas, con todo mi amor, ya que formaron parte de esta meta con todo su apoyo.

AGRADECIMIENTOS A:

**Universidad de
San Carlos de Guatemala**

Por ser una importante influencia en mi carrera, entre otras cosas.

Facultad de Ingeniería

Por ser una importante influencia en mi carrera, entre otras cosas.

**Mis amigos de la
Facultad**

Juan Carlos Capriel, Walter Pérez, Amner Cabrejo, Mynor Xinico, Edgar Flores.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XVII
OBJETIVOS.....	XIX
INTRODUCCIÓN	XXI
1. MARCO TEÓRICO.....	1
1.1. Sistema de Información Geográfica	1
1.1.1. Base de datos espacial.....	3
1.1.2. Datos espaciales	4
1.2. PostgreSQL.....	5
1.2.1. Características.....	5
1.2.2. Funciones	7
1.3. PostGIS	8
1.4. Python	9
1.5. GDAL.....	10
1.6. Enhanced Compressed Wavelet ECW	10
2. CONFIGURACIÓN DE INFRAESTRUCTURA.....	11
2.1. Instalación de software	11
2.1.1. Instalación de Java	11
2.1.2. Instalación de PostgreSQL	14
2.1.3. Instalación de FWTools	21
2.1.4. Instalación de Geoserver	22

3.	PROCESO DE CARGA DE ORTOFOTOS A BASE DE DATOS.....	27
3.1.	Unificación de ortofotos (mosaico de imágenes).....	28
3.2.	Generar pirámide de capas.....	30
3.3.	Configuración de mapa	32
3.3.1.	Generar <i>script</i> para crear tablas PostGIS.....	34
3.3.2.	Generando las tablas en la base de datos	35
3.4.	Importar capas a base de datos PostgreSql	38
4.	ESTUDIO DE FACTIBILIDAD.....	39
4.1.	Configurar capa en Geoserver con información almacenada en base de datos.....	39
4.1.1.	Publicar capa <i>raster</i>	41
4.1.2.	Previsualización de capa.....	45
4.2.	Configurar capa en Geoserver con ortofotos en formato TIF ...	47
4.3.	Configurar capa en Geoserver con archivos ECW.....	57
5.	RESULTADO DE ESTUDIO	61
5.1.	Espacio necesario en disco duro por escenario	61
5.1.1.	Base de datos PostGIS	61
5.1.2.	Ortofotos TIF	62
5.1.3.	Archivos ECW	63
5.2.	Seguridad en cada escenario.....	65
5.3.	Performance en cada escenario	66
5.4.	Costo de configuración por cada escenario	70
5.5.	Media de resultados.....	71

CONCLUSIONES	73
RECOMENDACIONES	75
BIBLIOGRAFÍA.....	77
ANEXOS.....	79

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Instalación de Java.....	12
2.	Activar propiedades de PC.....	13
3.	Configurando variable JAVA_HOME y PATH	13
4.	Información de java	14
5.	Asistente de Instalación de PostgreSQL	15
6.	Configuración de instalación de PostgreSQL.....	16
7.	Selección de extensión espacial PostGIS	17
8.	Instalación de PostGIS.....	18
9.	Generando GDAL_DATA como variable de entorno.....	19
10.	Página web de referencia espacial GTM para PostGIS	20
11.	Instalación de FWTools.....	21
12.	Instalación de aplicación Geoserver.....	23
13.	Página web de referencia espacial GTM para Geoserver.....	24
14.	Archivo de configuración epsg.properties modificado	25
15.	Ilustración de ortofotos en carpeta Cuadrícula	27
16.	Opción combinar del menú <i>raster</i>	28
17.	Formulario para combinar ortofotos en QuantumGIS.....	29
18.	Documento generado con la combinación de ortofotos	30
19.	Comando gdal_retile en ejecución	31
20.	Capas Generadas de mosaico de ortofotos (nivel 0)	31
21.	Xml con configuración de conexión a base de datos	32
22.	Xml con Información de asignación para tablas.....	33
23.	Xml con configuración de mapa	33

24.	Generar archivos de configuración para crear tablas en base de datos.....	34
25.	Archivos sql generados.....	35
26.	Instrucción para agregar tabla de capas en base de datos	36
27.	Instrucción para crear tablas de nivel de pirámide en base de datos ...	36
28.	Tablas generadas en la base de datos	37
29.	Registro con referencia espacial GTM para PostGIS	40
30.	Archivo epsg.properties con referencia espacial GTM para Geoserver	40
31.	Interfaz origen de datos Geoserver.....	41
32.	Interfaz de origen de datos ImageMosaicJDBC.....	42
33.	Capa <i>raster</i> con información PostGIS.....	43
34.	Editar capa <i>raster</i> PostGIS (1).....	43
35.	Editar capa <i>raster</i> PostGIS (2).....	44
36.	Capa <i>raster</i> PostGIS para publicación.....	45
37.	Publicando capa <i>raster</i> PostGIS	46
38.	Capa <i>raster</i> visualizado en OpenLayers	47
39.	Configuración de capa <i>raster</i> TIF.....	48
40.	Formulario origen de datos Imagemosaic para formato TIF	49
41.	Guardar configuración capa <i>raster</i> TIF	49
42.	Archivos de configuración generados para capa TIF	50
43.	Referencia espacial GTM para capa <i>raster</i> TIF	51
44.	Sitio web spatial reference con listado de opciones.....	51
45.	Archivo de configuración capa <i>raster</i> TIF modificado	52
46.	Listado de capas configuradas en Geoserver.....	53
47.	Opción publicar capa <i>raster</i> TIF.....	53
48.	Configuración de capa <i>raster</i> TIF (1)	54
49.	Configuración de capa <i>raster</i> TIF (2)	55
50.	Configuración de capa <i>raster</i> TIF (3)	55

51.	Previsualizar capa <i>raster</i> TIF	56
52.	Visualizar capa <i>raster</i> TIF con OpenLayers	57
53.	Agregando origen de datos ECW	58
54.	Parámetros de configuración para capa <i>raster</i> ECW	59
55.	Opción previsualizar capa <i>raster</i> ECW.....	60
56.	Espacio ocupado por base de datos (calculado en PostgreSQL)	62
57.	Espacio requerido por archivos con formato TIF	63
58.	Espacio requerido por archivos con formato ECW.....	64
59.	Gráfica de resultados (espacio requerido en GB)	65
60.	Nivel de seguridad por escenario (porcentaje).....	66
61.	Página web de tiempos (milisegundos).....	67
62.	Promedio de tiempos (ms) por escenario.....	69
63.	Nivel de performance por escenario.....	69
64.	Comparación de costos por escenario	70
65.	Media de resultados	71

TABLAS

I.	Tiempos de carga en milisegundos por escenario	68
----	--	----

LISTA DE SÍMBOLOS

Símbolo	Significado
CMD	Command.
CSV	Comma-Separated Values.
XML	Extensible Markup Language.
srtext	Formato de archivo informático para imágenes.
GTM	Guatemala Transversa Mercator.
IP	Internet Protocol.
IPv4	Internet Protocol version 4 (en español: Protocolo de Internet versión 4).
SRID	Spatial Reference System.
WKT	Well-Known Text.

GLOSARIO

Base de datos	Almacén de datos relacionados con diferentes modos de organización, permitiendo manipularlos fácilmente y mostrarlos de diversas formas.
Código fuente	Texto escrito en un lenguaje de programación específico y que puede ser leído.
Configuración	En informática la configuración es un conjunto de datos que determina el valor de algunas variables de un programa o de un sistema operativo. Estas opciones generalmente son cargadas en su inicio.
Contraseña	Una contraseña o clave (<i>password</i>) es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso.
Copias de seguridad	Copia total o parcial de información importante del disco duro, CDs, bases de datos u otro medio de almacenamiento.
ECW	Enhanced Compression Wavelet (ECW) Formato de archivo propietario para imágenes <i>raster</i> desarrollado por la empresa Earth Resource Mapping.

Geografía	Ciencia que trata de la descripción o de la representación gráfica de la Tierra.
Hardware	Conjunto de componentes físicos tecnológicos, que trabajan o interactúa de algún modo con la computadora.
Información	Conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.
Instalación	Proceso por el cual, un programa se encarga de copiar todos los ficheros de una aplicación desde un disquete o un CD-ROM al disco duro. Además, durante la instalación se efectúan todas las configuraciones necesarias que también se guardan en el disco duro.
Internet	Red que conecta redes y computadoras distribuidas por todo el mundo, permite la comunicación y transferir información sin grandes requerimientos tecnológicos ni económicos relativos para el individuo.
Java script	Lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web.

Mapa	Representación gráfica y métrica de una porción de territorio generalmente sobre una superficie bidimensional.
Menú	En informática un menú es una serie de opciones que el usuario puede elegir para realizar determinada tarea.
Mosaico de imágenes	En fotografía, un mosaico de imágenes consiste en un montaje de varias imágenes que forman entre todas una sola imagen.
Navegador web	Programas como firefox, internet explorer, google chrome, etc. que permiten ver las páginas web.
Página web	Es parte de un sitio web y es un único archivo con un nombre de archivo asignado.
Pirámide de capas	Se utilizan para mejorar el rendimiento, son una versión con submuestreo del <i>dataset raster</i> original y pueden contener varias capas con submuestreo.
Raster	En sistemas de información geográfica, <i>raster</i> es imagen de mapa de bits o matricial.
Script	En informática, es un guión, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

Seguridad	En informática, se refiere a las características y condiciones de sistemas de procesamiento de datos y su almacenamiento, para garantizar su confidencialidad, integridad y disponibilidad.
Servidor	En redes, es computadora central en un sistema de red que provee servicios a otras computadoras. En internet, los servidores son los proveedores de todos sus servicios, incluyendo la www (las páginas web), el FTP, el correo electrónico, los grupos de noticias, etc.
SIG	Sistema de Información Geográfica.
Sistema	En informática, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información.
Sitio web	Es un sitio (localización) en la World Wide Web que contiene documentos (páginas web) organizados jerárquicamente.
Software	Programa o aplicación, programado para realizar tareas específicas.
SQL	Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

TIF

Formato de archivo de gráficos de mapa de bits.

URL

En informática, URL (Uniform Resource Locator) especifica el lugar donde se identificó un recurso disponible y el mecanismo para la recuperación de ella.

RESUMEN

El presente trabajo de graduación muestra un estudio de factibilidad con uso de información geográfica en tres escenarios de información *raster* (información espacial): almacenada en base de datos, en archivos TIF y en archivos ECW.

Describe los pasos a realizar para cargar información *raster* de archivos a base de datos PostgreSQL, asimismo el proceso para configurar las capas espaciales en los tres escenarios de prueba.

La configuración de publicación de la información espacial, se realiza en la aplicación Geoserver como servidor de mapas, y la presentación de las capas en *openlayers*.

Por último se realiza un estudio, comparando el performance, seguridad, espacio de almacenamiento en disco necesario, y costo en cada escenario de pruebas.

OBJETIVOS

General

Encontrar configuración óptima para un modelo de información geográfica con base en un estudio de factibilidad, considerando la forma de almacenamiento, visualización eficiente e interactiva, seguridad e integridad de la información geográfica al menor costo en recursos utilizados.

Específicos

1. Realizar un estudio bibliográfico para manejo de sistemas de información geográfica.
2. Configurar 3 escenarios de pruebas para realizar evaluación de visualización eficiente e interactiva de la información geográfica.
3. Encontrar configuración óptima para el manejo eficiente de un Sistema de Información Geográfica (SIG) tomando en cuenta el uso de información almacenada en archivos e información de base de datos.

INTRODUCCIÓN

La geografía es la ciencia de la Tierra y con el Sistema de Información Geográfica (SIG), permitirá entender y aplicar el conocimiento geográfico a una serie de actividades humanas.

Asimismo, crear conocimientos geográficos mediante la medición de la Tierra, organización de los datos, análisis y modelización de diferentes procesos y sus relaciones, permite aplicar este conocimiento a la forma de diseñar, planificar y cambiar el mundo.

Hay una creciente toma de conciencia del valor económico y estratégico de los SIG en las organizaciones, y en casi todas las industrias. Los beneficios de los SIG en general son: ahorro de costes, mayor eficiencia, mejor toma de decisiones, mejora de la comunicación y gestión geográfica.

1. MARCO TEÓRICO

1.1. Sistema de Información Geográfica

Es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica.

También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre, y construido para satisfacer unas necesidades concretas de información.

Es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada.

Son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

Esta tecnología puede ser utilizada para investigaciones científicas, gestión de los recursos, gestión de activos, arqueología, evaluación del impacto ambiental, planificación urbana, cartografía, sociología, geografía histórica, *marketing*, logística y más. Por ejemplo, un SIG podría permitir a los grupos de emergencia calcular fácilmente los tiempos de respuesta en caso de un desastre natural, o para encontrar los humedales que necesitan protección contra la contaminación; pueden ser utilizados por una empresa para ubicar un

nuevo negocio y aprovechar las ventajas de una zona de mercado con escasa competencia.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Las principales cuestiones que puede resolver un Sistema de Información Geográfica, ordenadas de menor a mayor complejidad, son:

- Localización: preguntar por las características de un lugar concreto.
- Condición: el cumplimiento o no de unas condiciones impuestas al sistema.
- Tendencia: comparación entre situaciones temporales o espaciales distintas de alguna característica.
- Rutas: cálculo de rutas óptimas entre dos o más puntos.
- Pautas: detección de pautas espaciales.
- Modelos: generación de modelos a partir de fenómenos o actuaciones simuladas.¹

¹ http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n_geogr%C3%A1fica.
Consulta: 5 de septiembre de 2013.

1.1.1. Base de datos espacial

Es un sistema administrador de bases de datos que maneja datos existentes en un espacio o datos espaciales.

En este tipo de bases de datos es imprescindible establecer un cuadro de referencia (SRE, Sistema de Referencia Espacial) para definir la localización y relación entre objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo, no es un valor absoluto. Los sistemas de referencia espacial pueden ser de dos tipos: georreferenciados (se establecen sobre la superficie terrestre. Son los que normalmente se utilizan, ya que es un dominio manipulable, perceptible y que sirve de referencia) y no georreferenciados (sistemas que tienen valor físico, pero que pueden ser útiles en determinadas situaciones).

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por el lenguaje de las computadoras actuales. Este proceso de abstracción tiene diversos niveles y normalmente comienza con la concepción de la estructura de la base de datos, generalmente en capas. En esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir.

La estructuración de la información espacial procedente del mundo real en capas conlleva cierto nivel de dificultad. En primer lugar, la necesidad de abstracción que requieren los computadores implica trabajar con primitivas básicas de dibujo, de tal forma que toda la complejidad de la realidad ha de ser reducida a puntos, líneas o polígonos.

En segundo lugar, existen relaciones espaciales entre los objetos geográficos que el sistema no puede obviar; la topología, que en realidad es el método matemático-lógico usado para definir las relaciones espaciales entre los objetos geográficos puede llegar a ser muy compleja, ya que son muchos los elementos que interactúan sobre cada aspecto de la realidad.²

1.1.2. Datos espaciales

Un modelo de datos geográfico es una abstracción del mundo real que emplea un conjunto de objetos dato, para soportar el despliegue de mapas, consultas, edición y análisis. Los datos geográficos, presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen.

Un dato espacial es una variable asociada a una localización del espacio. Normalmente se utilizan datos vectoriales, los cuales pueden ser expresados mediante tres tipos de objetos espaciales.

- Puntos: se encuentran determinados por las coordenadas terrestres medidas por latitud y longitud. Por ejemplo, ciudades, accidentes geográficos puntuales, hitos.
- Líneas: objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos, aunque debido a la forma esférica de la tierra

² http://es.wikipedia.org/wiki/Base_de_datos_espacial. Consulta: 5 de septiembre de 2013.

también se le consideran como arcos. Líneas telefónicas, carreteras y vías de trenes son ejemplos de líneas geográficas.

- Polígonos: figuras planas conectadas por distintas líneas u objetos cerrados que cubren un área determinada, como por ejemplo: países, regiones o lagos.

1.2. PostgreSQL

Es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global DevelopmentGroup).

1.2.1. Características

- Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

- Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria
 - Texto de largo ilimitado
 - Figuras geométricas
 - Direcciones IP (IPv4 e IPv6)
 - Bloques de direcciones estilo CIDR
 - Direcciones MAC
 - Arrays

Adicionalmente, los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características

- Claves ajenas: también denominadas llaves ajenas o claves foráneas (*foreignkeys*).
- Disparadores (*triggers*): se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
 - Nombre del disparador o *trigger*
 - Momento en que el disparador debe arrancar

- Sobre qué deberá activarse el disparador
- Tabla donde el disparador se activará
- Frecuencia de ejecución
- Función que podría llamar

Combinando estas seis características, PostgreSQL permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (*triggers*).

- Vistas
- Integridad transaccional
- Herencia de tablas
- Tipos de datos y operaciones geométricas
- Soporte para transacciones distribuidas

1.2.2. Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Algunos de los lenguajes que se pueden usar son los siguientes:

- PL/PgSQL (similar al PL/SQL de oracle)
- C
- C++
- Java PL/Java web

- PL/Perl
- plPHP
- PL/Python
- PL/Ruby
- PL/sh
- PL/Tcl
- PL/Scheme
- Lenguaje para aplicaciones estadísticas R por medio de PL/R

PostgreSQL soporta funciones que retornan filas, donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (*query*).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como procedimientos almacenados (*storedprocedures*)³.

1.3. PostGIS

Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la licencia pública general de GNU.

PostGIS ha sido desarrollado por la empresa canadiense RefractionResearch, especializada en productos Open Source entre los que habría que citar a Udig. PostGIS es hoy en día un producto veterano que ha

³ <http://es.wikipedia.org/wiki/PostgreSQL>. Consulta: 6 de septiembre de 2013.

demostrado versión a versión su eficiencia. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de muchos, es muy similar a la versión geográfica de la archiconocida Oracle.

Un aspecto que debemos de tener en cuenta es que PostGIS ha sido certificado en 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas también interoperables. PostGIS almacena la información geográfica en una columna del tipo GEOMETRY, que es diferente del homónimo GEOMETRY utilizado por PostgreSQL, donde se pueden almacenar la geometría en formato WKB (Well-KnownBinary), aunque hasta la versión 1.0 se utilizaba la forma WKT (Well-Known Text)⁴.

1.4. Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores⁵.

⁴ <http://es.wikipedia.org/wiki/PostGIS>. Consulta: 7 de septiembre de 2013.

⁵ <http://es.wikipedia.org/wiki/Python>. Consulta: 10 de septiembre de 2013.

1.5. GDAL

Geospatial Data Abstraction Library o GDAL es una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales, publicada bajo la MIT License por la fundación geoespacial de código abierto (Open Source Geospatial Foundation).

Como biblioteca, presenta un único modelo abstracto de datos al uso que llama para todos los formatos soportados. También viene con una variedad de utilidades en línea de comando para la traducción y el proceso de datos geoespaciales⁶.

1.6. Enhanced Compressed Wavelet ECW

Es formato de archivo propietario para imágenes *raster* desarrollado por la empresa Earth Resource Mapping. Este formato presenta ratios de compresión, desde 10:1 hasta de 50:1, mediante el uso de técnicas wavelet.

La técnica reduce considerablemente el tamaño de los archivos, manteniendo la calidad gráfica de las imágenes, permitiendo una rápida compresión y descompresión con poco uso de memoria RAM⁷.

⁶ <http://es.wikipedia.org/wiki/GDAL>. Consulta: 11 de septiembre de 2013.

⁷ http://es.wikipedia.org/wiki/Enhanced_Compressed_Wavelet.
Consulta: 12 de septiembre de 2013.

2. CONFIGURACIÓN DE INFRAESTRUCTURA

Para poder realizar el estudio, se necesitan programas para uso de sistemas de información geográfica.

Ahora se dará una breve explicación del proceso de instalación.

2.1. Instalación de software

Para realizar el estudio de factibilidad, se necesita configurar un servidor para pruebas. La configuración estará compuesta por:

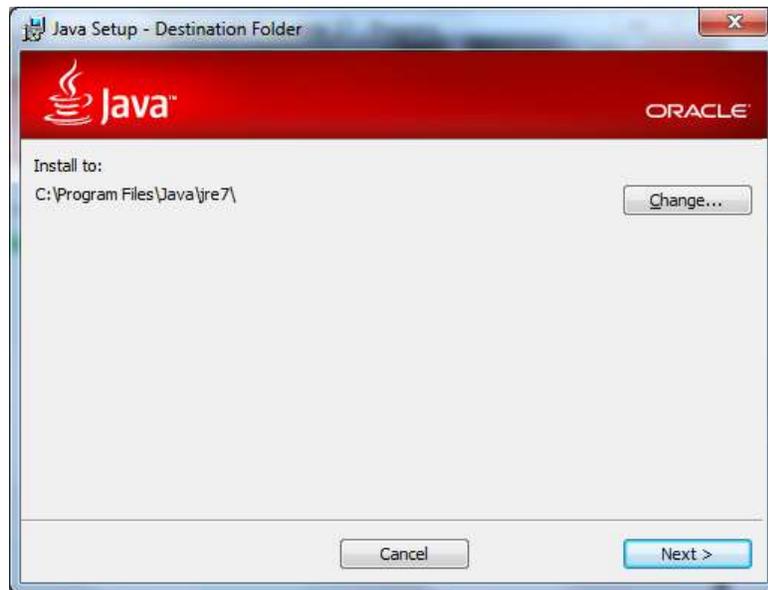
- Java, plataforma principal de trabajo.
- PostgreSQL, sistema de gestión de base de datos.
- PostGIS, módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional.
- Geoserver, servidor de mapas.
- FWTools, conjunto de programas para sistemas de información geográfica.
- QuantumGIS.

2.1.1. Instalación de Java

Algunas aplicaciones que se instalarán para poder realizar el estudio, requieren de Java, por eso se instalará primero.

En la figura 1 se puede ver la ventana de instalación de Java.

Figura 1. Instalación de Java



Fuente: elaboración propia.

Para poder usar Java en el equipo es necesario realizar la siguiente configuración:

- Configurar variable JAVA_HOME
- Configurar variable PATH

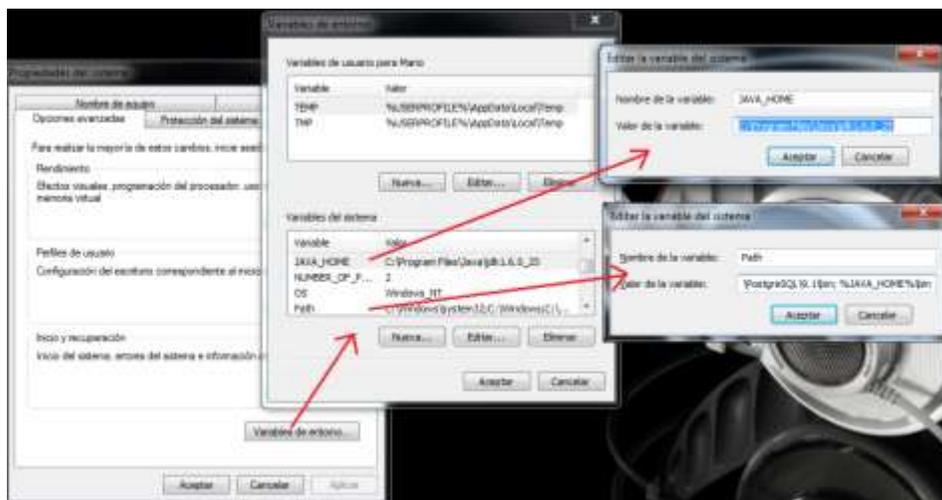
Para configurar la variable JAVA_HOME, se hace clic sobre Mi PC con el botón derecho del ratón, luego se escoge la opción Propiedades. En la ventana propiedades se escoge la opción Configuración avanzada del sistema, luego la opción variables de entorno, luego se crea o edita del listado de variables JAVA_HOME colocando el directorio de instalación de java. En la figura 2 se observa la activación de propiedades de Mi PC y en la figura 3 se observa los pasos para configurar variable JAVA_HOME Y PATH.

Figura 2. Activar propiedades de PC



Fuente: elaboración propia.

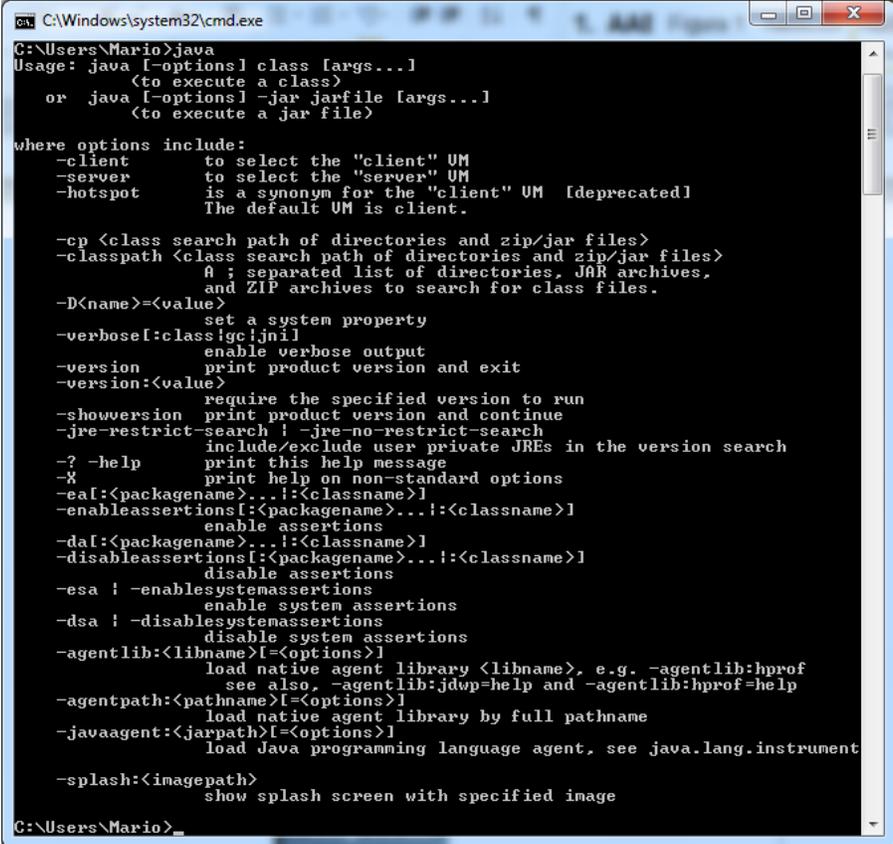
Figura 3. Configurando variable JAVA_HOME y PATH



Fuente: elaboración propia.

Para comprobar que todo esté bien configurado, se ejecuta el programa CMD de Windows y se escribe el comando java, se tendrá que mostrar información de java. Ver figura 4.

Figura 4. Información de java



```
C:\Windows\system32\cmd.exe
C:\Users\Mario>java
Usage: java [-options] class [args...]
           (to execute a class)
   or java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
  -client           to select the "client" VM
  -server          to select the "server" VM
  -hotspot         is a synonym for the "client" VM [deprecated]
                   The default VM is client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
                   A ; separated list of directories, JAR archives,
                   and ZIP archives to search for class files.
  -D<name>=<value> set a system property
  -verbose[:class[:gc]:jni] enable verbose output
  -version          print product version and exit
  -version:<value>  require the specified version to run
  -showversion     print product version and continue
  -jre-restrict-search | -jre-no-restrict-search
  -includejre | -excludejre include/exclude user private JREs in the version search
  -? -help         print this help message
  -X              print help on non-standard options
  -ea[:<packagename>...[:<classname>]]
  -enableassertions[:<packagename>...[:<classname>]]
                   enable assertions
  -da[:<packagename>...[:<classname>]]
  -disableassertions[:<packagename>...[:<classname>]]
                   disable assertions
  -esa | -enablesystemassertions
                   enable system assertions
  -dsa | -disablesystemassertions
                   disable system assertions
  -agentlib:<libname>[:<options>]
                   load native agent library <libname>, e.g. -agentlib:hprof
                   see also, -agentlib:jdwp=help and -agentlib:hprof=help
  -agentpath:<pathname>[:<options>]
                   load native agent library by full pathname
  -javaagent:<jarpath>[:<options>]
                   load Java programming language agent, see java.lang.instrument
  -splash:<imagepath>
                   show splash screen with specified image

C:\Users\Mario>
```

Fuente: elaboración propia.

2.1.2. Instalación de PostgreSQL

Como sistema de base de datos para el estudio que se realizara se utilizara la versión 9.1 de PostgreSQL, la cual se descarga del sitio web de

postgresql <http://www.enterprisedb.com/products-services-training/pgdownload#windows>.

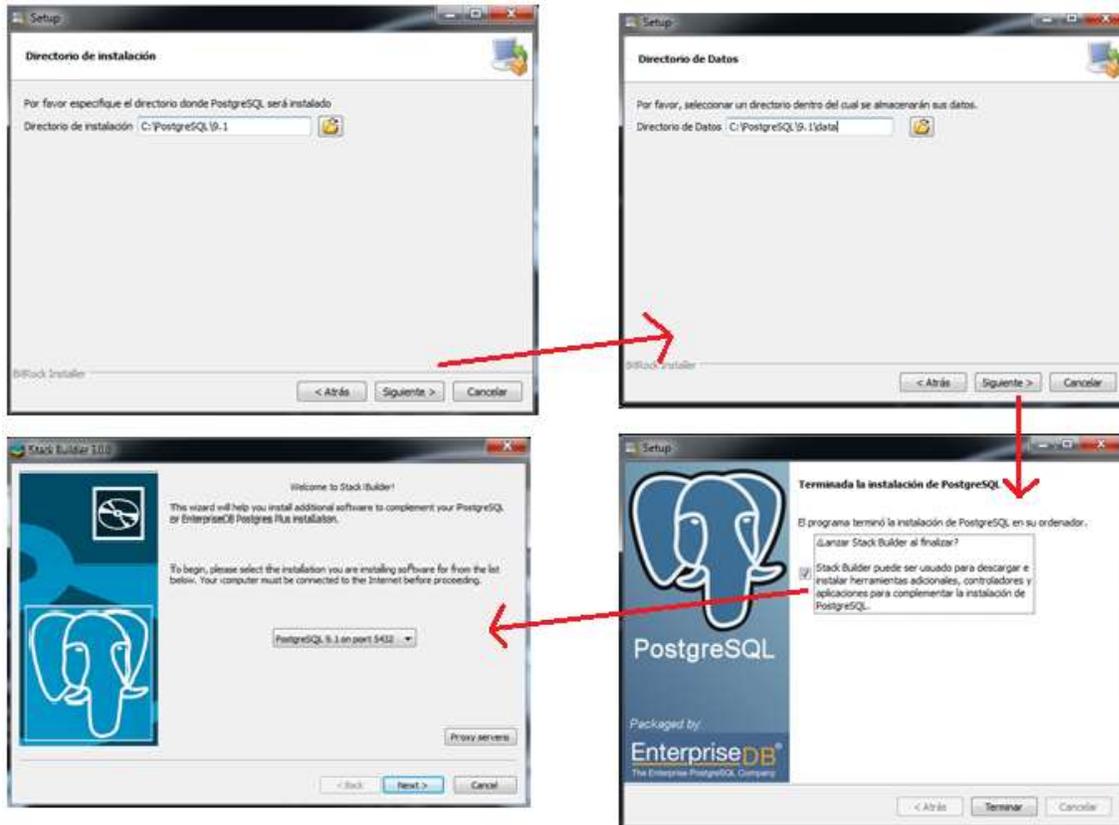
Luego de haber descargado el instalador de PostgreSQL, se hace clic con el botón derecho del ratón y se escoge la opción ejecutar como administrador. En la figura 5 se observa la ventana asistente de instalación.

Figura 5. Asistente de Instalación de PostgreSQL



Fuente: elaboración propia.

Figura 6. Configuración de instalación de PostgreSQL



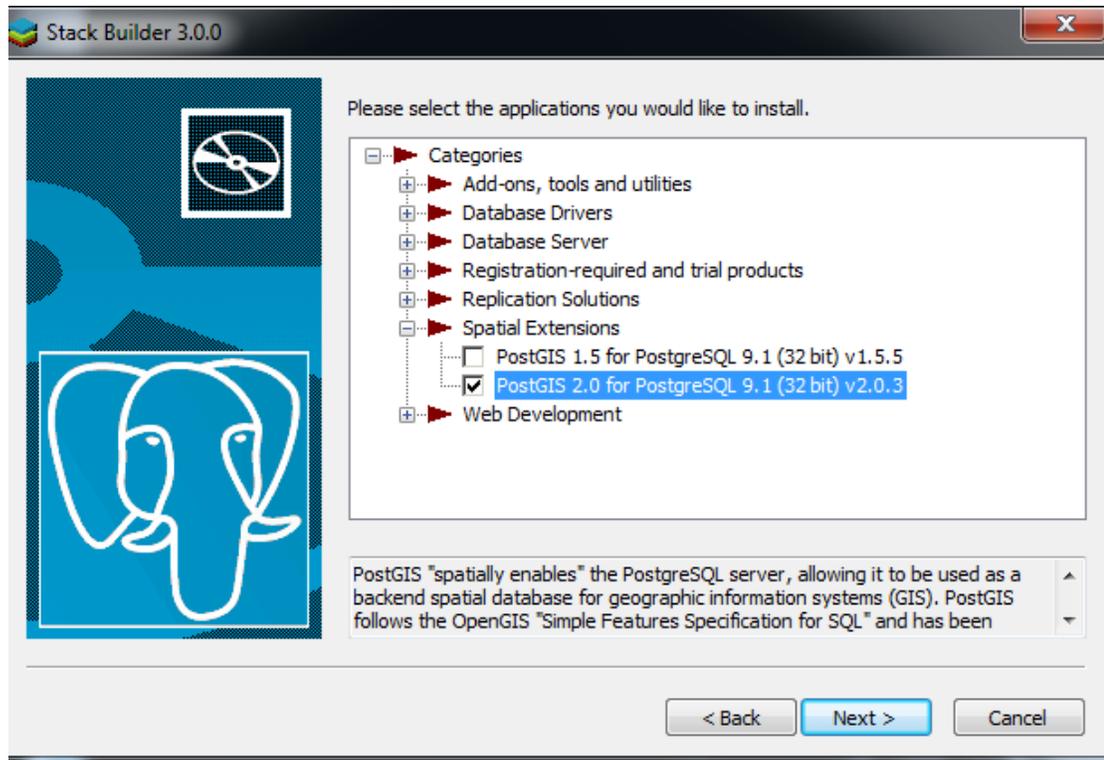
Fuente: elaboración propia.

El asistente de instalación solicita el directorio de instalación, el directorio de datos, y al terminar la instalación pregunta si se quiere correr StackBuilder para agregar herramientas adicionales a PostgreSQL. En la figura 6 se observa los pasos de instalación.

Se selecciona el checkStackBuilder en el asistente de instalación porque es necesario instalar el complemento Postgis en PostgreSQL.

En la figura 7 se observa la ventana de StackBuilder, y en ella se escoge la opción SpatialExtensions para instalar PostGIS 2.0 para PostgreSQL 9.1

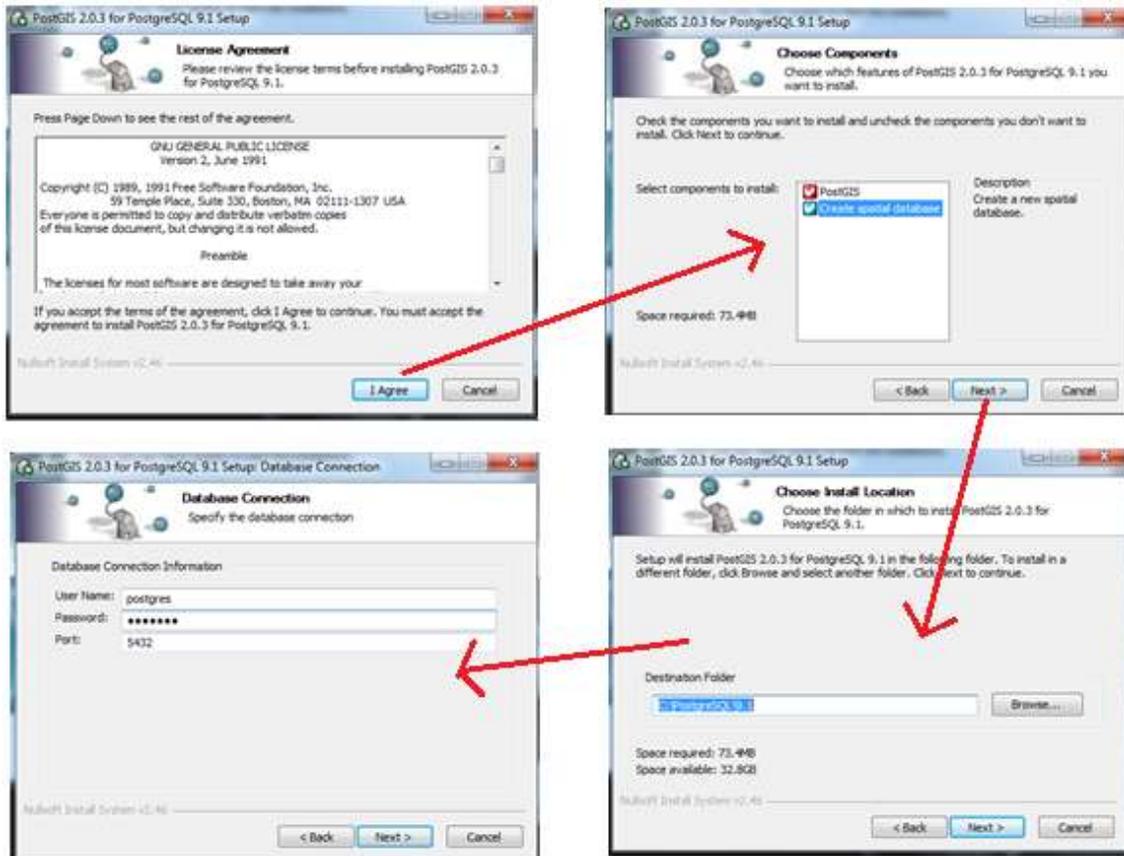
Figura 7. Selección de extensión espacial PostGIS



Fuente: elaboración propia.

Se aceptan los términos de licencia y se indica el directorio de instalación. Pedirá credenciales de conexión para poder crear base de datos espacial en PostgreSQL. En la figura 8 se puede ver ejemplo de instalación de PostGIS.

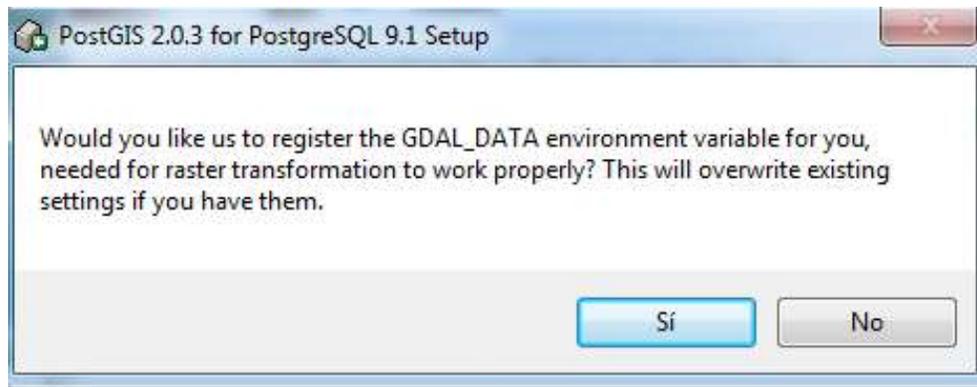
Figura 8. Instalación de PostGIS



Fuente: elaboración propia.

Al finalizar de instalar PostGIS, el programa de instalación pregunta si se quiere registrar GDAL_DATA como variable de entorno. Se escoge la opción si, en figura 9 se observa ejemplo de pregunta.

Figura 9. Generando GDAL_DATA como variable de entorno



Fuente: elaboración propia.

La instalación de PostGIS genera una tabla `spatial_ref_sys` que guarda las proyecciones de referencias espaciales.

En esta tabla hay que agregar un registro para la referencia GTM, indispensable para poder cargar las ortofotos a la base de datos.

La configuración de referencia espacial GTM se descarga de la página web www.spatialreference.org.

En la figura 10 se observa un listado con configuraciones para varias opciones, para este caso nos interesa la opción `PostGISspatialrefsys INSERT`.

Figura 10. Página web de referencia espacial GTM para PostGIS



Fuente: elaboración propia.

La instrucción sql Insert es la siguiente:

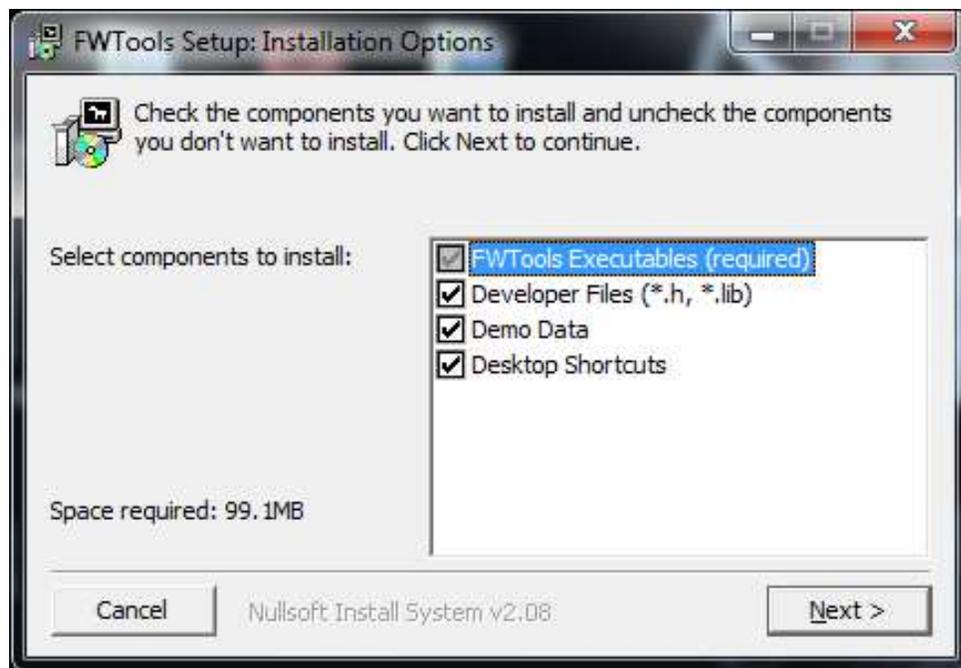
```
INSERT into spatial_ref_sys (srid, auth_name, auth_srid, proj4text, srtext) values ( 96866, 'sr-org', 6866, '+proj=tmerc +lat_0=0 +lon_0=-90.5 +k=0.9998 +x_0=500000 +y_0=0 +ellps=WGS84 +units=m +no_defs ', 'PROJCS["GTM",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-90.5],PARAMETER["Scale_Factor",0.9998],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]');
```

2.1.3. Instalación de FWTools

Es un conjunto de programas de código abierto para sistemas de información geográfica, se instalará para el estudio porque se necesita las librerías y comandos GDAL.

La aplicación se descarga del sitio web <http://fwtools.maptools.org/>. En la figura 11 se observa la ventana de instalación.

Figura 11. Instalación de FWTools



Fuente: elaboración propia.

En el proceso de instalación, se pregunta la ubicación de instalación de GDAL, dicha ubicación es la que se tiene que indicar en la variable de entorno GDAL_DATA.

Uno de los comandos que se usará para generar pirámide de capas con los archivos *raster* es `gdal_retile`, pero el archivo `gdal_retile.py` que viene en la versión 2.4.7 de FWTools tiene un error. La resolución del error se encuentra descrita en el anexo A.

2.1.4. Instalación de Geoserver

Se instalará la aplicación de Geoserver como servidor de mapas para el estudio que se realizara para los escenarios:

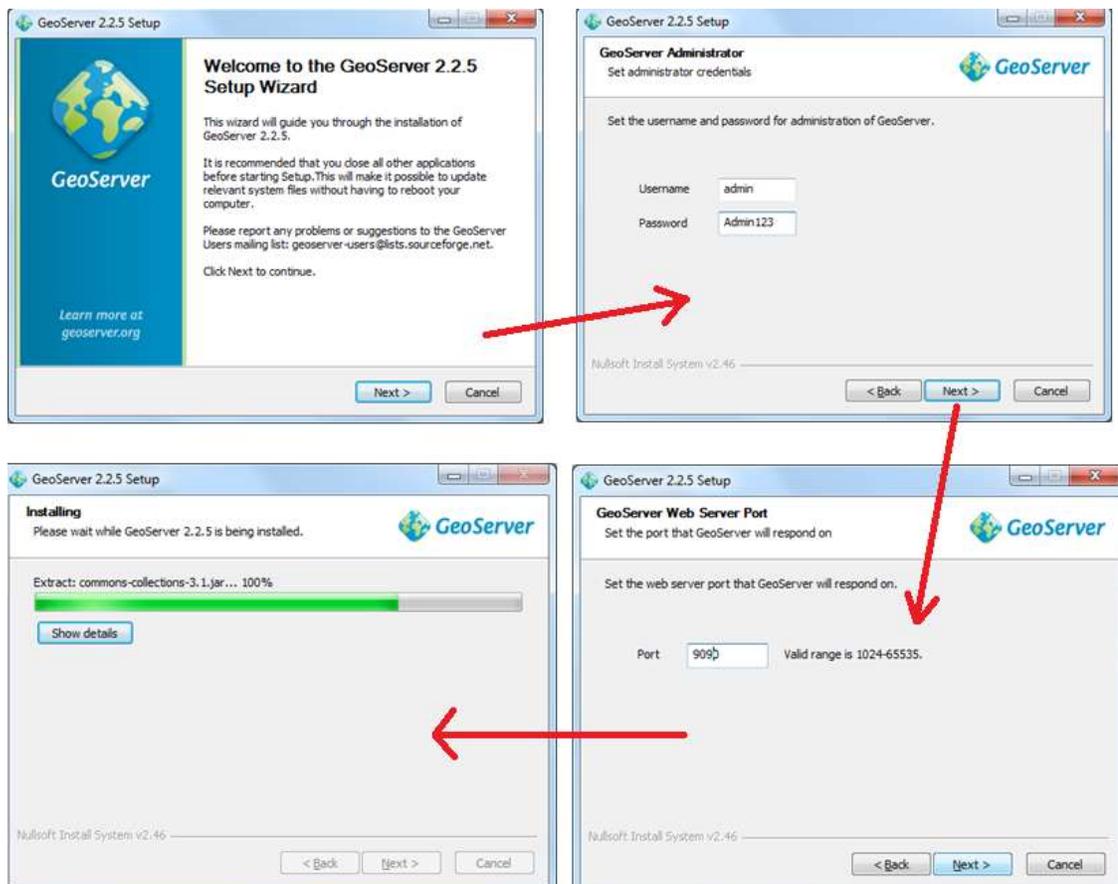
- Información *raster* almacenada en base de datos
- Ortofotos con formato TIF
- Archivos ECW

La ventana de instalación pide:

- Ubicación de instalación
- Usuario administrador
- Puerto de servicio

En la figura 12 se observan los pasos a seguir para realizar la instalación de la aplicación Geoserver.

Figura 12. Instalación de aplicación Geoserver



Fuente: elaboración propia.

Al terminar la instalación se edita el archivo `epsg.properties` ubicado en el directorio `C:\Program Files\GeoServer 2.4\data_dir\user_projections` para agregar la referencia espacial GTM.

La configuración GTM se descarga del sitio web www.spatialreference.org.

En la figura 13 se observa un listado con configuraciones para varias opciones, para este caso nos interesa la opción PRJ File.

Figura 13. Página web de referencia espacial GTM para Geoserver



Fuente: elaboración propia.

El contenido del archivo descargado lo agregamos en el archivo epsg.properties.

En la figura 14 podemos ver el contenido del archivo editado.

Figura 14. Archivo de configuración epsg.properties modificado

```
10 100009=PROJCS["WGS84 / Google Mercator", GEOGCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.251222663, AUTHORITY["EPSG","7030"]], AUTHORITY["EPSG","6326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG","4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG","9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], UNIT["m", 1.0], AUTHORITY["EPSG","900913"]]
```

```
11 102119=PROJCS["WGS84 / Google Mercator", GEOGCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.251222663, AUTHORITY["EPSG","7030"]], AUTHORITY["EPSG","6326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG","4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG","9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], UNIT["m", 1.0], AUTHORITY["EPSG","900913"]]
```

```
12 900913=PROJCS["WGS84 / Google Mercator", GEOGCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.251222663, AUTHORITY["EPSG","7030"]], AUTHORITY["EPSG","6326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG","4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG","9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], UNIT["m", 1.0], AUTHORITY["EPSG","900913"]]
```

```
13 9100=PROJCS["UTM", GEOGCS["WGS 1984", DATUM["WGS 1984", SPHEROID["WGS 1984", 6378137.0, 298.251222663], PRIMEM["Greenwich", 0.0], UNIT["degree", 0.017453292519943295], AXIS["Longitude", EAST], AXIS["Latitude", NORTH]], PROJECTION["Transverse_Mercator"], PARAMETER["central_meridian", -90.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["scale_factor", 0.9996], PARAMETER["false_easting", 500000.0], PARAMETER["false_northing", 0.0], UNIT["m", 1.0], AXIS["x", EAST], AXIS["y", NORTH]
```

Fuente: elaboración propia.

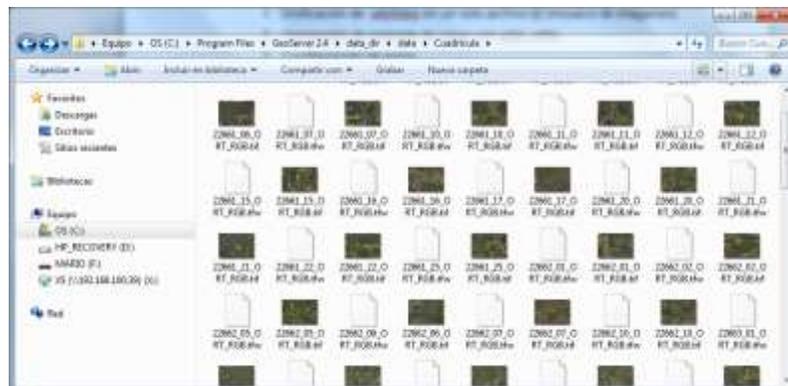
3. PROCESO DE CARGA DE ORTOFOTOS A BASE DE DATOS

En este capítulo se abordara el tema de carga de ortofotos en la base de datos postgresql, el proceso implica lo siguiente:

- Unificación de ortofotos de estudio en un solo archivo TIF (mosaico de imágenes)
- Generación de pirámide de capas con gdal_retile
- Configuración de mapa
- Generación de scripts para generar tablas necesarias en base de datos
- Importar capas (imágenes) a base de datos

En la figura 15 se observa la carpeta Cuadrícula con todos los archivos ortofotos a cargarse en la base de datos postgresql.

Figura 15. Ilustración de ortofotos en carpeta Cuadrícula

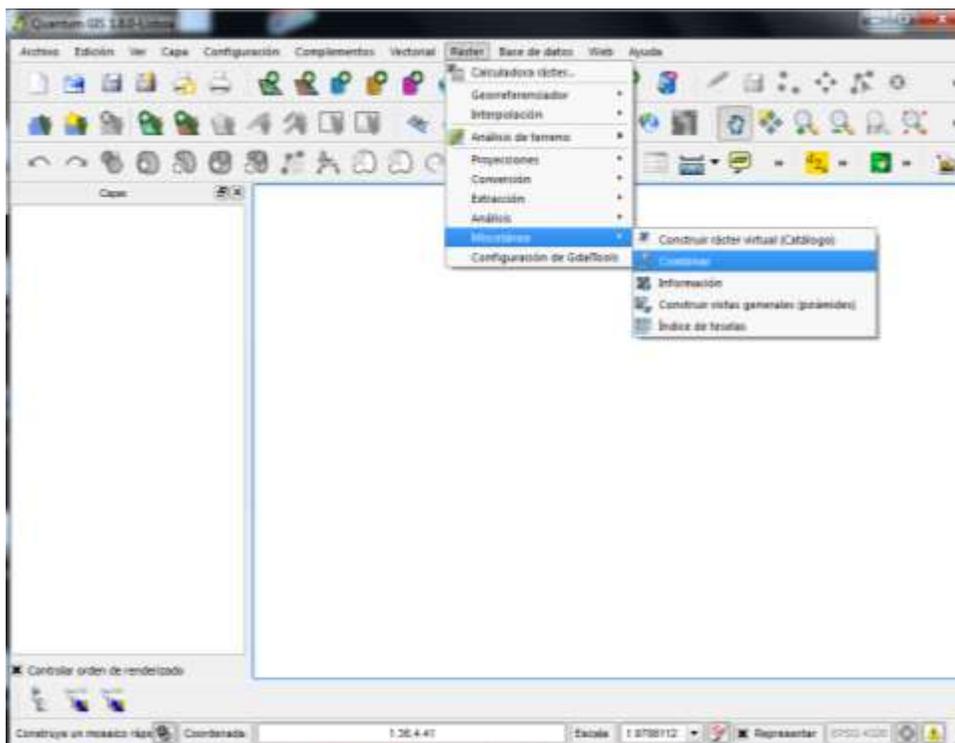


Fuente: elaboración propia.

3.1. Unificación de ortofotos (mosaico de imágenes)

Para proceder con la generación del mosaico de ortofotos, se utilizará el programa QuantumGis. En el menú de opciones de la aplicación seleccionamos la opción Raster, luego la opción Miscelánea, y luego la opción Combinar. En la figura 16 se puede ver la selección de opciones para generar el mosaico.

Figura 16. Opción combinar del menú *raster*

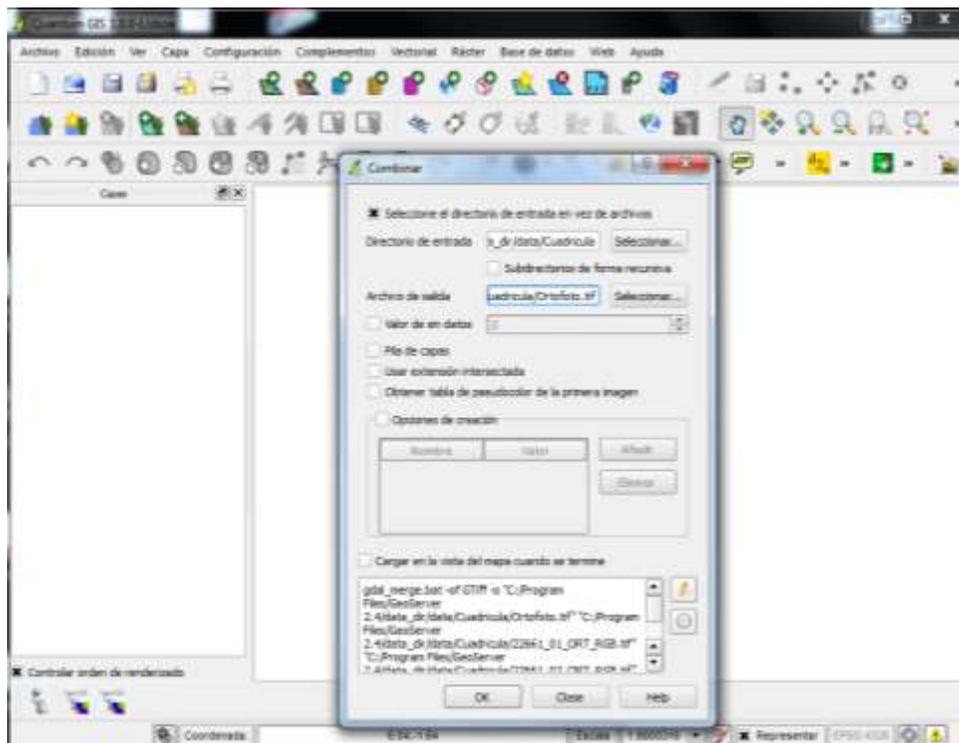


Fuente: elaboración propia.

La aplicación activa la ventana para realizar la combinación de las imágenes, activamos la opción directorio de entrada ya que se procederá con unificar varios archivos y con esta opción la aplicación selecciona todos los archivos del directorio especificado. Indicamos el archivo de salida Ortofoto.tif y

luego activamos el botón OK, con esto se inicia el proceso de unificación de todas las ortofotos. En la figura 17 se puede ver el formulario para combinar ortofotos.

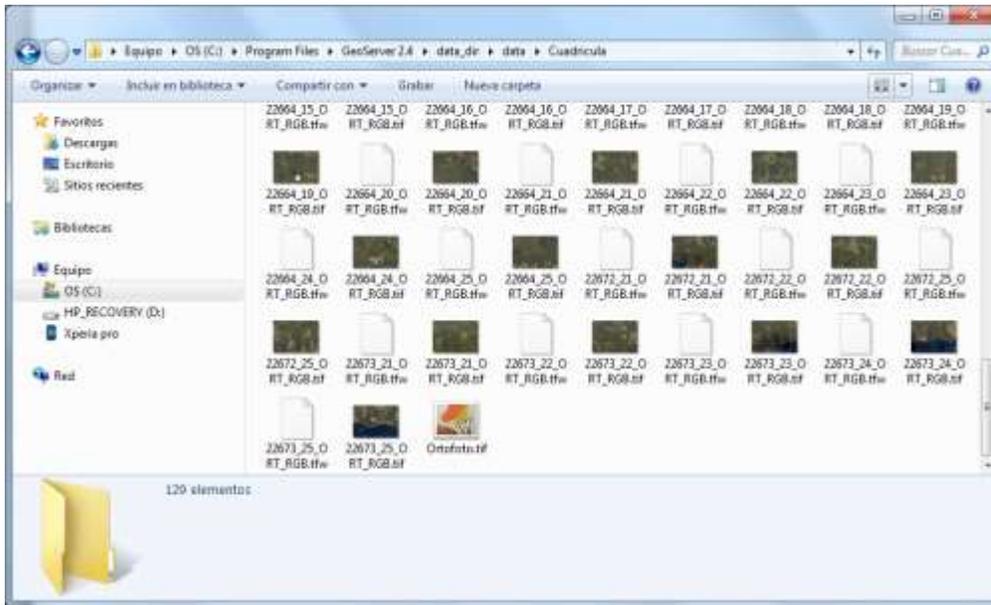
Figura 17. Formulario para combinar ortofotos en QuantumGIS



Fuente: elaboración propia.

El tiempo que tarda el proceso es proporcional a la cantidad de imágenes que se unificarán. En la figura 18 se puede observar el documento generado Ortofoto.tif (mosaico de imágenes).

Figura 18. Documento generado con la combinación de ortofotos



Fuente: elaboración propia.

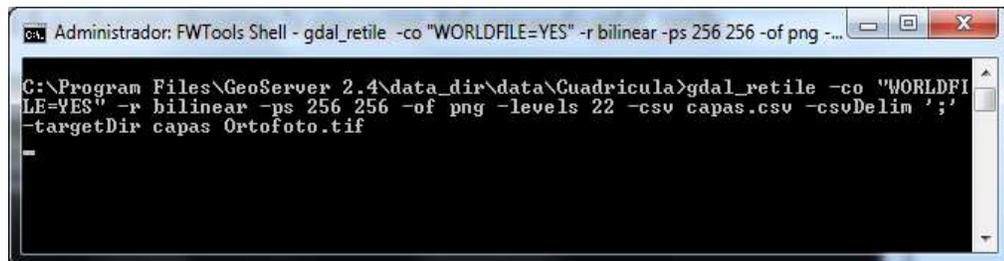
3.2. Generar pirámide de capas

Para proceder con la generación de pirámide de capas, se utilizará el comando `gdal_retile` que viene con la aplicación `FWTools`. La instrucción es la siguiente:

```
gdal_retile -co "WORLDFILE=YES" -r bilinear -ps 256 256 -of png -levels 22 -csv capas.csv -csvDelim ';' -targetDir capas Ortofoto.tif
```

En la figura 19 se puede ver la ejecución del comando `gdal_retile`.

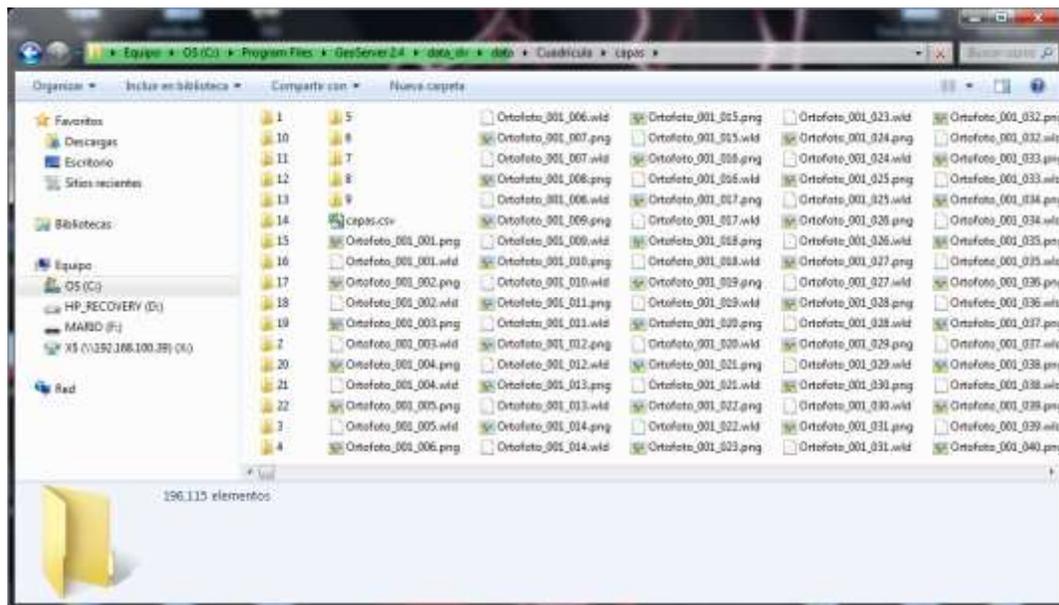
Figura 19. Comando gdal_retile en ejecución



Fuente: elaboración propia.

En la figura 20 se puede ver la información de capas generada en la carpeta capas, esta es la capa 0 y dentro de ella se encuentran 22 carpetas para las demás capas (22 capas en total)

Figura 20. Capas Generadas de mosaico de ortofotos (nivel 0)



Fuente: elaboración propia.

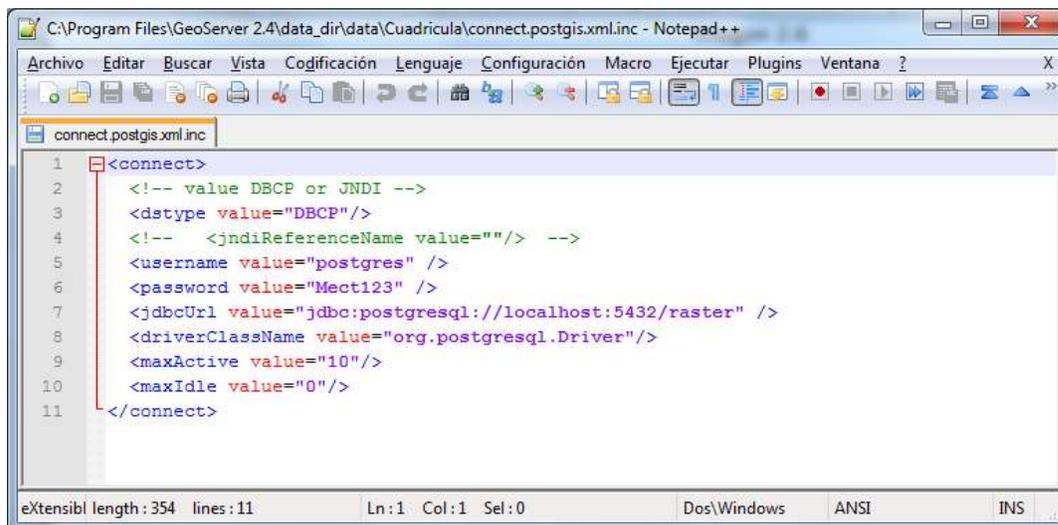
3.3. Configuración de mapa

La configuración del mapa se realiza por medio de tres archivos xml.

- Archivo de conexión a base de datos connect.postgis.xml.inc
- Información de asignación para tablas mapping.postgis.xml.inc
- Datos de configuración para el mapa osm.postgis.xml

En las figuras 21, 22, y 23 se puede ver la configuración que contiene cada archivo xml.

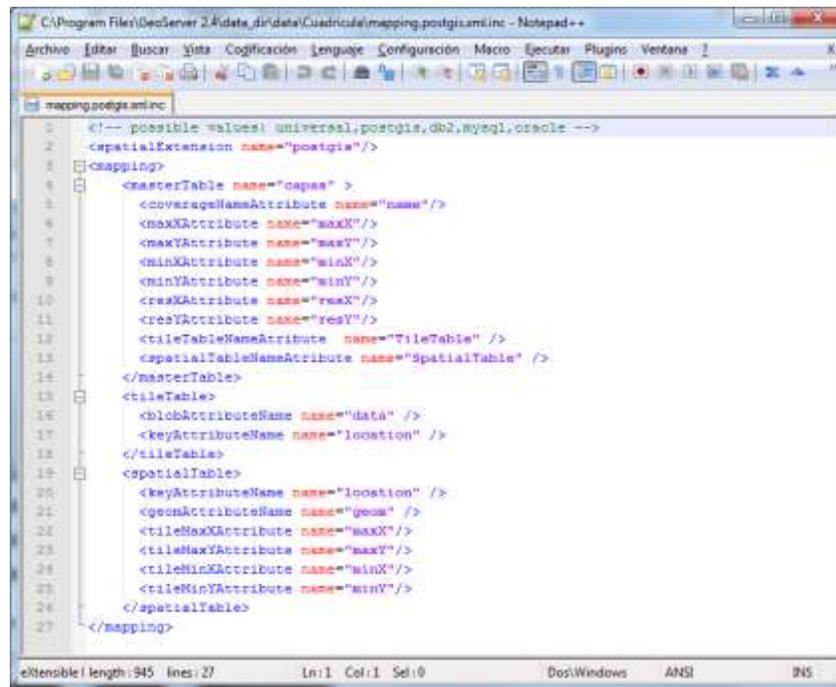
Figura 21. Xml con configuración de conexión a base de datos



```
1 <connect>
2   <!-- value DBCP or JNDI -->
3   <dstype value="DBCP"/>
4   <!-- <jndiReferenceName value=""/> -->
5   <username value="postgres" />
6   <password value="Meot123" />
7   <jdbcUrl value="jdbc:postgresql://localhost:5432/raster" />
8   <driverClassName value="org.postgresql.Driver"/>
9   <maxActive value="10"/>
10  <maxIdle value="0"/>
11 </connect>
```

Fuente: elaboración propia.

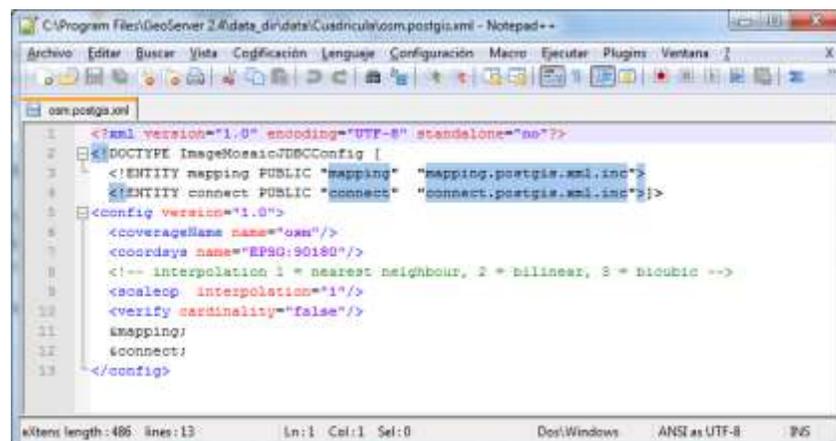
Figura 22. Xml con Información de asignación para tablas



```
1 <!-- possible values: universal, postgis, db2, mysql, oracle -->
2 <spatialExtension name="postgis"/>
3 <mapping>
4   <masterTable name="ogpms" >
5     <coverageNameAttribute name="name"/>
6     <maxXAttribute name="maxX"/>
7     <maxYAttribute name="maxY"/>
8     <minXAttribute name="minX"/>
9     <minYAttribute name="minY"/>
10    <crossAttribute name="maxX"/>
11    <creaYAttribute name="yesY"/>
12    <tileTableNameAttribute name="TileTable" />
13    <spatialTableNameAttribute name="SpatialTable" />
14  </masterTable>
15  <tileTable>
16    <blobAttribute name="data" />
17    <keyAttribute name="lposition" />
18  </tileTable>
19  <spatialTable>
20    <keyAttribute name="lposition" />
21    <geomAttribute name="geom" />
22    <tileMaxXAttribute name="maxX"/>
23    <tileMaxYAttribute name="maxY"/>
24    <tileMinXAttribute name="minX"/>
25    <tileMinYAttribute name="minY"/>
26  </spatialTable>
27 </mapping>
```

Fuente: elaboración propia.

Figura 23. Xml con configuración de mapa



```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE ImageNoasicJDBCConfig [
3   <!ENTITY mapping PUBLIC "mapping" "mapping.postgis.xml.inc">
4   <!ENTITY connect PUBLIC "connect" "connect.postgis.xml.inc">
5 <config version="1.0">
6   <coverageName name="osm"/>
7   <coordsys name="EPSG:90180"/>
8   <!-- Interpolation 1 = nearest neighbour, 2 = bilinear, 3 = Bicubic -->
9   <scaleop interpolation="1"/>
10  <verify cardinality="false"/>
11  <mapping/>
12  <connect/>
13 </config>
```

Fuente: elaboración propia.

3.3.1. Generar *script* para crear tablas PostGIS

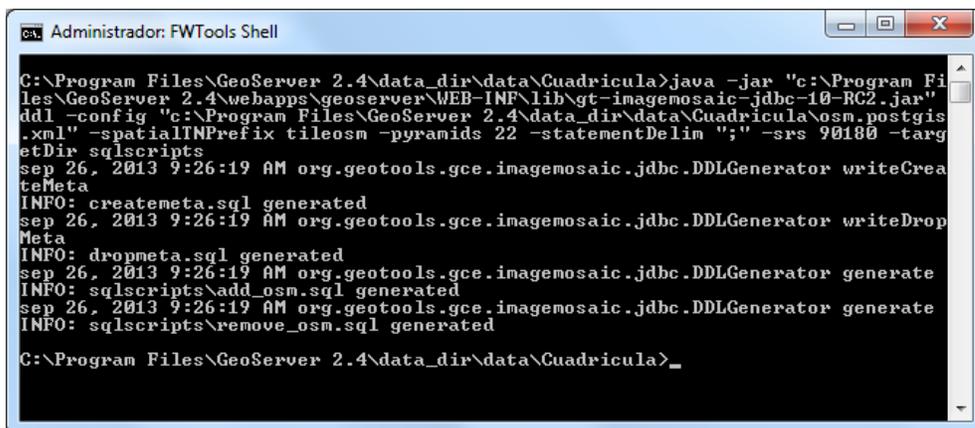
Para crear los *scripts* con instrucciones sql para generar las tablas en la base de datos, se utiliza la utilidad de generación de DDL java.

Para el estudio, se necesita generar la pirámide de 22 capas, la instrucción es la siguiente:

```
java -jar "c:\Program Files\GeoServer 2.4\webapps\geoserver\WEB-INF\lib\gt-  
imagenesa- jdbc-10-RC2.jar" ddl -config "c:\Program Files\GeoServer  
2.4\data_dir\data\Cuadrícula\osm.postgis.xml" -spatialTNPrefix tileosm -pyramids  
22 -statementDelim ";" -srs 90180 -targetDir sqlscripts
```

El comando ejecutado se observa en la figura 24.

Figura 24. Generar archivos de configuración para crear tablas en base de datos

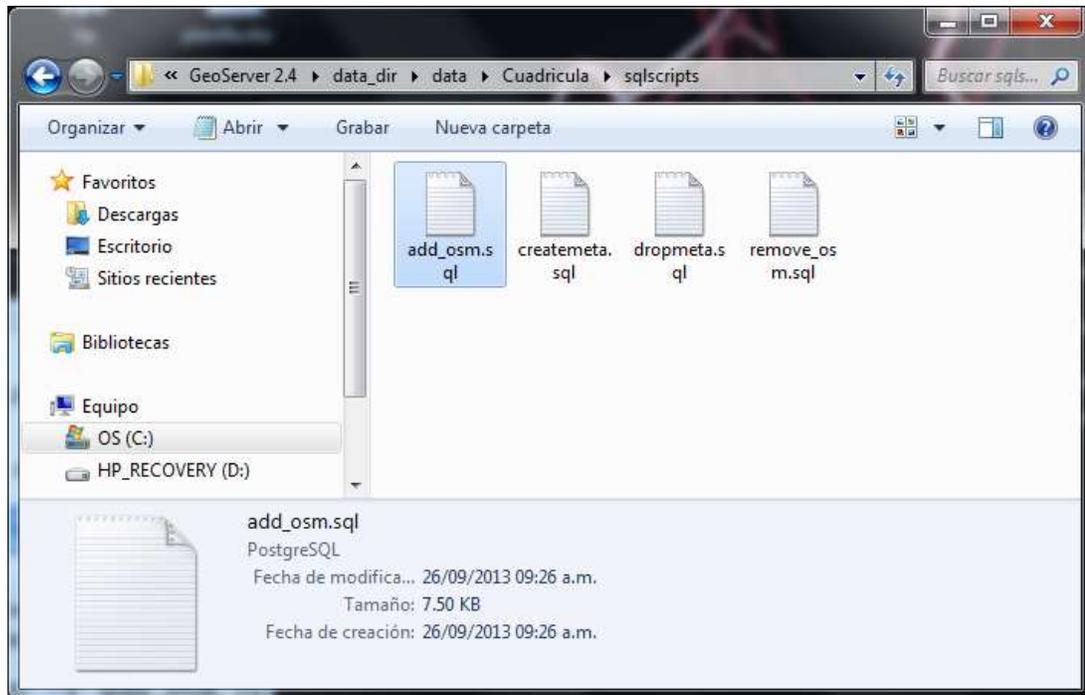


```
ca. Administrador: FWTools Shell  
C:\Program Files\GeoServer 2.4\data_dir\data\Cuadrícula>java -jar "c:\Program Fi  
les\GeoServer 2.4\webapps\geoserver\WEB-INF\lib\gt-imagenesa- jdbc-10-RC2.jar"  
ddl -config "c:\Program Files\GeoServer 2.4\data_dir\data\Cuadrícula\osm.postgis  
.xml" -spatialTNPrefix tileosm -pyramids 22 -statementDelim ";" -srs 90180 -targ  
etDir sqlscripts  
sep 26, 2013 9:26:19 AM org.geotools.gce.imagemosaic.jdbc.DDLGenerator writeCrea  
teMeta  
INFO: createmeta.sql generated  
sep 26, 2013 9:26:19 AM org.geotools.gce.imagemosaic.jdbc.DDLGenerator writeDrop  
Meta  
INFO: dropmeta.sql generated  
sep 26, 2013 9:26:19 AM org.geotools.gce.imagemosaic.jdbc.DDLGenerator generate  
INFO: sqlscripts\add_osm.sql generated  
sep 26, 2013 9:26:19 AM org.geotools.gce.imagemosaic.jdbc.DDLGenerator generate  
INFO: sqlscripts\remove_osm.sql generated  
C:\Program Files\GeoServer 2.4\data_dir\data\Cuadrícula>_
```

Fuente: elaboración propia.

La instrucción anterior genera cuatro archivos sql en la carpeta sqlscripts. En la figura 25 se observan los archivos generados.

Figura 25. Archivos sql generados



Fuente: elaboración propia.

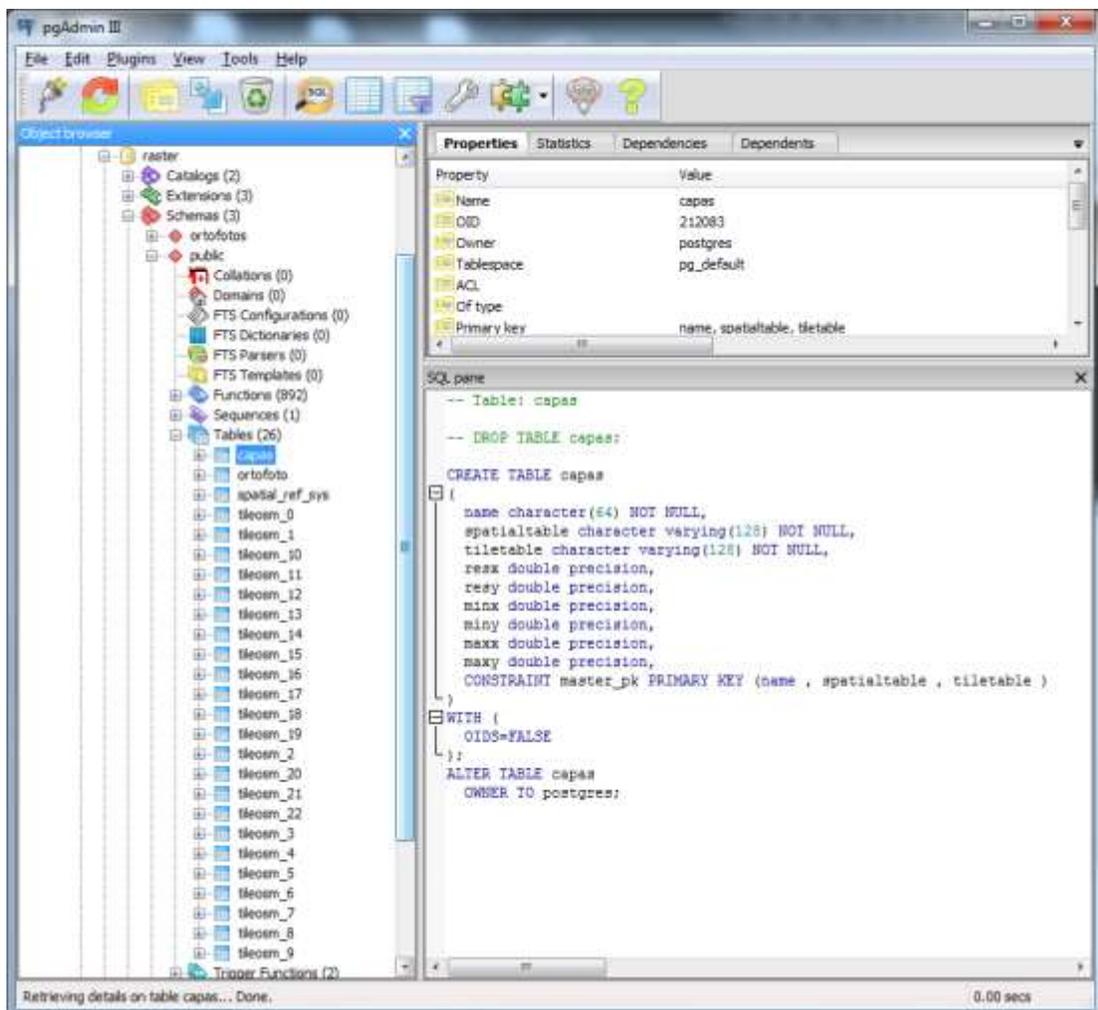
3.3.2. Generando las tablas en la base de datos

Con los archivos creados (<code>createmeta.sql</code> y <code>add_osm.sql</code>) y el comando <code>psql</code> de Postgresql se procede a crear las tablas necesarias para guardar información de capas *raster* en la base de datos. La ejecución del comando se puede ver en la figura 26 y 27.

En la figura 28 se pueden ver las tablas agregadas en la base de datos raster.

- capas: tabla principal
- tileosm_0...tileosm_22: tablas para capas de nivel 0 a nivel 22

Figura 28. Tablas generadas en la base de datos



Fuente: elaboración propia.

3.4. Importar capas a base de datos PostgreSql

Para cargar información de las capas con imágenes en la base de datos se usa el paquete `gt-imagemosaic-jdbc-10-RC2.jar`, la instrucción es la siguiente:

```
java -jar "c:\Program Files\GeoServer 2.4\webapps\geoserver\WEB-INF\lib\gt-  
imagemosaic-jdbc-10-RC2.jar" import -config "c:\Program Files\GeoServer  
2.4\data_dir\data\Cuadrícula\osm.postgis.xml" -spatialTNPrefixtileosm -  
tileTNPrefixtileosm -csv capas\capas.csv -csvDelim ';'.  
.
```

4. ESTUDIO DE FACTIBILIDAD

Para realizar el estudio se necesita configurar los siguientes escenarios:

- Configurar capa en Geoserver con información almacenada en base de datos.
- Configurar capa en Geoserver con ortofotos en formato ECW.
- Configurar capa en Geoserver con ortofotos en formato TIF.

4.1. Configurar capa en Geoserver con información almacenada en base de datos

Luego de haber cargado las imágenes de todas las capas a la base de datos, se procede a realizar el proceso de publicación de la información *raster* en Geoserver.

Es importante recalcar que para realizar la configuración se utilizara el paquete `gt-imagemosaic-jdbc-10-RC2.jar`, la cual provee las herramientas para leer las capas *raster* almacenadas en base de datos, y el paquete `postgresql-9.1-903.jdbc4.jar` para realizar conexiones a base de datos postgresql.

Otra de las consideraciones es que tanto Postgresql y Geoserver deben tener configurado como referencia espacial la proyección GTM 90180. En Postgresql, en la tabla `spatial_ref_sys` se debe agregar el registro de proyección, y en Geoserver agregarlo en el archivo `epsg.properties` ubicado en el directorio `C:\Program Files\GeoServer 2.4\data_dir\user_projections\`

En las figuras 29 y 30 se observa el registro de la tabla spatial_ref_sys y el archivo de configuración epsg.properties de Geoserver.

Figura 29. Registro con referencia espacial GTM para PostGIS

srid	auth_name	auth_srid	auth_authorized	proj4text	proj4init
31464	EPSG	31464	PROJCS["WGS 84 / UTM zone 18Q", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	PROJCS["WGS 84 / UTM zone 18Q", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	
31465	EPSG	31465	PROJCS["WGS 84 / UTM zone 18R", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	PROJCS["WGS 84 / UTM zone 18R", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	
31466	EPSG	31466	PROJCS["WGS 84 / UTM zone 18S", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	PROJCS["WGS 84 / UTM zone 18S", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	
31467	EPSG	31467	PROJCS["WGS 84 / UTM zone 18T", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	PROJCS["WGS 84 / UTM zone 18T", GEOCS["WGS 84", DATUM["WGS 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]	

Fuente: elaboración propia.

Figura 30. Archivo epsg.properties con referencia espacial GTM para Geoserver

```

100009=PROJCS["WGS84 / Google Mercator", GEOCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], AUTHORITY["EPSG", "4326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]
102119=PROJCS["WGS84 / Google Mercator", GEOCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], AUTHORITY["EPSG", "4326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]
900913=PROJCS["WGS84 / Google Mercator", GEOCS["WGS 84", DATUM["World Geodetic System 1984", SPHEROID["WGS 84", 6378137.0, 298.257222563], AUTHORITY["EPSG", "7030"]], AUTHORITY["EPSG", "4326"]], PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PROJECTION["Mercator (ISF)", AUTHORITY["EPSG", "9804"]], PARAMETER["semi_major", 6378137.0], PARAMETER["semi_minor", 6378137.0], PARAMETER["latitude_of_origin", 0.0], PARAMETER["central_meridian", 0.0], PARAMETER["scale_factor", 1.0], PARAMETER["false_easting", 0.0], PARAMETER["false_northing", 0.0], AUTHORITY["EPSG", "900913"]]
9804=PROJCS["UTM", GEOCS["WGS 1984", DATUM["WGS 1984", SPHEROID["WGS 1984", 6378137.0, 298.257222563], PRIMEM["Greenwich", 0.0], UNIT["degree", 0.017453292519943295], AUTHORITY["EPSG", "4326"]], PRIMEM["Longitude", 0.0], AUTHORITY["EPSG", "8901"]], PROJECTION["Transverse_Mercator"], PARAMETER["central_meridian", -90.5], PARAMETER["latitude_of_origin", 0.0], PARAMETER["scale_factor", 0.9998], PARAMETER["false_easting", 500000.0], PARAMETER["false_northing", 0.0], UNIT["m", 1.0], AXIS["X", EAST], AXIS["Y", NORTH]
  
```

Fuente: elaboración propia.

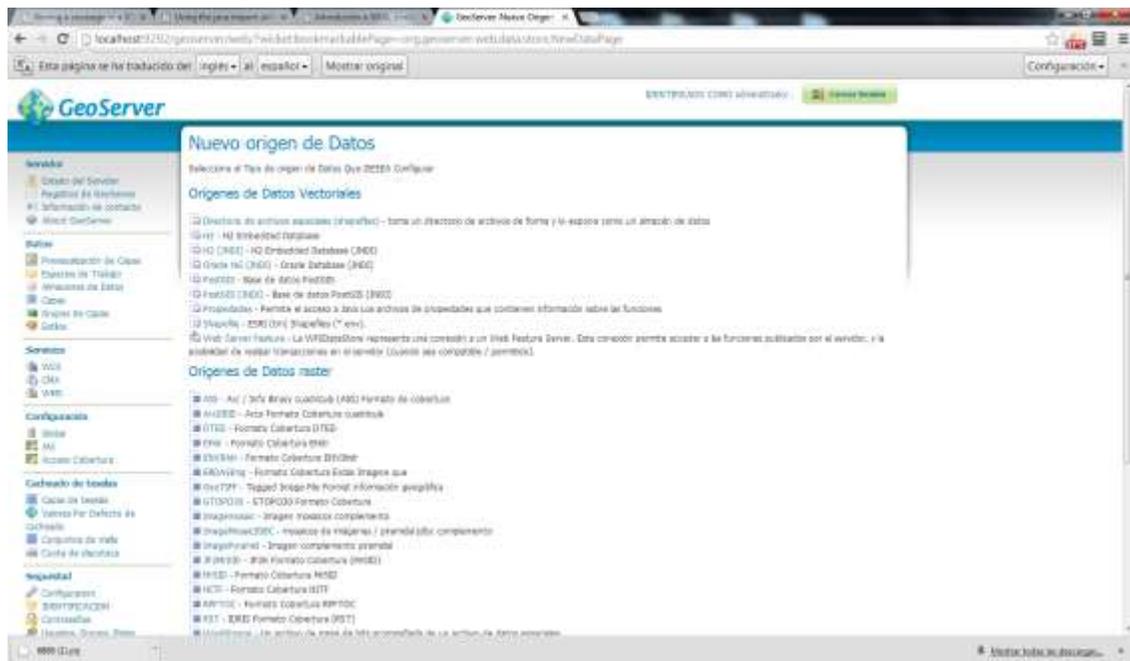
4.1.1. Publicar capa raster

El portal Geoserver provee un menú de opciones en el lado izquierdo, los pasos a seguir para realizar la configuración de publicación son:

- Crear Espacio de Trabajo
- Agregar almacén de datos
- Crear capa espacial y publicarla

En la figura 31 se observa la interfaz que provee Geoserver para agregar origen de datos. Para este caso se escoge la opción ImageMosaicJDBC de origen de datos raster.

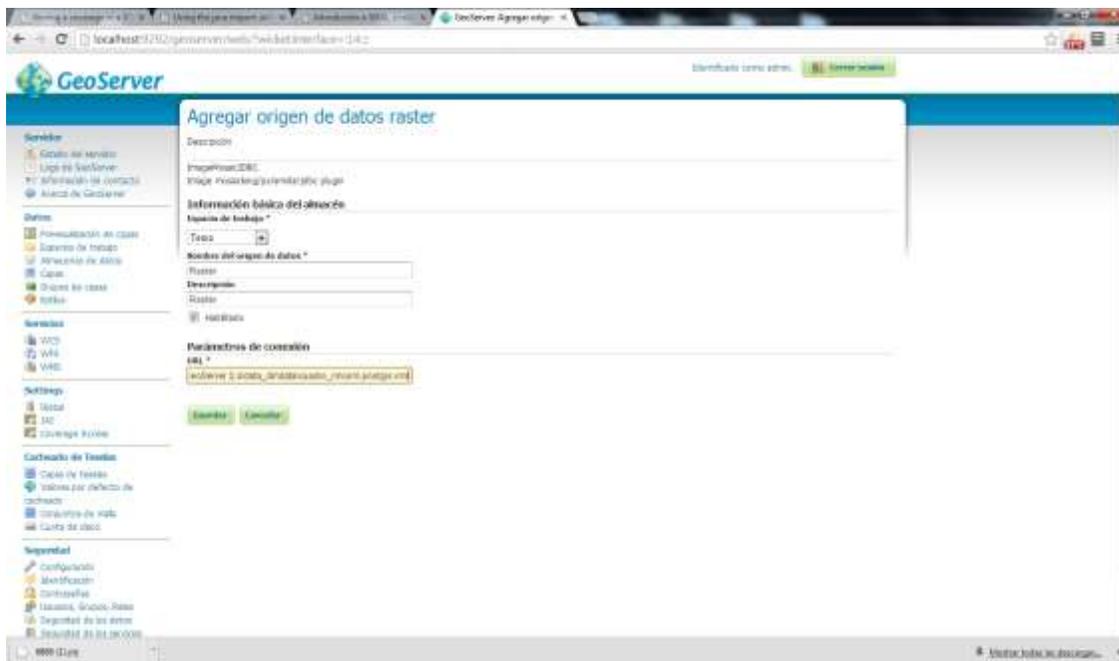
Figura 31. Interfaz origen de datos Geoserver



Fuente: elaboración propia.

Luego de escoger la opción ImageMosaicJDBC aparece el formulario para agregar origen de datos *raster*. Se escoge el espacio de trabajo Tesis, se coloca Raster como nombre y Raster como descripción, en parámetros de conexión se especifica la ubicación del archivo xml de configuración de mapa osm.postgis.xml. En la figura 32 se puede ver formulario de origen de datos.

Figura 32. Interfaz de origen de datos ImageMosaicJDBC



Fuente: elaboración propia.

Al guardar el origen de datos se muestra un listado de capas configuradas para publicación, es este caso aparece solo la capa osm y la opción publicación. En la figura 33 se puede ver la capa configurada.

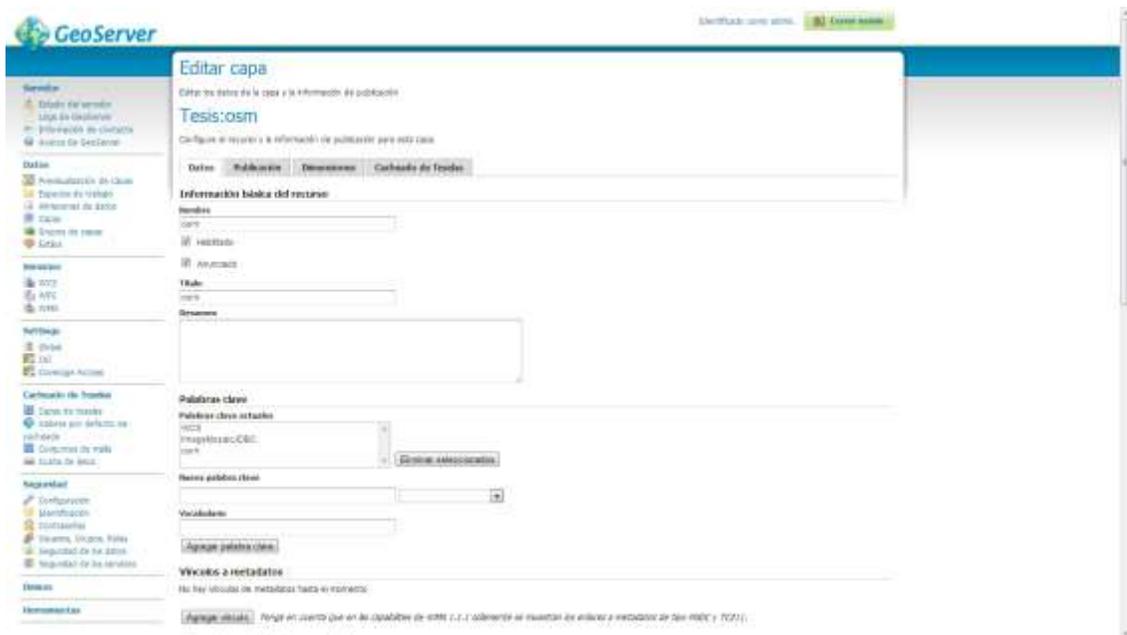
Figura 33. Capa raster con información PostGIS



Fuente: elaboración propia.

Se escoge la opción de publicación y Geoserver muestra datos de configuración de la capa almacenada en la base de datos. Ver figura 34 y 35.

Figura 34. Editar capa raster PostGIS (1)



Fuente: elaboración propia.

Figura 35. Editar capa *raster* PostGIS (2)

Sistema de referencia de coordenadas:

SRID destino:

SRID de destino:

Proyector de datos:

Proyector de destino al destino:

Encuadres:

Encuadre nativo:

Mín X	Mín Y	Máx X	Máx Y
170750.74	1.079200.20	180210.74	1.580220.20

Calcular desde los datos

Encuadre destino:

Mín X	Mín Y	Máx X	Máx Y
-98.7510738075; 16.8023801597	-98.8488030120; 17.8012888840		

Calcular desde el encuadre nativo

Parámetros de la cobertura:

BackgroundColor:

OutputTransparentColor:

Detalle de las bandas de la cobertura:

Bandas	Tipo de datos	Valores nulos	Unidades
com	int (formato 8 bits)	-	<input type="text" value=""/>
com	int (formato 8 bits)	-	<input type="text" value=""/>
com	int (formato 8 bits)	-	<input type="text" value=""/>

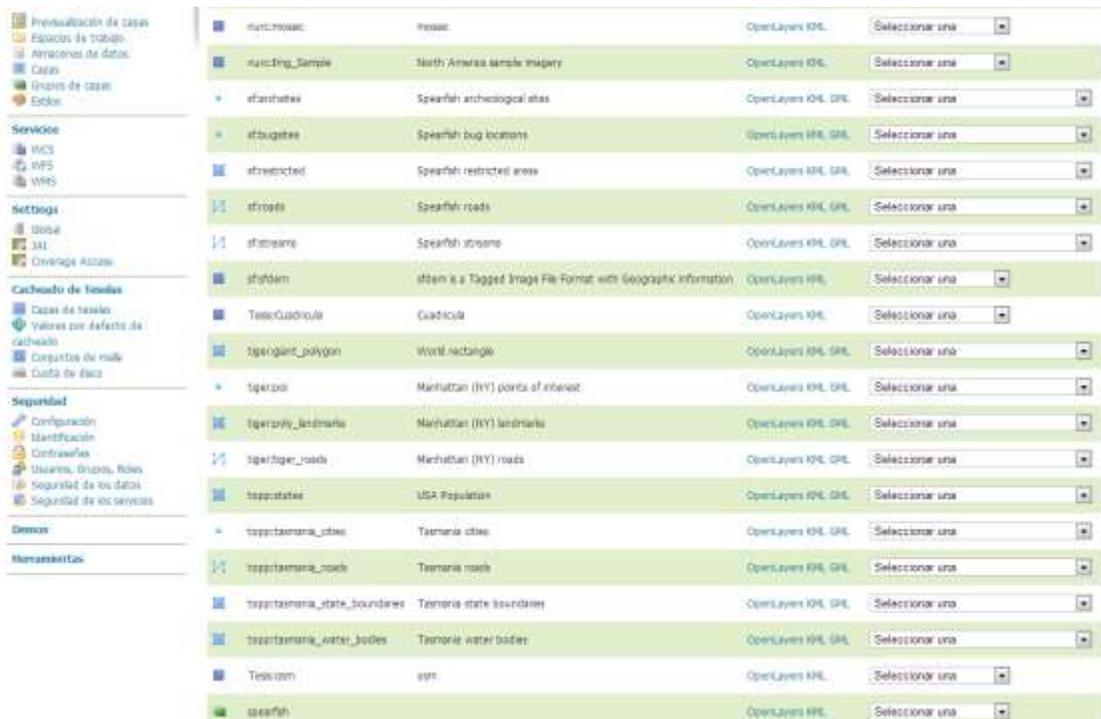
Actualizar las definiciones de datos

Fuente: elaboración propia.

Verificamos el sistema de referencia de coordenadas, y los encuadres (x,y) mínimos y (x,y) máximos, si todo esta correcto se hace clic en el botón guardar, y con esto se tendrá configurado la capa *raster* PostGIS (información *raster* almacenada en la base de datos).

En la figura 36 se observa un listado de capas, y entre ellas la capa osm, que es la que se acaba de configurar.

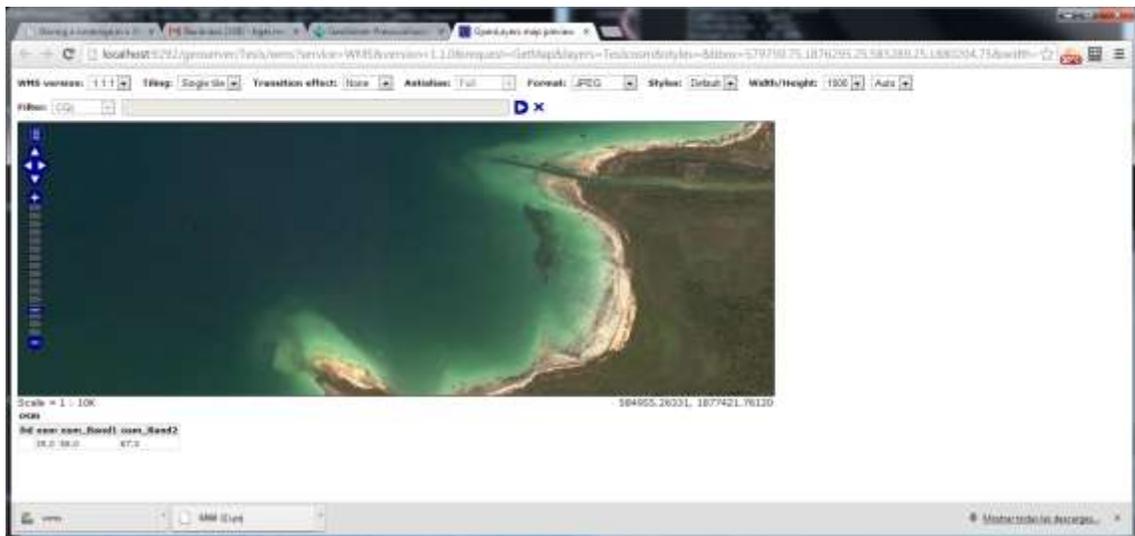
Figura 37. Publicando capa raster PostGIS



Fuente: elaboración propia.

Luego de escoger la capa osm, se observa la previsualización de la capa raster almacenada en la base de datos con *openlayers*. En la figura 38 se observa la capa raster.

Figura 38. Capa raster visualizado en OpenLayers



Fuente: elaboración propia.

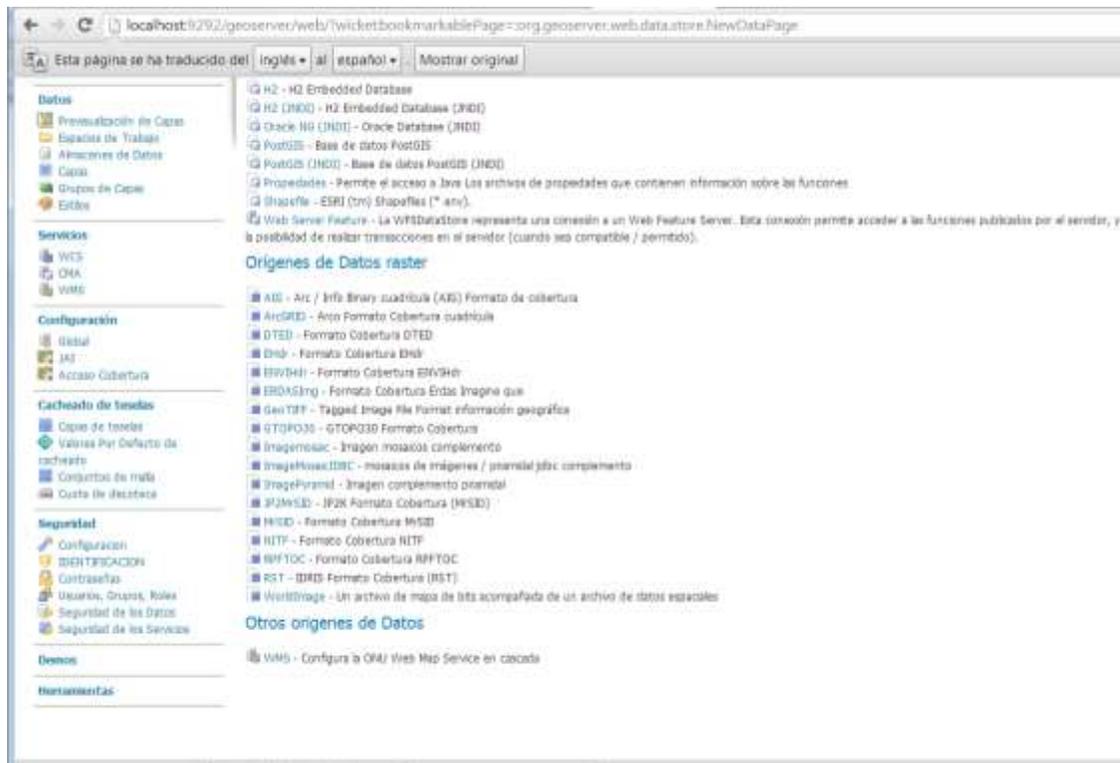
4.2. Configurar capa en Geoserver con ortofotos en formato TIF

El portal Geoserver provee un menú de opciones en el lado izquierdo, los pasos a seguir para realizar la configuración de publicación son:

- Crear Espacio de Trabajo
- Agregar almacén de datos
- Crear capa espacial y publicarla

En la figura 39 se observa la interfaz que provee Geoserver para agregar origen de datos. Para este caso se escoge la opción Imagemosaic de origen de datos *raster*.

Figura 39. Configuración de capa raster TIF

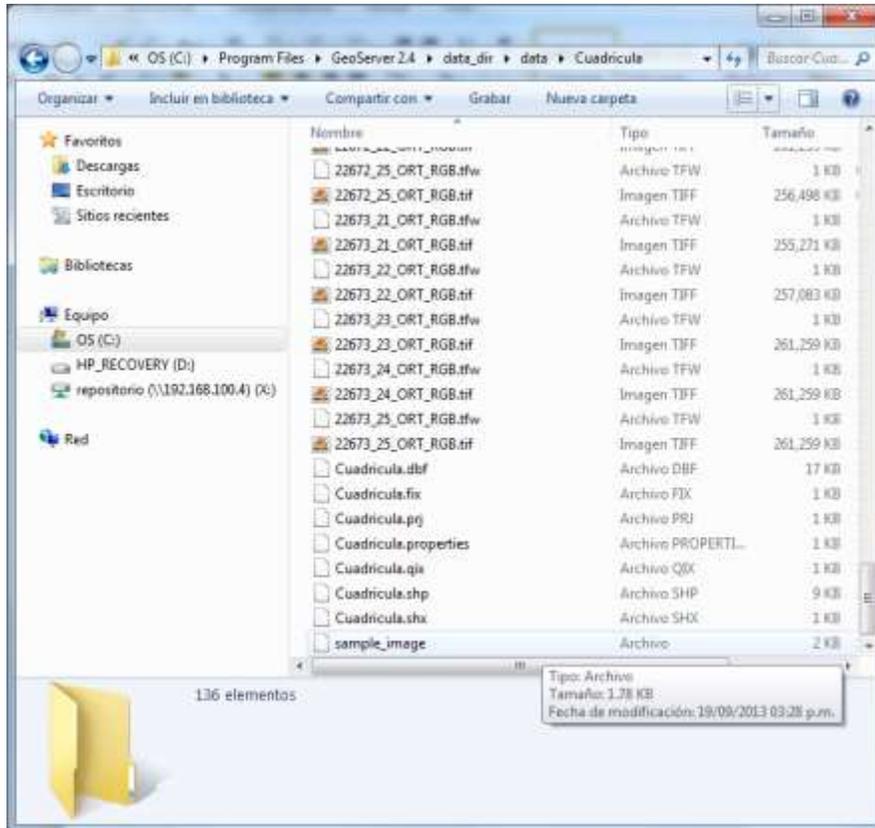


Fuente: elaboración propia.

Luego de escoger la opción Imagemosaic aparece el formulario para agregar origen de datos raster. Se escoge el espacio de trabajo Tesis, se escribe Ortofotos-tif como Nombre y Ortofotos-tif como descripción, en parámetros de conexión se especifica el directorio Cuadrícula que es donde se encuentran las ortofotos que se quieren agregar en la capa espacial. En la figura 40 se puede ver el formulario de edición de origen de datos.

Geoserver genera varios archivos de configuración en el directorio especificado para la capa que se está configurando. Ver figura 42.

Figura 42. Archivos de configuración generados para capa TIF

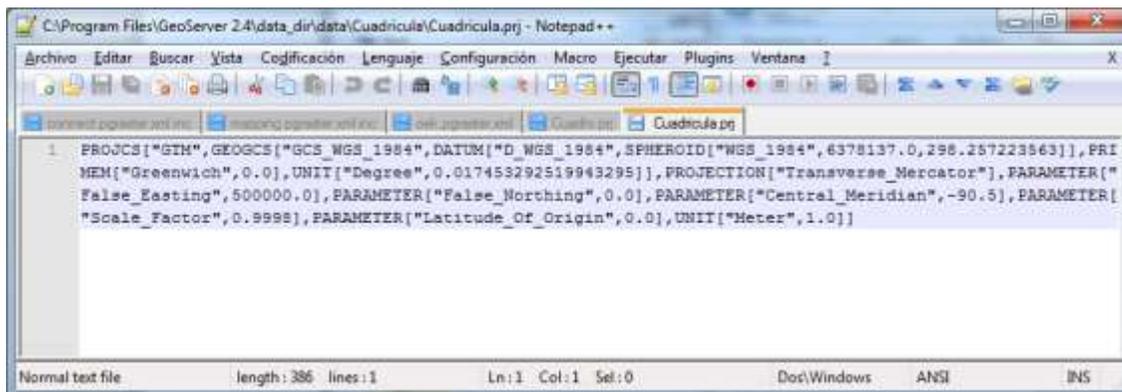


Fuente: elaboración propia.

El contenido del archivo Cuadrícula.prjse se tendrá que modificar con información de referencia espacial GTM. El contenido que genera Geoserver se puede ver en la figura 43.

Luego se modifica el contenido del archivo Cuadrícula.prj con el contenido (configuración de referencia espacial GTM) descargado de la página web. Ver figura 45.

Figura 45. Archivo de configuración capa raster TIF modificado



```
1 PROJCS["GTM",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.017453292519943295]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-90.5],PARAMETER["Scale_Factor",0.9998],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]
```

Fuente: elaboración propia.

Teniendo ya listo la configuración de referencia espacial, se procede a crear la capa espacial en Geoserver. En el menú de opciones se escoge en el apartado Datos la opción Capas, al hacer esto se muestra un listado de capas configuradas y su opción de publicación. Ver figura 46.

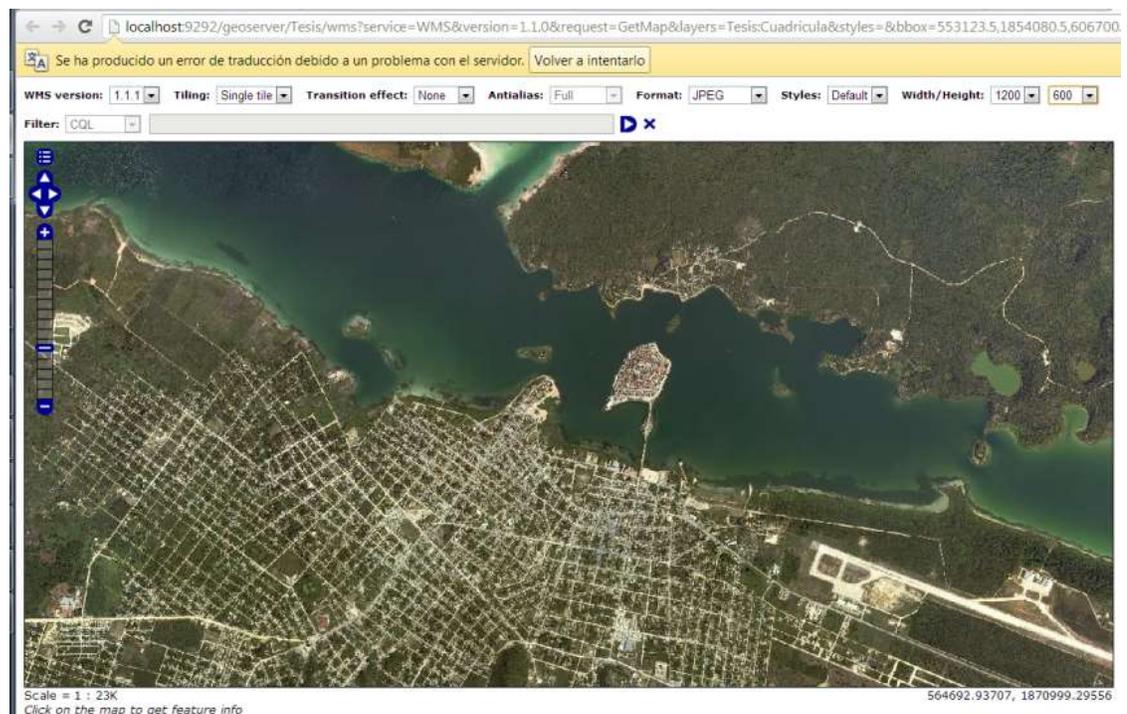
Al escoger la opción de publicación, aparece el formulario con los datos de configuración de la capa. Geoserver detecta la mayoría de los campos, y si todo está bien se activa la opción Guardar, de lo contrario procedemos con modificar los datos de configuración necesarios. Ver figura 48, 49, 50.

Figura 48. Configuración de capa raster TIF (1)



Fuente: elaboración propia.

Figura 52. Visualizar capa raster TIF con OpenLayers



Fuente: elaboración propia.

4.3. Configurar capa en Geoserver con archivos ECW

El portal Geoserver provee un menú de opciones en el lado izquierdo, los pasos a seguir para realizar la configuración de publicación son:

- Crear espacio de trabajo
- Agregar almacén de datos
- Crear capa espacial y publicarla

Los pasos para crear origen de datos es idéntica a la del formato TIF, para el caso de formato ECW se usará también el paquete ImageMosaic. En la figura 53 se observa la interfaz para agregar origen de datos.

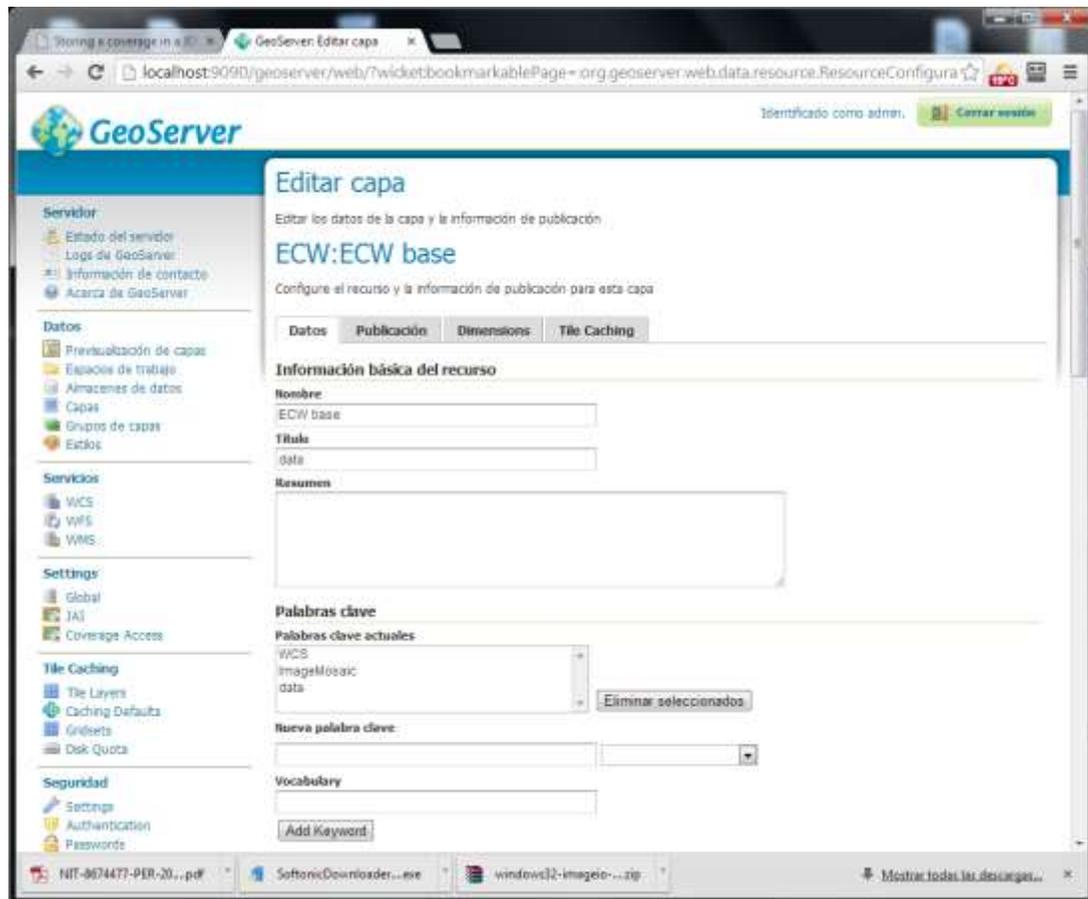
Figura 53. Agregando origen de datos ECW



Fuente: elaboración propia.

En la figura 54 se observa los parámetros de configuración para la capa raster ECW que se está configurando.

Figura 54. Parámetros de configuración para capa raster ECW

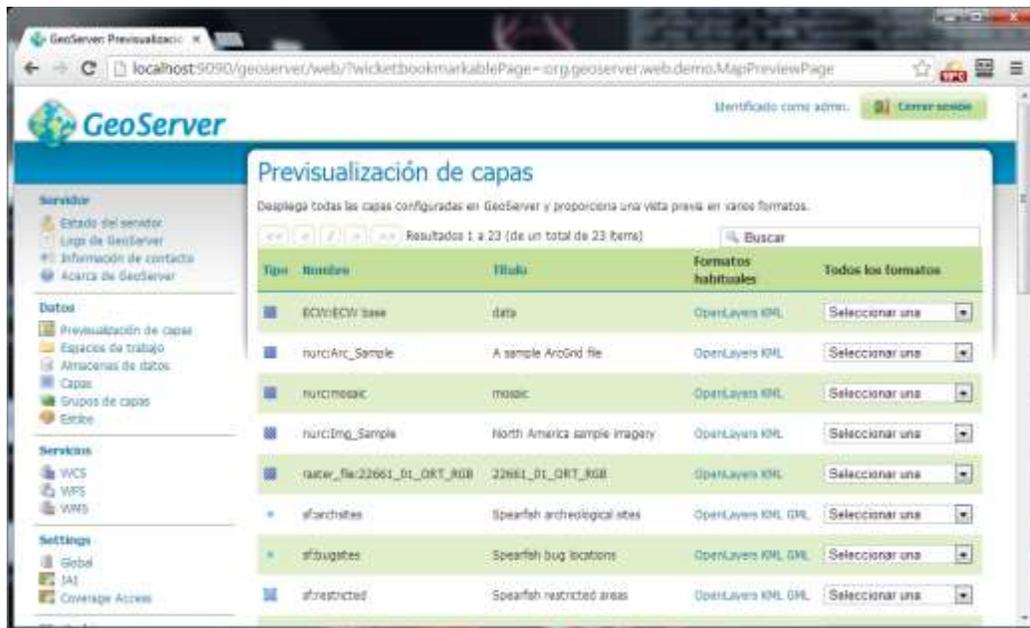


Fuente: elaboración propia.

Para previsualizar la capa configurada, se escoge la opción Previsualización de capas en el menú de opciones de Geoserver.

En la figura 55 se observa el listado de capas configuradas, se previsualiza la capa con título base de ECW:base.

Figura 55. Opción previsualizar capa *raster* ECW



Fuente: elaboración propia.

5. RESULTADO DE ESTUDIO

El estudio se basa en los siguientes parámetros:

- Espacio necesario en disco duro por escenario
- Seguridad en cada escenario
- Performance en cada escenario
- Costo de configuración por cada escenario

5.1. Espacio necesario en disco duro por escenario

Cada escenario ocupa un espacio en disco duro del servidor, se calculará el espacio requerido por cada escenario para encontrar el que ocupa menos espacio en disco duro.

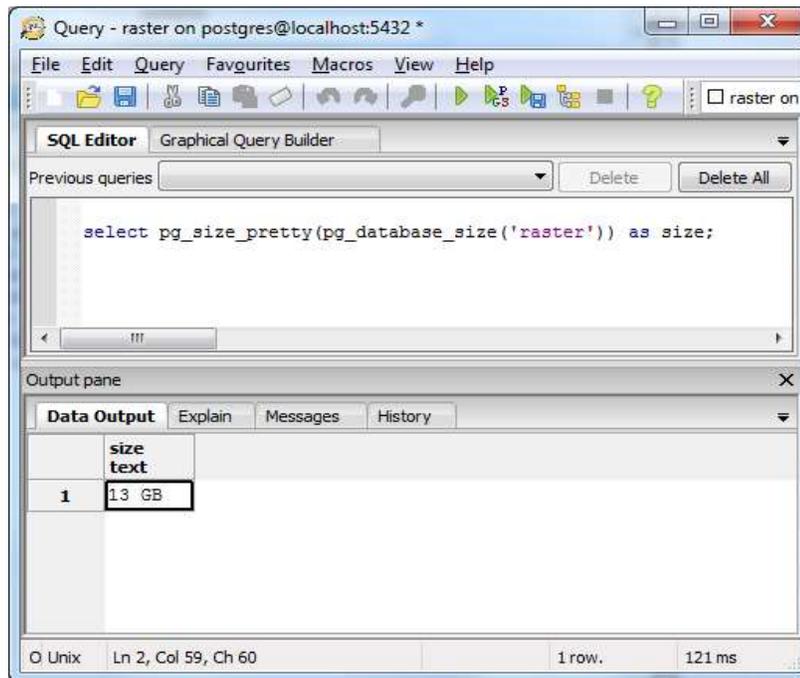
5.1.1. Base de datos PostGIS

Para calcular el espacio que usa la base de datos en disco, se usa la siguiente consulta:

```
selectpg_size_pretty(pg_database_size('raster')) as size;
```

En la figura 56 se observa que la base de datos ocupa 13 GB en disco duro.

Figura 56. Espacio ocupado por base de datos (calculado en PostgreSQL)

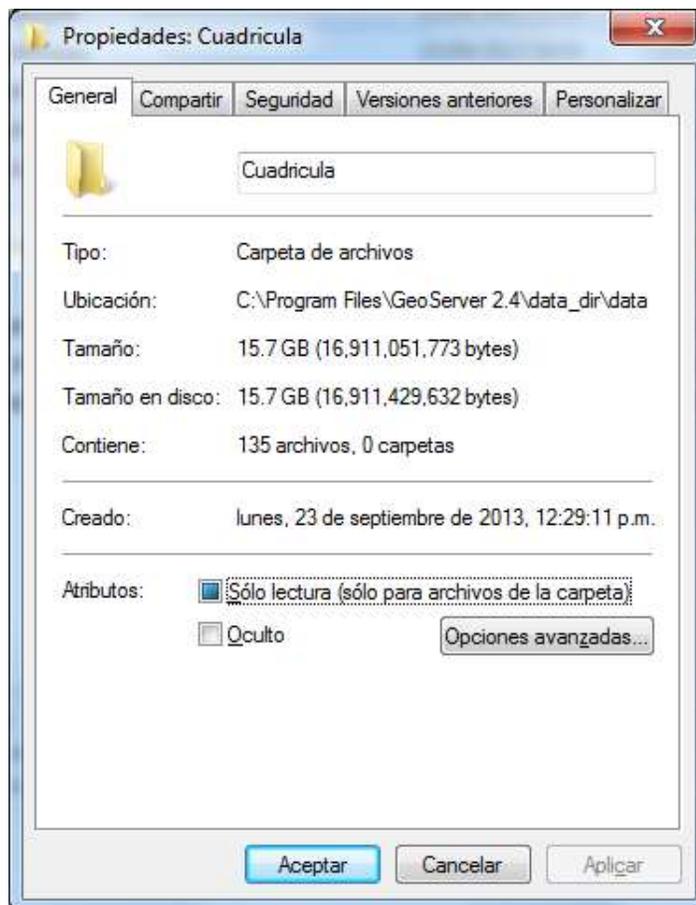


Fuente: elaboración propia.

5.1.2. Ortofotos TIF

Para verificar el espacio utilizado para almacenar las ortofotos TIF en el disco duro, se hace clic derecho en el directorio contenedor de las ortofotos y se selecciona la opción Propiedades. En la figura 57 se ve que las ortofotos ocupan un espacio de 17.7 GB en disco duro.

Figura 57. Espacio requerido por archivos con formato TIF

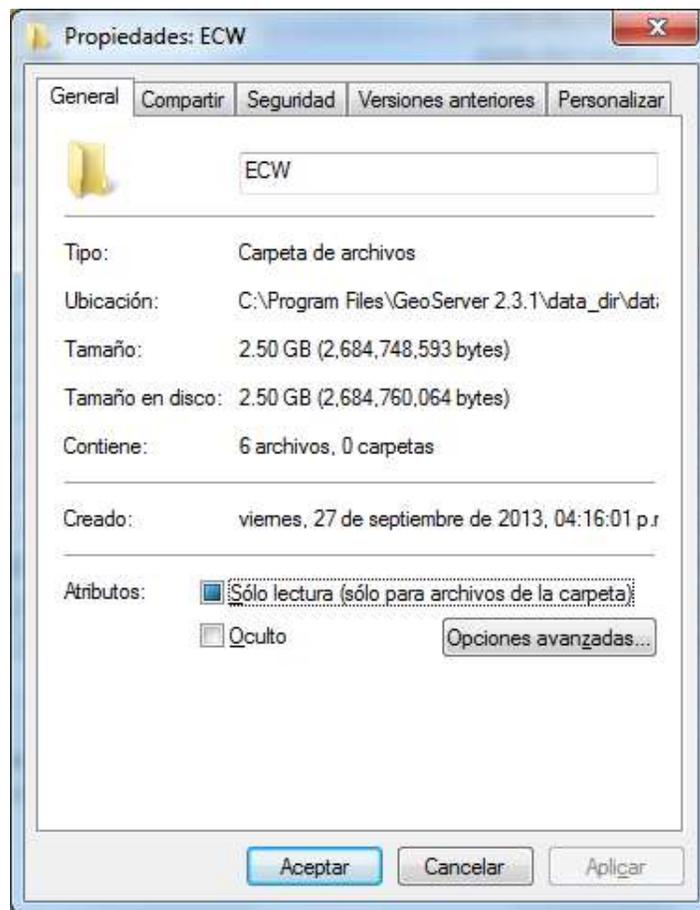


Fuente: elaboración propia.

5.1.3. Archivos ECW

Al igual que en las ortofotos TIF, para verificar el espacio utilizado para almacenar los archivos ECW en disco duro, se hace clic derecho en el directorio contenedor y se selecciona la opción Propiedades. En la figura 58 se ve que el espacio ocupado es de 2.5 GB en disco duro.

Figura 58. Espacio requerido por archivos con formato ECW

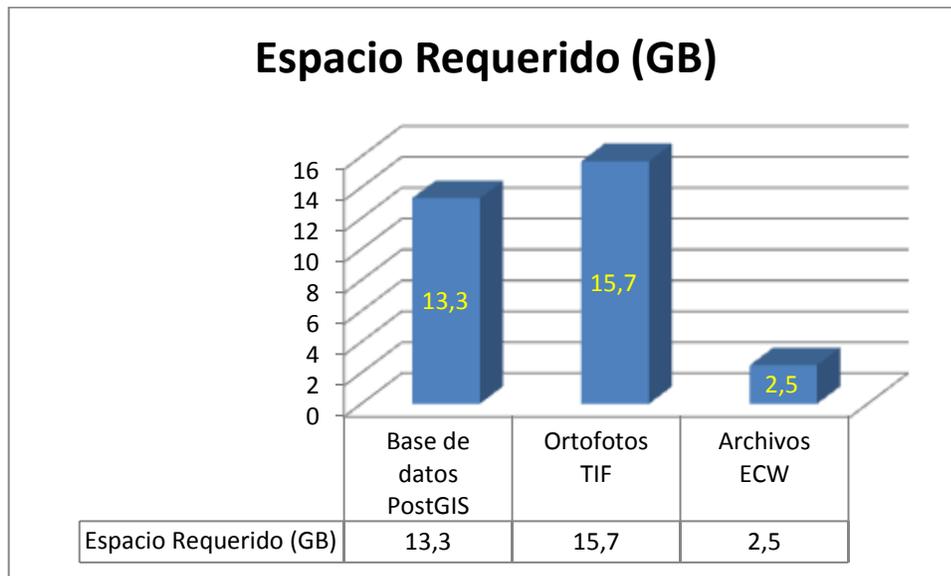


Fuente: elaboración propia.

Comparando los tres resultados, se observa que el escenario que ocupa menor espacio en disco duro, es el de los archivos ECW.

En la figura 59 se observa la gráfica de resultados

Figura 59. Gráfica de resultados (espacio requerido en GB)



Fuente: elaboración propia.

Si solo consideráramos el espacio para escoger el escenario, la mejor opción sería el de los archivos ECW.

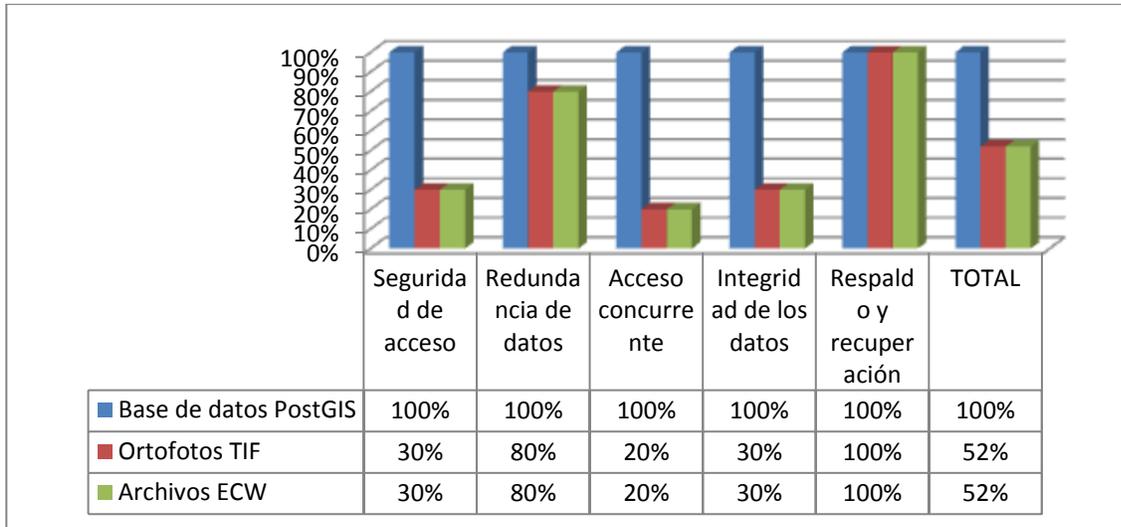
5.2. Seguridad en cada escenario

Existen varios aspectos para poder realizar la comparación de seguridad. La institución que colaboró con el estudio de factibilidad contempla los siguientes aspectos:

- Seguridad de acceso
- Redundancia de datos
- Acceso concurrente
- Integridad de los datos
- Respaldo y recuperación

En la institución el nivel de porcentaje de seguridad en cada escenario es el que se observa en la figura 60.

Figura 60. Nivel de seguridad por escenario (porcentaje)



Fuente: elaboración propia.

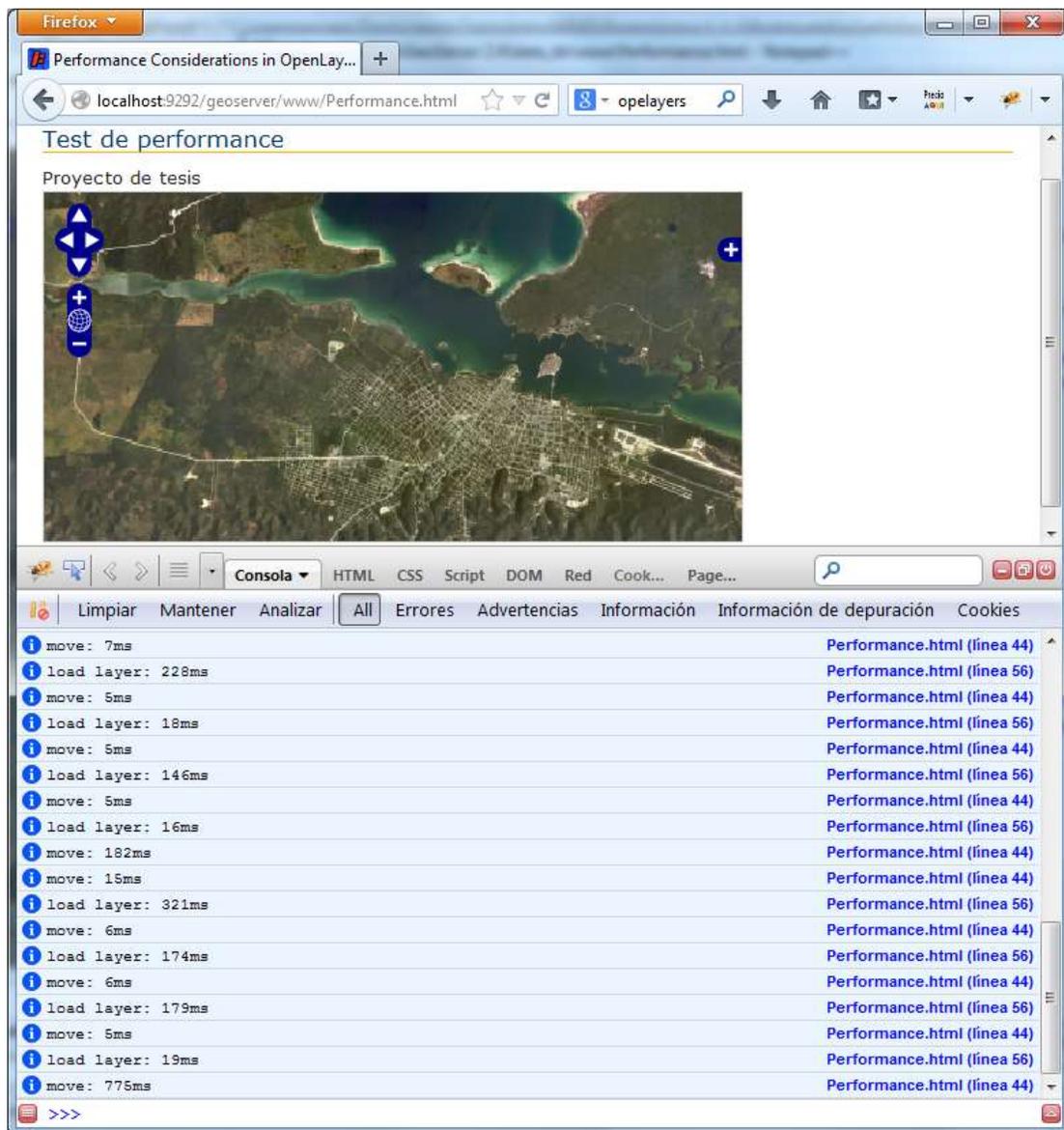
Si de seguridad se trata, el escenario con base de datos PostGIS es la mejor opción a considerar.

5.3. Performance en cada escenario

Para medir los tiempos de respuesta en cada escenario se creó la página web Performance.html (ver anexo C).

En la figura 61 se observa la página web Performance.html en ejecución para el escenario PostGIS.

Figura 61. Página web de tiempos (milisegundos)



Fuente: elaboración propia.

Los datos recabados en cada escenario se muestran en la tabla 1.

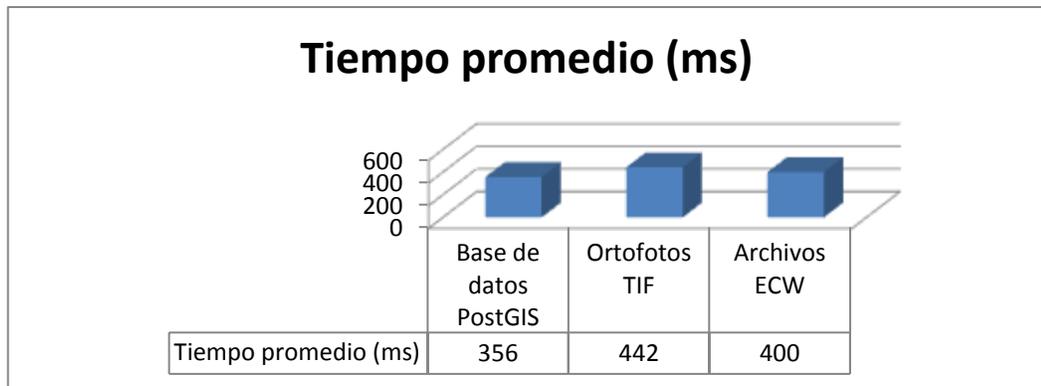
Tabla I. **Tiempos de carga en milisegundos por escenario**

	Base de datos PostGIS (ms)	Ortofotos TIF (ms)	Archivos ECW (ms)
	204	329	315
	403	401	384
	390	705	658
	434	558	601
	654	547	401
	548	564	325
	369	687	389
	337	584	716
	394	622	405
	549	715	315
	507	387	385
	410	585	287
	425	380	658
	373	445	299
	153	341	285
	172	289	409
	376	544	534
	469	355	367
	473	387	297
	456	321	345
	612	344	289
	154	366	245
	216	488	298
	228	205	369
	215	358	408
	146	544	462
	285	231	458
	321	469	387
	241	261	427
	179	258	286
Tiempo promedio	356	442	400

Fuente: elaboración propia.

Los tiempos promedios se observan en la figura 62.

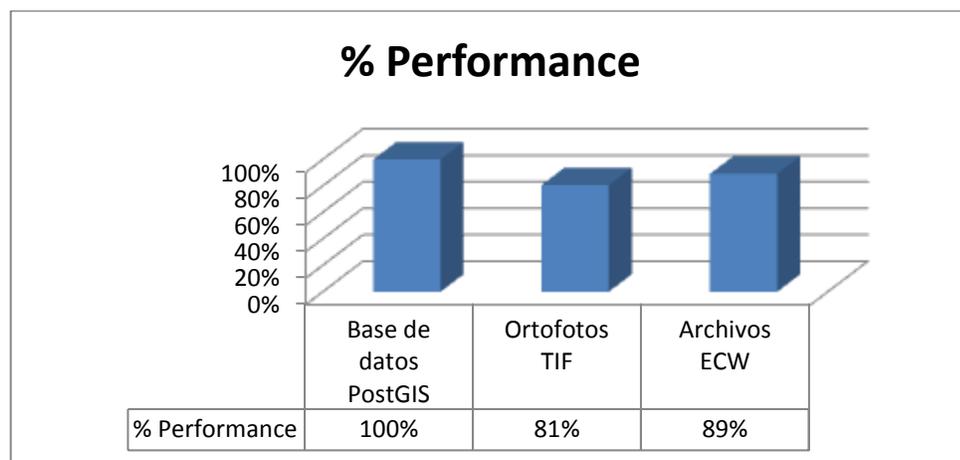
Figura 62. Promedio de tiempos (ms) por escenario



Fuente: elaboración propia.

El mejor resultado promedio es el de 356 milisegundos, en la figura 63 transformamos los resultados en porcentajes.

Figura 63. Nivel de performance por escenario



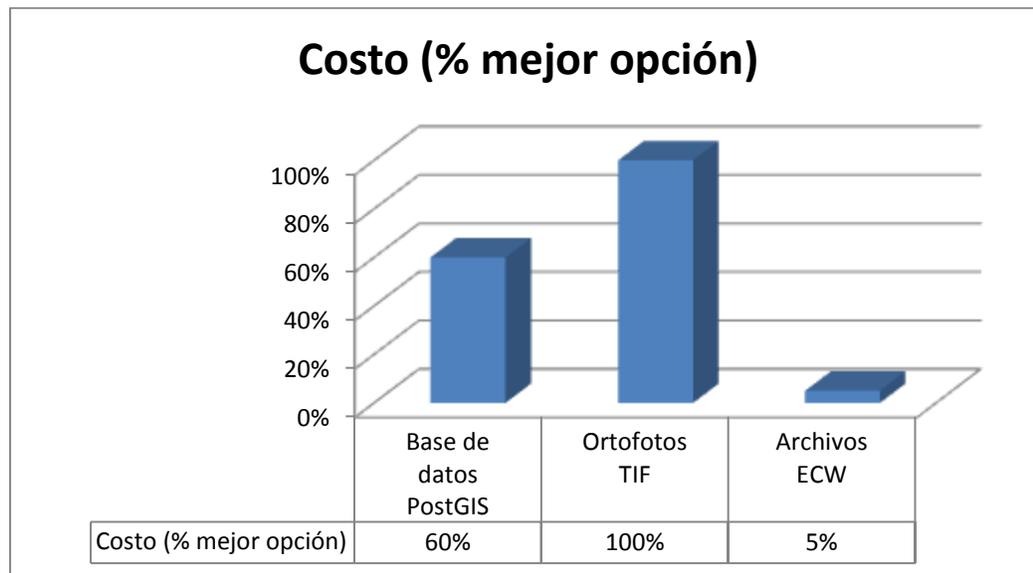
Fuente: elaboración propia.

Los resultados casi son parejos, ya que para los escenarios TIF y ECW se utilizó la opción cache provisto por Geoserver y en la base de datos la información está organizada en capas y se agregaron índices para los registros de las tablas.

5.4. Costo de configuración por cada escenario

Los costos en los que incurriría la institución para cada escenario lo podemos observar en la figura 64. Los porcentajes indican mejor opción en cuanto a precio.

Figura 64. Comparación de costos por escenario



Fuente: elaboración propia.

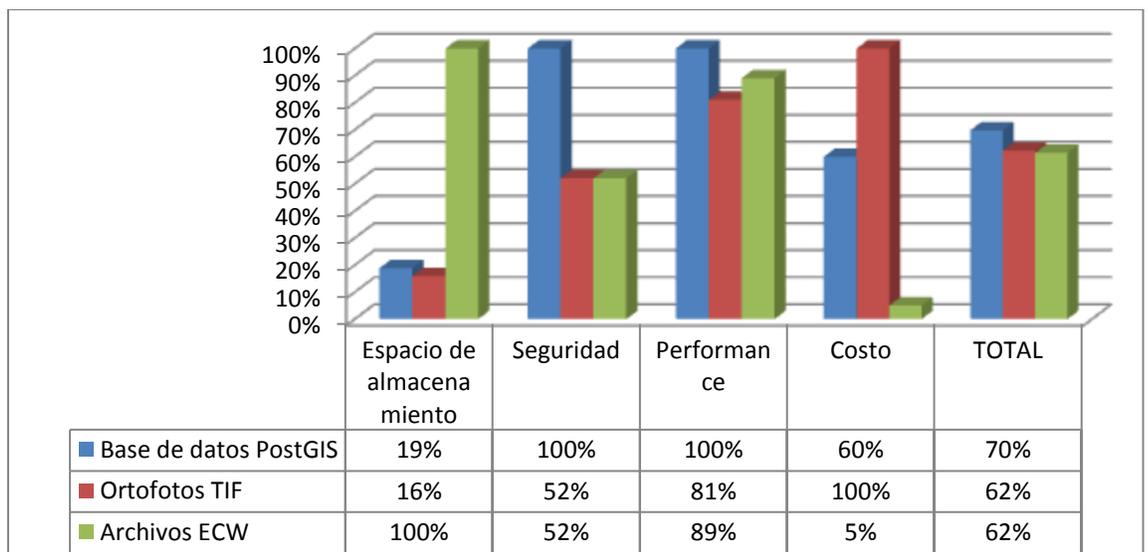
Como se puede ver, el escenario que incurre en menor costo es el escenario de ortofotos TIF (costo 0). Los costos están representados como porcentajes y el mejor es el de 100%.

Los costos están representados en porcentaje, porque la institución que colaboro en el estudio así lo pidió.

5.5. Media de resultados

Sacando el promedio de los resultados de cada escenario obtenemos la gráfica siguiente. Ver figura 65

Figura 65. Media de resultados



Fuente: elaboración propia.

La media de los resultados para los tres escenarios de pruebas indica que la mejor opción es la de tener la información *raster* en la base de datos PostGIS.

CONCLUSIONES

1. Material bibliográfico para el tema de información espacial almacenada en base de datos PostGIS es escasa, ya que el tema es aún nuevo.
2. Los tres escenarios se configuraron con éxito, y de los tres el más fácil de configurar es el escenario de Ortofotos TIF, seguido de Archivos ECW.
3. Tomando en cuenta la seguridad de la información, espacio requerido en disco duro, performance y costo; la media de los resultados nos indica que la opción óptima es la de tener la información en base de datos PostGIS.

RECOMENDACIONES

1. Es necesario contar con equipo robusto (capacidad de procesamiento y memoria) para la realización de una configuración de almacenamiento con información espacial y publicación en un servidor de mapas.
2. Otro escenario de pruebas es publicar el servidor Geoserver en un *hosting* y calificar el performance por internet.

BIBLIOGRAFÍA

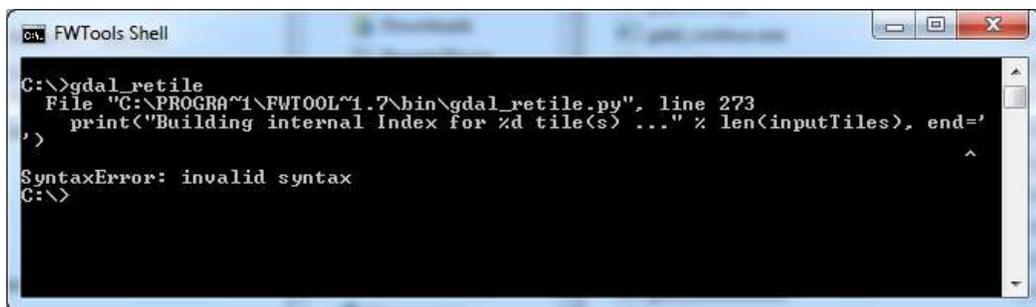
1. MARTÍNEZ LLARIO, José C. *PostGIS 2 análisis espacial avanzado*. 2013. 500 p.
2. OBE, Regina O.; HSU, Leo S. *PostGIS in Action*. Stamford: 2011. 520 p.
3. _____. *PostgreSQL: Up and Running*. O'Reilly Media. United States of America: 2012. 168 p.
4. YOUNGBLOOD, Brian. *Geoserver Beginner's Guide*. PacktPublishing, 2013. 350 p.

ANEXOS

A. Solución de Bug en gdal_retile.py

El archivo gdal_retile.py que viene en la versión 2.4.7 de FWTools tiene un bug. Al ejecutar el comando presenta el error siguiente: ver imagen A1

Figura A1

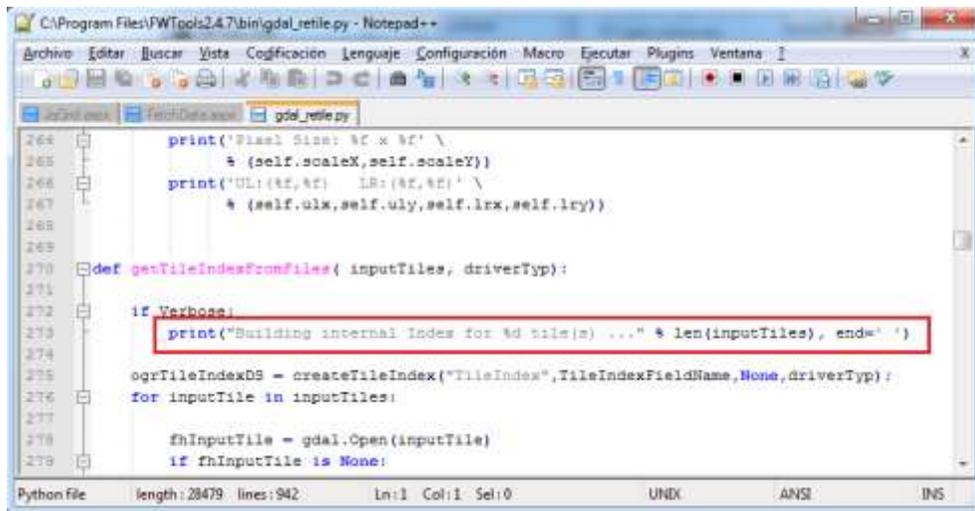


```
C:\>gdal_retile
File "C:\PROGRAMAS\FWTOOLS\1.7\bin\gdal_retile.py", line 273
    print("Building internal Index for %d tile(s) ..." % len(inputTiles), end='
    >
SyntaxError: invalid syntax
C:\>
```

Fuente: elaboración propia.

Para eliminar el error, se necesita editar el archivo gdal_retile.py que se encuentra en el directorio \bin del directorio de instalación de FWTools y buscar la línea de código fuente 273. En la figura A2 se observa la línea de código que tenemos que modificar.

Figura A2



```
C:\Program Files\FWTools2.4.7\bin\gdal_retile.py - Notepad++
Archivo  Editar  Buscar  Vista  Configuración  lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  I

264     print('Pixel Size: %C x %C' \
265           % (self.scaleX,self.scaleY))
266     print('UL: (%E,%E)  LR: (%E,%E)' \
267           % (self.ulx,self.uly,self.lrx,self.lry))
268
269
270     def getTileIndexFromFiles( inputTiles, driverTyp):
271
272     if Verbose:
273         print('Building internal Index for %d tile(s) ...' % len(inputTiles), end=' ')
274
275     ogrTileIndexDS = createTileIndex("TileIndex",TileIndexFieldName,None,driverTyp):
276     for inputTile in inputTiles:
277
278         fhInputTile = gdal.Open(inputTile)
279         if fhInputTile is None:
```

Fuente: elaboración propia.

La instrucción print:

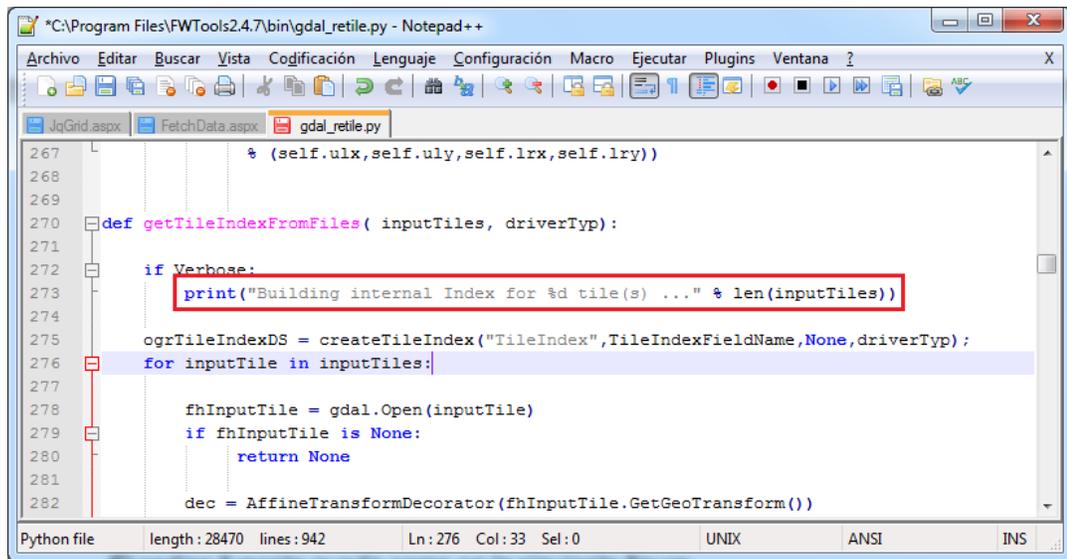
```
print("Building internal Index for %d tile(s) ..." % len(inputTiles), end=' ')
```

Se modifica por:

```
print("Building internal Index for %d tile(s) ..." % len(inputTiles))
```

El código fuente queda como se muestra en la figura A3.

Figura A3



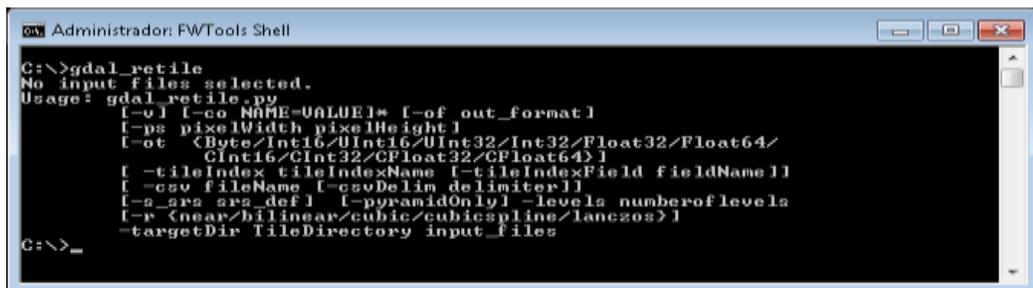
```
*C:\Program Files\FWTools2.4.7\bin\gdal_retile.py - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
gdal_retile.py
267         * (self.ulx,self.uly,self.lrx,self.lry))
268
269
270 def getTileIndexFromFiles( inputTiles, driverTyp):
271
272     if Verbose:
273         print("Building internal Index for %d tile(s) ..." % len(inputTiles))
274
275     ogrTileIndexDS = createTileIndex("TileIndex",TileIndexFieldName,None,driverTyp);
276     for inputTile in inputTiles:
277
278         fhInputTile = gdal.Open(inputTile)
279         if fhInputTile is None:
280             return None
281
282         dec = AffineTransformDecorator(fhInputTile.GetGeoTransform())
```

Python file length: 28470 lines: 942 Ln: 276 Col: 33 Sel: 0 UNIX ANSI INS

Fuente: elaboración propia.

Se vuelve a ejecutar el comando gdal_retile y el resultado es: ver figura A4

Figura A4



```
Administrador: FWTools Shell
C:\>gdal_retile
No input files selected.
Usage: gdal_retile.py
    [-v] [-co NAME=VALUE] [-of out_format]
    [-ps pixelWidth pixelHeight]
    [-ot <Byte/Int16/UInt16/Int32/Float32/Float64/
    CInt16/CInt32/CFloat32/CFloat64>]
    [-tileIndex tileIndexName [-tileIndexField fieldName]]
    [-csv fileName [-csvDelim delimiter]]
    [-s_srs srs def] [-pyramidOnly] [-levels numberOfLevels]
    [-r <near/bilinear/cubic/cubicspline/lanczos>]
    -targetDir TileDirectory input_files
C:\>_
```

Fuente: elaboración propia.

B. Modificación en paquete gt-imagemosaic-jdbc-10-RC2.jar

El paquete gt-imagemosaic-jdbc-RC2.jar es una extensión para Geoserver para poder configurar capas espaciales con información *raster* almacenados en base de datos.

Este paquete lo utilizamos para:

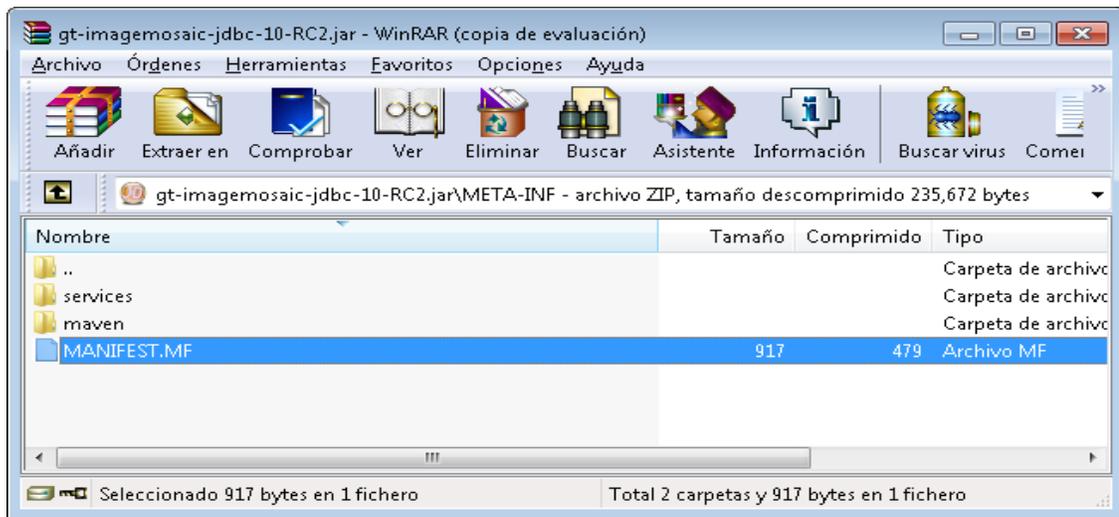
- Crear *scripts* sql con la definición de las tablas que almacenaran información de las capas con información espacial.
- Importar imágenes a la base de datos.
- Configurar la capa espacial *raster* en Geoserver.

Cuando el paquete trataba de conectarse a la base de datos con los parámetros de conexión, se generaba el error `ClassNotFoundException JDBC org.postgresql.Driver`.

Para realizar conexiones a base de datos se usa el paquete `postgresql-9.1-903-jdbc4.rar`, y es este paquete el que no encuentra, y que genera la *exception* anterior.

Luego de verificar las variables de entorno de java en la PC estudio, y el error persistía, se optó por modificar el paquete rar `gt-imagemosaic-jdbc-RC2.jar`. En la figura B1 se puede ver el contenido del paquete rar.

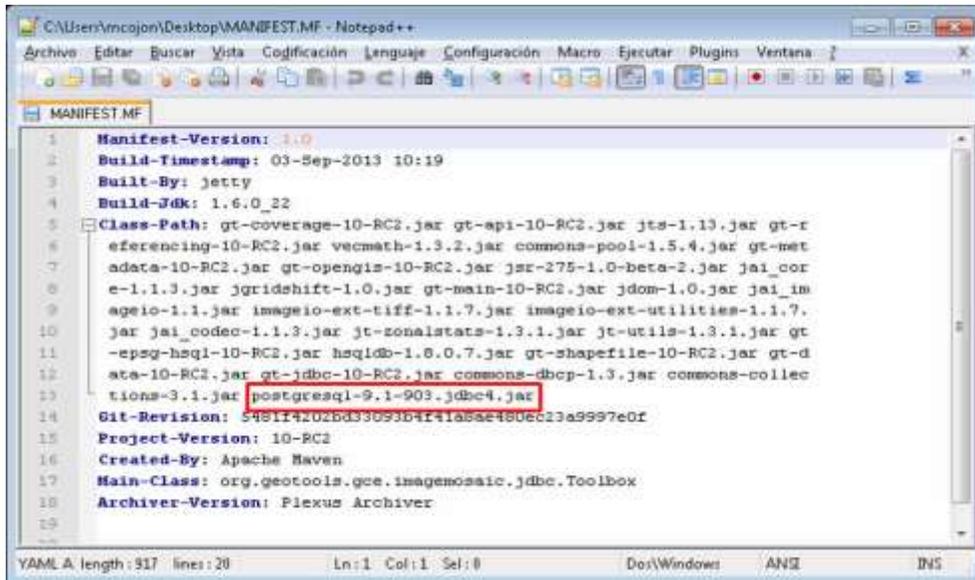
Figura B1



Fuente: elaboración propia.

En el paquete rar se modifica el archivo MANIFEST.MF, y para ello se extrae el archivo. En la figura B2 se puede observar el contenido en el cuadro rojo agregado en el archivo. El cambio en el archivo es para indicar que el paquete postgresql-9.1-903-jdbc4.rar es una dependencia del paquete gt-imagemoaic-jdbc-RC2.jar.

Figura B2



Fuente: elaboración propia.

C. Código fuente de página web para medir performance

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Performance</title>

  <link rel="stylesheet" href="http://openlayers.org/api/2.10/theme/default/style.css" type="text/css"
  />
  <link rel="stylesheet" href="http://openlayers.org/dev/examples/style.css" type="text/css" />
</head>
<body>
  <h1 id="title">Test de performance</h1>
  <div id="tags"></div>
  <div id="shortdesc">Proyecto de tesis</div>
  <div id="map" class="smallmap"></div>
  <div id="docs">
  </div>
  <script type="text/javascript" src="http://openlayers.org/api/OpenLayers.js"></script>
  <script type="text/javascript" src="http://ol-performance.appspot.com/script/OpenLayers-wms-
layerswitcher.js"></script>
  <script type="text/javascript">
    varnumLayers = 0, notLoading = true;

    var bounds = new OpenLayers.Bounds(
      553123.5, 1854083.5,
```

```

        606699.5, 1883991.5
    );
    var map = new OpenLayers.Map('map', {
        projection: "EPSG:90180",
        units: "m",
        maxExtent: bounds,
        maxResolution: 209.28125,
        numZoomLevels: 26,
        theme: null,
        allOverlays: true,
        eventListeners: {
            movestart: function(e) {
                console.time("move");
            },
            moveend: function() {
                console.timeEnd("move");
            },
            preaddlayer: function(e) {
                numLayers++;
                e.layer.events.on({
                    loadstart: function() {
                        notLoading&&console.time("load layer");
                        notLoading = false;
                    },
                    loadend: function() {
                        numLayers--;
                        if (numLayers == 0) {
                            console.timeEnd("load layer");
                            notLoading = true;
                            numLayers = map.layers.length;
                        }
                    }
                });
            }
        }
    });
    var layerNames = ["osm"];
    var layerSwitcher = new OpenLayers.Control.LayerSwitcher({
        setMap: function() {
            var layers = new Array(layerNames.length);
            this.map = {layers: layers};
            for (var i=0, ii=layerNames.length; i<ii; ++i) {
                layers[i] = this.monkeyPatchLayer(layerNames[i]);
            }
        },
        monkeyPatchLayer: function(name) {
            var layers = this.map.layers, layer = {
                name: name,
                displayInLayerSwitcher: true,
                inRange: true,
                visibility: true,
                getVisibility: function() {
                    return layer.visibility;
                },
                setVisibility: function(visibility) {

```

```

        layer.visibility = visibility;
        var newLayers = [];
        for (var i=0, ii=layers.length; i<ii; ++i) {
            layers[i].visibility && newLayers.push(layers[i].name);
        }
        map.layers[0].mergeNewParams({
            LAYERS: newLayers
        });
    }
    };
    return layer;
}
});
map.addControl(layerSwitcher);
layerSwitcher.maximizeControl();
map.addLayer(new OpenLayers.Layer.WMS(
    "medford (group)", "http://localhost:9292/geoserver/Tesis/wms",
    {layers: layerNames, transparent: true},
    {buffer: 0}
));
map.zoomToMaxExtent();
</script>
</body>
</html>

```

Fuente: elaboración propia