

UNIVERSIDAD TÉCNICA DEL NORTE



**Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales**

**IMPLEMENTACIÓN DE IMÁGENES RASTER DENTRO DEL GEO PORTAL DE
LA UNIVERSIDAD TÉCNICA DEL NORTE PARA LA GESTIÓN DE LOS
SERVICIOS ESPACIALES**

**Trabajo de grado previo a la obtención del título de Ingeniero en Sistemas
Computacionales**

Autor:

Cristoper Alexander Toapanta Rodríguez

Director:

Ing. Pablo Andrés Landeta López

Ibarra - Ecuador

Marzo 2020



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO			
CÉDULA DE IDENTIDAD:	1003836721		
APELLIDOS Y NOMBRES:	TOAPANTA RODRIGUEZ CRISTOPHER ALEXANDER		
DIRECCIÓN:	SAN ANTONIO DE IBARRA – BARRIO PUCAHUAICO, DIRECCION AL CONVENTO “MONSEÑOR LEONIDAS PROAÑO”		
EMAIL:	catoapantar@utn.edu.ec		
TELÉFONO FIJO:	062-550-420	TELÉFONO MÓVIL:	0989198028

DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DE IMÁGENES RASTER DENTRO DEL GEO PORTAL DE LA UNIVERSIDAD TÉCNICA DEL NORTE PARA LA GESTIÓN DE LOS SERVICIOS ESPACIALES
AUTOR (ES):	TOAPANTA RODRÍGUEZ CRISTOPER ALEXANDER
FECHA: DD/MM/AAAA	05/03/2020
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	MSc. PABLO LANDETA

2. CONSTANCIA

CONSTANCIA

El autor(es) manifiesta(n) que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular(es) de los derechos patrimoniales, por lo tanto, que asume(n) la responsabilidad sobre el contenido de la misma y saldrá(n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los cinco días del mes de marzo de 2020

EL AUTOR:



CRISTOPER ALEXANDER TOAPANTA RODRÍGUEZ

1003836721

CERTIFICACIÓN DEL DIRECTOR



UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DIRECTOR

Por medio de la presente, yo MSc. Pablo Andrés Landeta López, certifico que el Sr. **CRISTOPHER ALEXANDER TOAPANTA RODRÍGUEZ**, portador de la cédula de identidad Nro. 1003836721, ha trabajado en el desarrollo del proyecto de grado **"IMPLEMENTACIÓN DE IMÁGENES RASTER DENTRO DEL GEO PORTAL DE LA UNIVERSIDAD TÉCNICA DEL NORTE PARA LA GESTIÓN DE LOS SERVICIOS ESPACIALES"**, previo a la obtención del título de Ingeniero en Sistemas Computacionales, realizando con interés profesional y responsabilidad, que verifico en honor a la verdad.

Atentamente,

A handwritten signature in blue ink, appearing to be "P. Landeta", is written over a horizontal dotted line.

MSc Pablo Landeta

DIRECTOR DEL TRABAJO DE GRADO

AUTORÍA

AUTORÍA

Yo, CRISTOPER ALEXANDER TOAPANTA RODRÍGUEZ, portador de la cédula de ciudadanía número 1003836721, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, **IMPLEMENTACIÓN DE IMÁGENES RASTER DENTRO DEL GEO PORTAL DE LA UNIVERSIDAD TÉCNICA DEL NORTE PARA LA GESTIÓN DE LOS SERVICIOS ESPACIALES**, que no ha sido previamente presentada para ningún grado, ni calificación profesional, y que se ha respetado las diferentes fuentes y referencias.

A handwritten signature in blue ink, appearing to read 'Cristoper', with a stylized flourish underneath.

CRISTOPHER ALEXANDER TOAPANTA RODRÍGUEZ

C.I. 1003836721

DEDICATORIA

El presente trabajo va dedicado:

A Dios, por ser el principal guía durante toda la etapa universitaria, dándome la fuerza necesaria para superar los obstáculos presentados y por haber escuchado mis oraciones, es él quien nunca abandona y extiende una mano en los momentos que menos se espera.

A mis padres y hermanos, por siempre confiar en mí y darme ánimos durante cada una de las etapas complicadas de la vida universitaria, siendo ellos la principal inspiración para ser alguien en la vida.

"He peleado la buena batalla, he terminado la carrera, me he mantenido en la fe"

2 Timoteo 4:7

AGRADECIMIENTOS

A Dios Padre, Jesucristo, fiel amigo, protector, luz y guía en los momentos más complicados de la vida

A mi madre, Esthela Rodríguez, por darme la vida, entregarme su amor, preocupación y cariño, que me supo guiar en el camino adecuado y estuvo de pie en los momentos difíciles gracias por mantener a la familia unida.

A mi padre, Carlos Toapanta, quien supo instruirme desde pequeño y me demostró que la mejor forma de ser alguien en la vida es: sin nunca rendirse y saber superar cada uno de los obstáculos que se presentan en el diario vivir.

A mi hermano mayor Carlos, que siempre me soporto y fue mi ejemplo a seguir desde la niñez, gracias por escucharme en esos momentos en los que no sabía qué hacer y por ser siempre el impulso energético y anímico en mi vida académica.

A mi hermana menor, Pamela, con quien compartí momentos de estudio hasta altas horas de la noche, compañía perfecta para minutos de relajación y consejos inesperados de hermana menor a hermano mayor durante toda la carrera.

A mis tías, Teresa, Kethy y Karina Rodríguez, por ser las principales artífices de mi amor por el estudio y la Universidad Técnica del Norte y a aquellos familiares, que supieron estar pendientes de mi proceso formativo como apoyo esencial durante momentos de necesidad y unión.

A mis compañeros de equipo: Bryan, Kevin, Richard, Nelson, Franklin, compañeros de una y mil batallas, con quienes compartí momentos buenos y momentos complicados, dentro y fuera del aula, conocedores de la realidad universitaria.

A todas las personas que Dios supo poner en mi vida para cambiar mi actitud y aprender a madurar en diversos aspectos esenciales para la mejora personal y académica.

Finalmente, agradezco a mi ingeniero tutor MSc. Pablo Landeta por su paciencia, guía, enseñanza y apoyo durante el proceso de la realización del proyecto presente.

ÍNDICE DE CONTENIDOS

1. IDENTIFICACIÓN DE LA OBRA	II
2. CONSTANCIAS.....	III
CERTIFICACIÓN DIRECTOR.....	¡Error! Marcador no definido.
AUTORÍA.....	IV
DEDICATORIA.....	VI
AGRADECIMIENTOS.....	VII
ÍNDICE DE CONTENIDOS	VIII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS	XII
RESUMEN	XIII
ABSTRACT.....	XIV
INTRODUCCIÓN.....	XV
A. PARTE INTRODUCTORIA	XV
A.1. PROBLEMA.....	XV
A.2. OBJETIVOS	XVI
A.2.1. OBJETIVO GENERAL	XVI
A.2.2. OBJETIVOS ESPECÍFICOS.....	XVI
A.3. ALCANCE	XVI
A.4. JUSTIFICACIÓN.....	XVIII
CAPÍTULO I.....	1
1. MARCO TEÓRICO.....	1
1.1. Infraestructura de Datos Espaciales.....	1
1.2. Sistemas de Información Geográfica (SIG).....	2
1.2.1. Conceptos: ¿Qué es un Sistema de Información Geográfica?	2
1.2.2. Características	3
1.3. Componentes de un GIS.....	4
1.3.1. Base de Datos: PostgreSQL con PostGIS	4
1.3.2. Gestión de Capas: ArcMap y QGIS.....	7
1.3.3. Servidor de Mapas: MapServer.....	10
1.3.4. Visor de Mapas: P. mapper.....	12
1.3.5. Datos GIS: Vectorial vs Raster.....	14
1.3.6. Codificación: Yii Framework.....	16
1.4. Imágenes Raster.....	17
1.4.1. Usos Espaciales	17

1.4.2.	Georreferenciación	19
1.5.	Estándar: Open Geospatial Consortium	19
1.5.1.	Definición	19
1.5.2.	Servicios Geoespaciales	19
1.5.3.	Web Coverage Services.....	20
1.6.	Metodología de Desarrollo: XP.....	21
1.6.1.	Componentes XP.....	21
1.6.2.	Ciclo de vida XP.....	21
CAPÍTULO II.....		23
2.	DESARROLLO.....	23
2.1.	Fase de Exploración: Análisis.....	24
2.1.1.	Especificación de Tecnologías	24
2.1.2.	Cartillas de Historias de Usuario	25
2.2.	Fase de Planificación: Diseño	31
2.2.1.	Planificación de Iteraciones	31
2.2.2.	Tarjetas CRC.....	31
2.2.3.	Diagramas UML: Casos de Uso.....	32
2.2.4.	Diseño Simple: Arquitectura.....	34
2.3.	Fase de Iteraciones: Desarrollo	35
2.3.1.	Iteración N° 1	36
2.3.2.	Iteración N° 2.....	38
2.3.3.	Iteración N° 3.....	39
2.4.	Generación de Servicios WCS.....	40
2.5.	Fase de Pruebas.....	41
2.5.1.	Pruebas: Escenario de Carga y Gestión de Archivos Raster	41
2.5.2.	Pruebas: Escenario de Creación y Gestión de Menú Raster	42
2.5.3.	Pruebas: Escenario de Muestra de la Información	44
CAPÍTULO III		45
3.	RESULTADOS.....	45
3.1.	Validación de Resultados.....	45
3.2.1.	Pruebas de Rendimiento: J-Meter	45
3.2.2.	Escenarios para el análisis.....	46
3.2.3.	Datos Obtenidos	49
3.2.4.	Análisis de Resultados.....	50
CONCLUSIONES.....		57
RECOMENDACIONES.....		58

REFERENCIAS	59
-------------------	----

ÍNDICE DE FIGURAS

Fig. 1: Arquitectura del Módulo de Gestión Ráster	XVII
Fig. 2: Factores de mejora continua	3
Fig. 3: Tipos de datos gestionados PostGIS.....	6
Fig. 4: Arquitectura de funcionamiento de PostGIS	7
Fig. 5: Tabla de contenidos en ArcMap	8
Fig. 6: Catalogo de objetos en ArcMap.....	8
Fig. 7: Interfaz Principal de QGIS	9
Fig. 8: Capa codificada en un archivo MAP	10
Fig. 9: Interfaz de Geo Server	11
Fig. 10: Visor de mapas p. mapper (menú y visor)	13
Fig. 11: Ejemplo de capa ráster.....	13
Fig. 12: Tipos de datos que se procesan en un GIS	14
Fig. 13: Ejemplos de capas vectoriales	15
Fig. 14: Estructura de una aplicación con Yii.....	16
Fig. 15: Ortofoto base con capa de puntos.....	17
Fig. 16: Modelo Digital del Terreno.....	18
Fig. 17: Digitalización de un mapa post análisis	18
Fig. 18: Arquitectura Web Coverage Service.....	20
Fig. 19: Ciclo de vida de la metodología XP	22
Fig. 20: Arquitectura de funcionamiento de PHP	24
Fig. 21: Caso de Uso 1 – Funciones del Módulo de Carga.....	32
Fig. 22: Caso de Uso 2 – Escenario: Carga de Capas	32
Fig. 23: Caso de Uso 2 – Escenario: Edición de Información	33
Fig. 24: Caso de Uso 3 – Escenario: Creación de Menús.....	33
Fig. 25: Caso de Uso 4 – Escenario: Edición de un Menú.....	33
Fig. 26: Caso de Uso 5 – Escenario: Eliminación	34
Fig. 27: Arquitectura del Prototipo	35
Fig. 28: Pantalla principal del generador Gii	35
Fig. 29: Arquitectura CRUD generada por Gii.....	36
Fig. 30: Formulario de Carga y sus componentes	36
Fig. 31: Repositorio de Capas de MapServer	37
Fig. 32: Pantalla de Edición de Etiqueta	37

Fig. 33: Pantalla para la creación de Menús.....	38
Fig. 34: Pantallas de Edición para nodos padre e hijo	38
Fig. 35: Pantalla inicial del visor de mapas	39
Fig. 36: Ejemplo de capa en el archivo MAP	39
Fig. 37: Nodo padre con un nodo hijo asignado (submenú).....	40
Fig. 38: Nodo hijo con una capa asignada.....	40
Fig. 39: Escritura para una capa subida	40
Fig. 40: Arquitectura inicial de un Proyecto J-Meter.....	45
Fig. 41: Petición HTTP para Capas Ligeras.....	46
Fig. 42: Petición HTTP para Capas Medianas.....	47
Fig. 43: Petición HTTP para Capas Pesadas	48
Fig. 44: Matriz de Correlación General	51
Fig. 45: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)	52
Fig. 46: Resultados y Gráfica de Correlación para Muestra-Error.....	53
Fig. 47: Resultados y Gráfica de Correlación para Muestra-Rendimiento.....	53
Fig. 48: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)	54
Fig. 49: Resultados y Gráfica de Correlación para Muestra - Error.....	54
Fig. 50: Resultados y Gráfica de Correlación para Usuarios-Rendimiento.....	55
Fig. 51: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)	55
Fig. 52: Resultados y Gráfica de Correlación para Muestra - Error.....	56
Fig. 53: Resultados y Gráfica de Correlación para Muestra - Rendimiento.....	56

ÍNDICE DE TABLAS

TABLA 1: VENTAJAS Y DESVENTAJAS DE POSTGRESQL	5
TABLA 2: COMPARATIVA ENTRE SERVIDORES DE MAPAS	12
TABLA 3: COMPARATIVA ENTRE TIPO DE DATOS	15
TABLA 4: OBJETOS DEL FRAMEWORK YII.....	16
TABLA 5: HISTORIA DE USUARIO N° 1	25
TABLA 6: HISTORIA DE USUARIO N° 2.....	25
TABLA 7: HISTORIA DE USUARIO N° 3.....	26
TABLA 8: HISTORIA DE USUARIO N° 4.....	26
TABLA 9: HISTORIA DE USUARIO N° 5.....	27
TABLA 10: HISTORIA DE USUARIO N° 6.....	27
TABLA 11: HISTORIA DE USUARIO N° 7.....	28
TABLA 12: HISTORIA DE USUARIO N° 8.....	28
TABLA 13: HISTORIA DE USUARIO N° 9.....	29
TABLA 14: HISTORIA DE USUARIO N° 10.....	29
TABLA 15: HISTORIA DE USUARIO N° 11.....	30
TABLA 16: HISTORIA DE USUARIO N° 12.....	30
TABLA 17: PLANIFICACIÓN INICIAL DE ITERACIONES	31
TABLA 18: TARJETA CRC DE LOS OBJETOS CAPAS RÁSTER.....	31
TABLA 19: TARJETA CRC DE LOS OBJETOS MENÚ RÁSTER	32
TABLA 20: PRUEBA DE FUNCIONAMIENTO DE CARGA DE CAPAS.....	41
TABLA 21: PRUEBA DE FUNCIONAMIENTO DE ELIMINACIÓN DE CAPAS	42
TABLA 22: PRUEBA DE FUNCIONAMIENTO DE CREACIÓN DE MENÚS	42
TABLA 23: PRUEBA DE FUNCIONAMIENTO DE EDICIÓN DE MENÚS.....	43
TABLA 24: PRUEBA DE FUNCIONAMIENTO PARA ELIMINAR MENÚ	43
TABLA 25: PRUEBA DE FUNCIONAMIENTO - DATOS GEOGRÁFICOS.....	44
TABLA 26: PARÁMETROS DE PRUEBA PARA LA PETICIÓN N° 1.....	46
TABLA 27: PARÁMETROS DE PRUEBA PARA LA PETICIÓN N° 2.....	47
TABLA 28: PARÁMETROS DE PRUEBA PARA LA PETICIÓN N° 3.....	48
TABLA 29: VALORES OBTENIDOS PARA CAPAS LIGERAS	49
TABLA 30: VALORES OBTENIDOS PARA CAPAS MEDIANAS.....	49
TABLA 31: VALORES OBTENIDOS PARA CAPAS PESADAS	50
TABLA 32: PRUEBAS DE NORMALIDAD	51

RESUMEN

El presente trabajo de grado se encuentra enfocado a mejorar los servicios espaciales que ofrece la plataforma web de tipo Infraestructura de Datos Espaciales (IDE) "Geo portal UTN", mediante el diseño de un módulo para gestionar imágenes ráster que permitan al usuario obtener una mejor visión de los sectores mostrados en un mapa, pertenecientes a la aplicación web y por ende asegurar un mejor servicio como plataforma web para empresas que se encargan de trabajar con temas relacionados a catastros y sectorización.

Como parte inicial del documento se tiene la Introducción, la cual va enfocada a analizar la necesidad de tener un módulo para gestionar imágenes ráster dentro de la IDE y cuáles son las metas u objetivos que se desea alcanzar con el desarrollo de dicho módulo. En esta sección se hace una referencia al punto que debe llegar el desarrollo del módulo y hacia qué objetivos de desarrollo sustentable se encuentra enfocado el proyecto.

Para el Capítulo I de la documentación se detalla la parte del Marco Teórico en la cual se toma en cuenta los temas relacionados a Sistemas de Información Geográfica (GIS), sus componentes, herramientas de gestión de datos espaciales y componentes para el desarrollo del módulo de software.

En el Capítulo II del presente proyecto se detalla la parte de desarrollo de software basado en una metodología para la creación de sistemas web, en la cual se define las tareas primarias, secundarias, el proceso con el que se lleva a cabo la codificación, además de la corrección de errores, pruebas y resultados del prototipo o producto final.

Para la parte final del documento, en el Capítulo III se realizaron pruebas de rendimiento con el fin de comprobar el funcionamiento del módulo de carga en un entorno simulado, enfocado a verificar los tiempos de respuesta según una petición web dirigida a las funciones de carga, además de la corrección de errores y mejora, el proceso para la validación de los datos obtenidos, se realizó mediante técnicas de análisis estadístico que permita conocer la relación entre los valores de funcionamiento obtenidos y los pesos de las capas que se solicitan en la petición web diseñada.

ABSTRACT

The present degree work is focused on improving the space services offered by the Spatial Data Infrastructure (IDE) web platform, "Geo portal UTN", by designing a module to manage raster images that allow the user to obtain a better vision of the sectors shown on a map, belonging to the web application and therefore ensure a better service as a web platform for companies that are responsible for working with issues related to cadasters and sectorization.

As an initial part of the document is the Introduction, which focuses on analyzing the need to have a module to manage raster images within the IDE and what are the goals or objectives to be achieved with the development of said module. In this section a reference is made to the point that the development of the module should reach and to which sustainable development goals the project is focused.

Chapter I of the documentation details the part of the Theoretical Framework in which the issues related to Geographic Information Systems (GIS), its components, spatial data management tools and components for the development of the module of software.

Chapter II of this project details the software development part based on a methodology for the creation of web systems, in which the primary, secondary tasks, the process with which the coding is carried out is defined, in addition of the correction of errors, tests and results of the prototype or final product.

For the final part of the document, in Chapter III performance tests were performed in order to verify the operation of the load module in a simulated environment, focused on verifying response times according to a web request addressed to the load functions, In addition to the correction of errors and improvement, the process for the validation of the data obtained, was carried out by means of statistical analysis techniques that allow to know the relationship between the obtained operating values and the weights of the layers that are requested in the web request designed.

INTRODUCCIÓN

A. PARTE INTRODUCTORIA

A.1. PROBLEMA

Como herramientas de gestión de información mediante mapas, los Sistemas de Información Geográfica (GIS) han evolucionado gracias a las nuevas extensiones que permiten interactuar a un sistema de este tipo con una base de datos, se puede decir que gracias a la capacidad de los GIS para gestionar información gráfica y alfanumérica estos sistemas se han vuelto importantes, dando paso la aparición de aplicaciones GIS más avanzadas.

“Los Sistemas de Información Geográfica (GIS) se están imponiendo a un ritmo acelerado como herramientas de gestión de los abastecimientos urbanos. Su capacidad para manejar la información gráfica y alfanumérica a través de un gestor de Base de Datos hace de los GIS una herramienta muy útil tanto en la planificación como en la gestión diaria.”
(Universitat Politècnica de Valencia, 2018)

Con la creación de los Geo portales se hizo común el dar un mejor uso de los mapas como servicios de información, mismos que tienen su utilidad en diversos campos empresariales, como, por ejemplo, entidades encargadas de la sectorización, catastros, análisis ambiental automatizado, etc. La versatilidad de estos sistemas permite mejorar el manejo de información mediante la aplicación de objetos geográficos que incrementen el aprovechamiento de esta.

“La razón fundamental para utilizar un GIS 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, y 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.” (Langle, 2012)

Para dar servicio a la comunidad, la Universidad Técnica del Norte cuenta con un GIS denominado “Geo Portal UTN”, el cual tiene como objetivo, la gestión, publicación y procesamiento de información geoestadística basada en mapas. El visualizador de mapas del Geo portal utiliza como capa base una Interfaz de Programación de Aplicaciones (API) proporcionada por la empresa Google, basada en las funciones de Google Satélite, Google Streets, Google Terrain, y Open Street Map. Este visor permite la visualización únicamente

de imágenes satelitales que pierden definición al momento de realizar un zoom a ciertos sectores de los que se desee obtener la información.

La Infraestructura de Datos Espaciales (IDE) UTN ha sido desarrollada con el fin de proporcionar información espacial y alfanumérica de calidad para la comunidad interna y externa, y para futuros proyectos de la institución en sí, con lo que se busca proporcionar un servicio eficaz, eficiente y preciso al momento de proporcionar datos geoespaciales.

“La IDE (Infraestructura de datos Espaciales) es un sistema informático integrado por un conjunto de servicios (catálogos, visor y editor de mapas), con tecnología de software libre, que ayudan a gestionar la Información Geográfica (mapas, imágenes satelitales, obtención de datos estadísticos y otros), disponibles en internet, que cumplen una serie de condiciones de interoperabilidad (normas, especificaciones, protocolos, interfaces) que son regidas por el Consorcio Ecuatoriano para la difusión del Internet avanzado” (Universidad Técnica del Norte, 2018)

A.2. OBJETIVOS

A.2.1. OBJETIVO GENERAL

Desarrollo de un módulo que permita la carga y gestión de imágenes ráster dentro del Geo Portal de la Universidad Técnica del Norte con el fin de agregar funcionalidades a sus servicios geoespaciales, basándose en el estándar WCS (Web Coverage Services) del OGC (Open Geospatial Consortium), para objetos ráster.

A.2.2. OBJETIVOS ESPECÍFICOS

- a) Establecimiento de un marco teórico que permita conocer el proceso de gestión de imágenes ráster dentro de una Infraestructura de Datos Espaciales.
- b) Implementación de un módulo de software que permita la carga y gestión de imágenes ráster dentro del Geo Portal UTN.
- c) Validación de los resultados

A.3. ALCANCE

El presente proyecto tiene como objetivo desarrollar un módulo de Gestión Raster que permita el ingreso de una capa ráster a la base de datos espacial, de modo que el visor de mapas pueda acceder a esta imagen y mostrarla al usuario, tomando como referencia los datos alfanuméricos y espaciales de la imagen ráster, de este modo, se pretende agregar un servicio Geoespacial más a la IDE “Geo Portal UTN” para dar calidad a sus servicios.

Para el desarrollo del CRUD es necesario establecer un marco teórico que permita conocer los conceptos que involucran un Sistema de Información Geográfica y los elementos que lo componen como son un Servidor de Mapas, Bases de Datos: Espaciales y Alfanuméricas, también se ve la necesidad de conocer los conceptos y la importancia de los objetos ráster ya que estos serán gestionados dentro del módulo software, su importancia permitirá conocer el tipo de información que brindan los mismos y que herramientas permiten manipular datos ráster.

A continuación, se describe las funciones del Módulo de Gestión Ráster

- Los objetos ráster se editan en una herramienta de escritorio de edición de capas de GIS.
- El módulo permite subir las capas guardadas a la base de datos de la aplicación.
- El módulo permite al visor de mapas acceder a las capas ráster que están en la base de datos, las cuales se puede relacionar según su información espacial y alfanumérica.
- El módulo realiza las operaciones básicas de un CRUD: Inserción, Edición, Eliminación, Carga, sobre los objetos ráster. La arquitectura del módulo de gestión ráster es del tipo MVC: Modelo, Vista, Controlador.

La arquitectura del módulo de gestión ráster es del tipo MVC: Modelo, Vista, Controlador

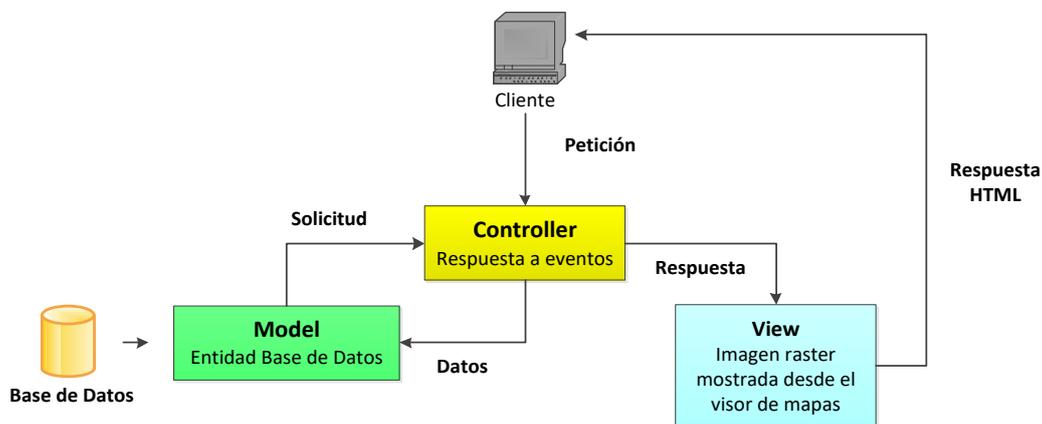


Fig. 1: Arquitectura del Módulo de Gestión Ráster

Fuente: Propia

“Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.” (Universidad de Alicante, 2015)

A continuación, se definen las herramientas para el desarrollo del módulo:

Lenguaje de Programación:	PHP
Servidor Web:	Apache
Herramientas de Diseño:	CSS, Bootstrap, J-Query
Base de Datos:	PostgreSQL
Gestor de Bases de Datos Espaciales:	PostGIS
Servidor de Mapas:	MapServer / GeoServer
Visor de Mapas Open Source:	p.mapper
API:	Google Chrome

Una de las metodologías para el desarrollo del módulo software será XP, como continuación a los proyectos anteriores relacionados al Geo portal.

“XP, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el aprendizaje de los desarrolladores, se basa en simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.” (Flores, 2011)

La metodología para validar los resultados se definirá durante el transcurso del proyecto.

A.4. JUSTIFICACIÓN

Una de las razones que se puede nombrar para la implementación de la iniciativa de un módulo de gestión ráster, es que el geo portal UTN trabaja con un conjunto de imágenes satelitales, ya incluidas en la API de Google Maps, que pierden definición al momento de buscar un preciso sector o zona, al no contar con un conjunto de imágenes ráster sobre la Zona 1 del Ecuador, se produce la pérdida de información a mayor profundidad en la búsqueda de datos a nivel regional.

De las áreas con las que se encuentra relacionado el proyecto se puede nombrar dos aspectos.

- a. Tecnológico: El Geo portal se encuentra alojado en una plataforma web, por lo que conlleva a la necesidad de un hardware para su uso, y software para la gestión del código principal.

- b. Socioeconómico: Ya que el Geo portal está auspiciado por la Universidad Técnica del Norte debe ofrecer servicios de calidad a nivel externo, como, por ejemplo, proyectos a empresas alineadas con la Zona 1, además debe proporcionar información precisa, clara y entendible para el usuario que requiera los servicios.

Con la creación de un nuevo servicio que gestionará imágenes ráster para el Geo portal UTN y al pertenecer a una entidad de educación superior, la parte socioeconómica y tecnológica del proyecto involucra dar apoyo a las metas del objetivo de desarrollo sostenible N° 9 de la Agenda 2030 de la (Organización de las Naciones Unidas, 2017): Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación, literales:

9.1. “Desarrollar infraestructuras fiables, sostenibles, resilientes y de calidad, incluidas infraestructuras regionales y transfronterizas, para apoyar el desarrollo económico y el bienestar humano, haciendo hincapié en el acceso asequible y equitativo para todos”.

9.2. “Modernizar la infraestructura y reconvertir las industrias para que sean sostenibles, utilizando los recursos con mayor eficacia y promoviendo la adopción de tecnologías y procesos industriales limpios y ambientalmente racionales, y logrando que todos los países tomen medidas de acuerdo con sus capacidades respectivas”.

9.3. “Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación”.

Dichos objetivos permiten un enfoque de apoyo hacia al Eje 2 del Plan Nacional de Buen Vivir, Objetivo 5 incisos de políticas 5.6, 5.10, que involucran el *“Impulso de la Competitividad y productividad para el desarrollo económico y social de manera equilibrada”* (República Nacional del Ecuador, 2017)

Todos estos objetivos se encuentran enlazados al Fortalecimiento de los Enfoques Productivos de la Zona 1, tratados en el Plan Estratégico Institucional del Gobierno Provincial de Imbabura.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Infraestructura de Datos Espaciales

Los datos que se manejan a diario dentro de una empresa proporcionan información vital a todo momento, sea esta para implementarla dentro del negocio con fines de mejora o como apoyo para aumentar la competitividad y toma de decisiones, dichos datos siempre se van a encontrar georreferenciados, con sus respectivas rectificaciones y correcciones, es decir, pertenecen a un conjunto de coordenadas geográficas de una determinada zona.

Cuando se tiene la necesidad de publicar la información geográfica se debe de contar con una infraestructura que sea capaz de permitir funciones como compartir, intercambiar, combinar, analizar y acceder a los datos de forma útil y eficaz. (Rodríguez Lloret & Olivella, 2019)

“La definición clásica de una IDE es básicamente tecnológica, ya que la presenta como una red descentralizada de servidores, que incluye datos y atributos geográficos; metadatos; métodos de búsqueda, visualización y valoración de los datos (catálogos y cartografía en red) y algún mecanismo para proporcionar acceso a los datos espaciales” (Veintimilla Reyes & Avila Larrea, 2015)

Los datos geográficos de una IDE se muestran en diferentes formas / formatos como, por ejemplo, fotos georreferenciadas, capas GIS, API'S, satélites etc. De acuerdo con (Bermejo, 2017), son varios los factores que contribuyen a la necesidad de crear IDE's y se muestran a continuación:

- Existencia de un gran volumen de datos espaciales por diversas entidades empresariales
- Distribución de aplicaciones para dispositivos móviles que utilizan la información espacial en cualquier lugar y momento.
- Accesibilidad por parte de los usuarios a información gratuita mediante diferentes servicios
- Demanda actual de la información geográfica a nivel mundial

1.2. Sistemas de Información Geográfica (SIG)

Para una entidad empresarial, los mapas representan una fuente de datos geográficos que se pueden manipular para la mejora del negocio, competitividad o apoyo en la toma de decisiones, por lo cual es necesario contar con un Sistema de Información Geográfica encargado del análisis de la información geográfica, mediante herramientas tecnológicas con el fin de sacar el máximo provecho de dicha información, prestando un servicio preciso, pero entendible para un usuario, además de aplicable para su propio beneficio.

“Conocer dónde está sucediendo algo es de vital importancia si se quiere ir allí o bien se desea enviar allí a alguien, e incluso si se busca encontrar alguna otra información sobre ese lugar o simplemente, por ejemplo, informar a la gente que vive alrededor” (Rodríguez Lloret, Jesús Olivella, Rosa, 2010)

1.2.1. Conceptos: ¿Qué es un Sistema de Información Geográfica?

Es un conjunto de componentes informáticos que permiten el manejo de información geográfica, se encuentran desarrollados a partir de una base de datos con objetos geográficos, que permite el control de funciones para el análisis espacial. (Domínguez Bravo, 2017)

Los Sistemas de Información Geográfica ofrecen diversos servicios e interactúan con un usuario mediante funciones como consultas para la obtención de información clave.

“Un Sistema de Información Geográfica (GIS) es un sistema de hardware, software, datos, personas, organizaciones y convenios institucionales para la recopilación, almacenamiento, análisis y distribución de información de territorios del planeta, esto es, conectando mapas con bases de datos. De esta forma, señalando un objeto se conocen sus atributos, e inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía” (Universidad Autónoma de Juárez, 2015)

Las fuentes de datos de un GIS pueden ser fotografías aéreas, imágenes satelitales, mapas en diferentes definiciones, cartas topográficas digitales, que permitan realizar funciones de consulta y analítica como respuesta a la solución de problemas, permitiendo una mejor visualización de los datos de un sector específico para el análisis de los fenómenos geográficos que se dan durante una línea de tiempo.

1.2.2. Características

Un GIS, trabaja con mapas que representan una zona o espacio específico, debido a las funciones geográficas que posee, permite un análisis rápido, profundo y preciso de una realidad espacial, dando solución a problemas de zonificación, dichos datos son importantes para cualquier entidad con el fin de aumentar los factores de mejora mostrados en la **(Fig. 2)**, dando a entender la importancia de contar con un Sistema de Información Geográfica:

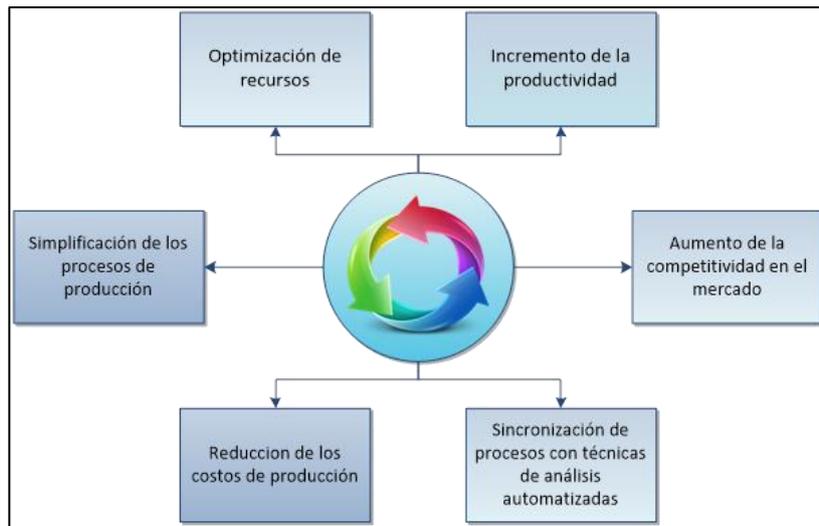


Fig. 2: Factores de mejora continua

Fuente: Propia

En comparación con la cartografía común, los mapas digitales aplicados en un GIS permiten obtener una visión real de cierta situación geográfica, proveyendo datos y resultados legibles por el usuario que realiza la petición, además, los datos analizados permiten el enriquecimiento del repositorio de datos espacial.

“El SIG funciona como una base de datos geográfica (datos alfanuméricos) asociada a los objetos existentes en un mapa digital y dan respuesta a las consultas interactivas de los usuarios, analizando y relacionando diferentes tipos de información con una sola localización geográfica” (Servicio Geológico Mexicano, 2017)

Mediante herramientas de analítica de datos incluso se puede obtener resultados mucho más precisos y valiosos, además de que un experto en GIS debe ser capaz de manejar objetos geográficos en aplicaciones de escritorio para capas de carácter

ráster o vectorial, facilitando la relación entre los resultados actuales, y los resultados que se espera obtener o brindar como servicio.

1.3. Componentes de un GIS

Para brindar un servicio adecuado, un GIS se encuentra formado por diversos componentes, un repositorio de datos que almacena la información geográfica, las capas como servicio, un servidor de mapas para recibir las peticiones, visor de mapas donde mostrar la cartografía, y una programación interna adecuada.

1.3.1. Base de Datos: PostgreSQL con PostGIS

- **PostgreSQL**

Las funcionalidades de PostgreSQL hacen de la herramienta una de las más adquiridas para el almacenamiento y manipulación de información, ya que soporta la mayoría de los sistemas operativos, adaptándose diversas funciones SQL disponibles

“PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado” (Zea Ordóñez, Molina Ríos, & Redrován Castillo, 2017)

Su versatilidad demuestra una confiabilidad e integridad de los datos de alto nivel y su capacidad escalable permite mantenerla actualizada constantemente.

- **Características Generales**

PostgreSQL es una herramienta completa y versátil para ser implementada en un GIS, de acuerdo con la página oficial de la herramienta, cuenta con las siguientes características:

- a. Compatible con características ACID al 100%
- b. Soporta la mayoría de los datos de diversos tipos
- c. Permite la creación de datos personalizados
- d. Base de datos con arquitectura de tipo objeto-relación (Herencia)
- e. Copias de seguridad, caracteres Unicode, internacionalización, regionalización
- f. Control de concurrencia multi-version (Multi Versión Concurrency Control), métodos de autenticación
- g. Documentación de apoyo y manuales de usuario en la web

h. Licencia BSD para liberación de código fuente

PostgreSQL posee diversas ventajas con respecto a su funcionalidad, razón por la cual es muy popular, aun así, a pesar de ser una herramienta flexible, como contrapartida PostgreSQL posee ciertas desventajas mostradas en la Tabla 1:

TABLA 1
VENTAJAS Y DESVENTAJAS DE POSTGRESQL

Ventajas	Desventajas
<ul style="list-style-type: none">○ Acceso Concurrente: Escritura de la tabla durante la ejecución de otros procesos sin bloqueos o esperas.	<ul style="list-style-type: none">○ Alto consumo de recursos: Debido a sus funciones de alto nivel
<ul style="list-style-type: none">○ Herramienta Libre: Permite el acceso a los códigos fuente gracias a su comunidad online	<ul style="list-style-type: none">○ Poco soporte online: Depende mucho del foro oficial de la herramienta
<ul style="list-style-type: none">○ Disponible para la mayoría de los sistemas Operativos	<ul style="list-style-type: none">○ Sintaxis poco intuitiva: transacciones erróneas, abortan rápidamente
<ul style="list-style-type: none">○ La concurrencia permite dar mejor respuesta ante el proceso de grandes volúmenes de datos	<ul style="list-style-type: none">○ Migraciones Complejas: Hay versiones que no soportan tablespaces del modelo de datos
<ul style="list-style-type: none">○ Posee varios complementos, que permiten convertir la BDD relacional en una BDD espacial	<ul style="list-style-type: none">○ Orientación a objetos mínima: Extensión simple con funciones básicas (Herencia)

Fuente: Página oficial de PostgreSQL

- **Complemento PostGIS**

La extensión PostGIS permite la gestión de objetos espaciales en una base de datos geográfica, y se aplica dentro de los GIS de empresas profesionales que trabajan con servicios geográficos. (PostGIS, 2016)

“PostGIS es un módulo que convierte al sistema de base de datos PostgreSQL. En una base de datos espacial, es estable, rápido, compatible con estándares y con cientos de funciones espaciales, actualmente es la base de datos espacial de código abierto más ampliamente usada” (Moreno Alonso, 2016)

Fácil en su gestión ya que permite la ejecución de consultas SQL espaciales sencillas y obtención de datos legible, trabajando sobre diversos tipos de datos

mostrados en la **(Fig. 3)** que pueden ser de carácter vectorial (polígono, punto, línea) o de carácter ráster (fotos aéreas, orto fotografías).

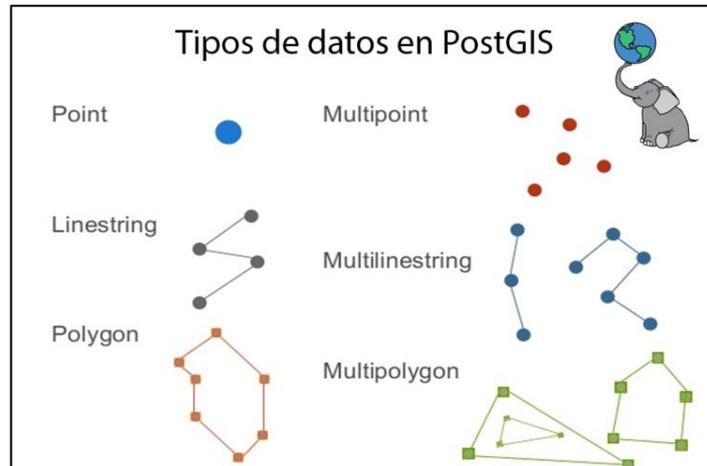


Fig. 3: Tipos de datos gestionados PostGIS

Fuente: (PostGIS, 2016)

“Aunque visualizar los resultados no es el objetivo principal de una consulta espacial, a menudo ayuda a comprender o explicar un resultado. Para visualizar los datos almacenados en PostGIS, existen diferentes alternativas, como exportar los datos a otro formato, como GML o ESRI Shapefile, o bien visualizarlos directamente con algún SIG” (Bollas, 2011)

➤ Características Generales

De acuerdo con la página web oficial de la herramienta (PostGIS, 2016) el complemento cuenta con las siguientes características:

- Es compatible con los estándares OGC (Open Geospatial Consortium)
- Posee funciones espaciales mediante SQL
- Realiza trabajos sincronizado con información topológica
- Incluye un visor de geometrías para la versión 4 de pgAdmin
- Permite el almacenamiento de varios tipos de geometrías en una sola tabla.
- Es rápido y seguro ya que consta de la indexación espacial de modo que se ahorran varios segundos en cada consulta
- La importación y exportación de capas se la realiza de manera sencilla
- Existen aplicaciones de escritorio que se adaptan a trabajar con PostGIS

Como se muestra en la **(Fig. 4)** la arquitectura de comunicación de PostGIS es sencilla y entendible, proporcionado diversidad de funciones, sean a nivel del cliente o a nivel de servidor

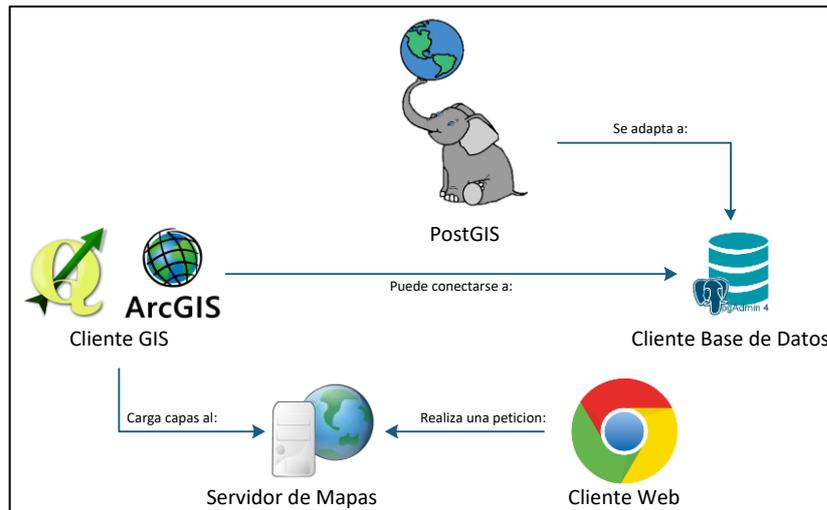


Fig. 4: Arquitectura de funcionamiento de PostGIS
Fuente: Propia

1.3.2. Gestión de Capas: ArcMap y QGIS

- **ArcMap**

Es una de las herramientas perteneciente al conjunto de programas de pago ArcGIS que permite la visualización y edición de los datos GIS, además es usada para el diseño mapas mediante capas, permitiendo la manipulación de éstas y de sus respectivos objetos espaciales (líneas, puntos, polígonos, ortofotos, TIN, MID).

“En esta herramienta se representa la información geográfica a ser publicada, como un conjunto de capas con diversos elementos comunes de un mapa, como, por ejemplo, barras de escalas, flechas de orientación, títulos, textos de descripción, leyenda de símbolos, todos estos conocidos como el marco de datos” (ESRI, 2016)

Las funciones que posee la ArcMap, según la página oficial de la herramienta, se muestran a continuación:

- **Trabajos con Mapas:** En un archivo mxd se habilita la muestra y edición de capas (**Fig. 5**), catálogo de datos, herramientas de geoprocésamiento, consultas etc.

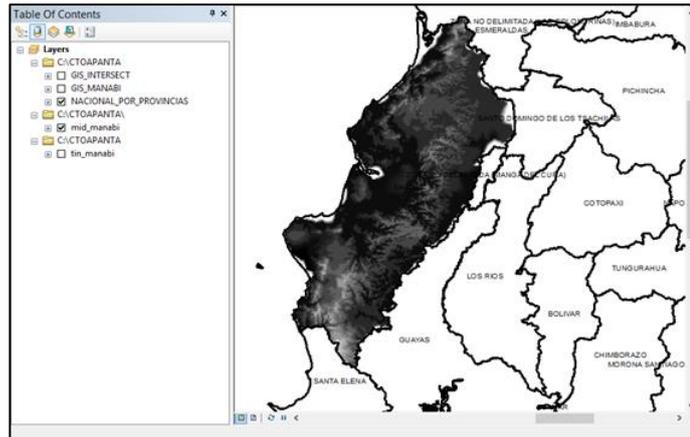


Fig. 5: Tabla de contenidos en ArcMap

Fuente: Propia

- **Impresión de mapas:** Permite la impresión de mapas básicos o complejos.
- **Compilación y edición de datos GIS:** Permite el uso automático de una base de datos geográfica para la manipulación de las capas
- **Geoprocesamiento y análisis de datos:** Se ejecutan comandos de geoprocesamiento y digitalización que se visualizan al instante en el visor
- **Publicación de mapas como servicio:** Los mapas editados se pueden publicar en un servidor de mapas externo.
- **Organización y administración objetos espaciales:** Consta de un catálogo de objetos espaciales (Fig. 6) para herramientas, BDD espaciales, capas, etc.

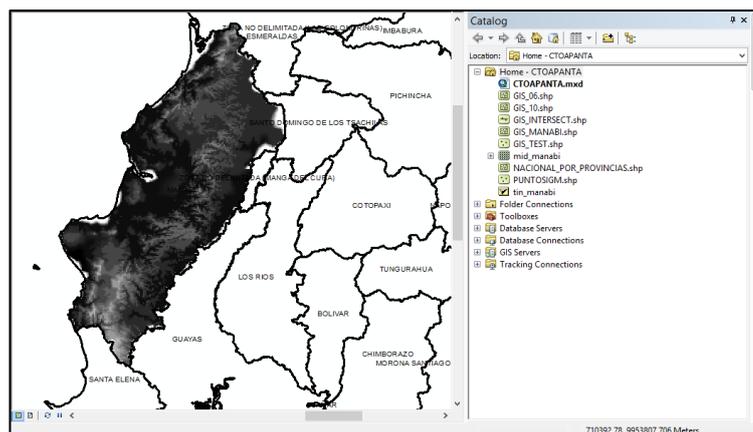


Fig. 6: Catalogo de objetos en ArcMap

Fuente: Propia

- **Documentación de los datos geográficos:** La descripción de la información ayuda a conocer, lo que representa un objeto de modo que, con el catálogo organizado, se puede documentar todo el contenido de un GIS.

- **QGIS**

A diferencia de los paquetes de ArcGIS, QGIS o Quantum GIS es una herramienta de carácter libre, que permite el uso abierto de la mayoría de las funciones de una herramienta GIS de escritorio. Soporta los datos clásicos: vectoriales y ráster, además de permitir la adición de otros formatos aplicando la estructura de complementos de QGIS. (Noe, Elena, Morales, & Ledesma, 2015)

“Quantum GIS (QGIS) es un Sistema de Información Geográfica con código abierto. El proyecto fue iniciado en mayo 2002 y establecido como proyecto en SourceForce en junio del mismo año. El objetivo ha sido democratizar los SIG para que lo pueda usar cualquiera con acceso básico a una computadora personal. QGIS corre actualmente sobre la mayoría de las plataformas Unix, Windows y Mac OSX” (Federación Internacional de la Cruz Roja, 2017)

Según la página web (OSGeoLive, 2015) , algunas características que identifican a la herramienta son las siguientes:

- a. Interfaz Gráfica (**Fig. 7**) con varias funcionalidades básicas como visor de capas, herramientas de geoprocésamiento, catálogo de herramientas etc.

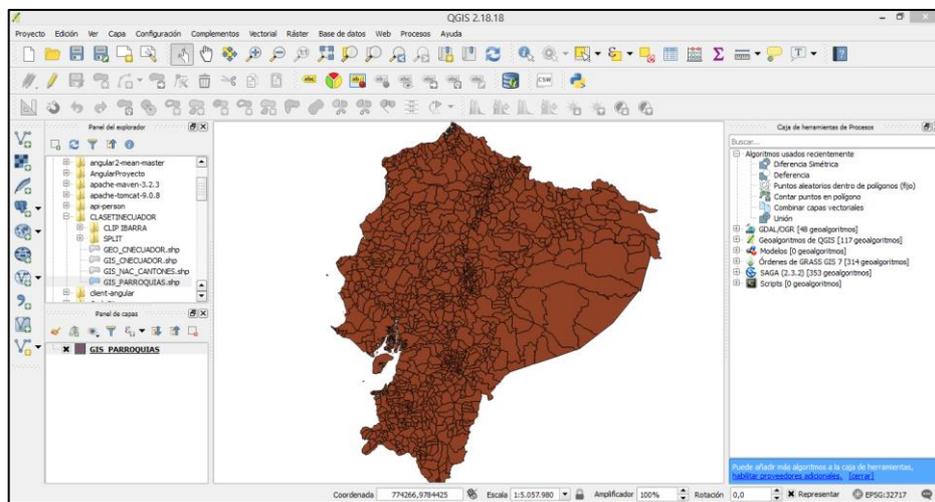


Fig. 7: Interfaz Principal de QGIS

Fuente: Propia

- b. Soporte de estándares OGC, con la capacidad de exportar su sistema de coordenadas
- c. Exportación de mapas personalizados a un servidor de mapas y plugin con funciones auxiliares (Ej. Lenguaje analítico 'R')

1.3.3. Servidor de Mapas: MapServer

- **MapServer**

Es una herramienta de desarrollo de carácter abierto (Open Source) desarrollado en C, que permite la creación de aplicaciones GIS en la web con el objetivo de almacenar, mostrar, consultar o analizar datos espaciales a través del internet, mediante diversos estándares como por ejemplo Servicio Web de Mapas “Web Map Services” (WMS). (Padrón, 2011)

Map Server proporciona una nueva forma de desarrollo para geo portales, dando acceso a diversas funciones geográficas, con cada uno de sus formatos del servidor, georreferenciados. La interfaz gráfica es personalizable soportando diversos lenguajes de alto nivel. (Ej. HTML, JavaScript, SQL) (Naranjo, 2017)

Los mapas solicitados por el usuario desde cualquier navegador se dan mediante una consulta al servidor de mapas, quien se encarga de acceder a la base de datos espacial propia o externa de una IDE para realizar la solicitud.

La configuración del servidor de mapas se encuentra en el archivo con extensión MAP, el cual describe los detalles geográficos y las capas a las que un usuario podrá acceder. Al modificar el archivo plano (**Fig. 8**), se está especificando que objetos (vectores, ortofotos) de un repositorio de datos podrán mostrarse.

```
#=====END LAYER gis_provincias=====#
#=====START LAYER gis_cnvcarchi=====#
LAYER
CONNECTION "dbname='gistestclon' host=localhost port=5432 user='admindata' password='D@4@G3@p@rt@1' sslmode=allow"
CONNECTIONTYPE POSTGIS
DATA "geom FROM "public"."gis_cnvcarchi" USING UNIQUE gid USING srid=24877"
EXTENT 688921.375 9451774 779528.3125 9610572
TEMPLATE "void"
METADATA
"ows_include_items" "all"
"gml_featureid" "gid"
"ows_extent" "688921.375 9451774.0 779528.3125 9610572.0"
"ows_srs" "EPSG:24877"
"ows_title" "gis_cnvcarchi"
"wms_getfeatureinfo_formatlist" "OGRGML"
"wms_bbox_extended" "true"
"gml_include_items" "all"
"PM_DESCRIPTION" "gis_cnvcarchi"
"PM_RESULT_FIELDS" "gid"
"PM_RESULT_FIELDS" "gid, iddivision, aniodivisi"
"PM_RESULT_HEADERS" "Numero, Division, Anio Division"
"PM_RESULT_HYPERLINK" '{ "gid": {"alias": null}}'
"PM_LAYER_ENCODING" "UTF-8"
END #METADATA
```

Fig. 8: Capa codificada en un archivo MAP

Fuente: Propia

Según la página web de (MapServer, 2017) es posible definir ciertas características como, por ejemplo:

- Ejecutable con Sistemas Operativos Linux y Windows
- Formatos ráster y vectoriales soportados
- Configuración sencilla de los parámetros vía GET en una URL
- Proporciona el complemento MapScript para tener acceso a las funciones del servidor de mapas, mediante lenguajes de programación como Java, PHP, Python, C, etc.
- Soporte de estándares como servicio del Open Geospatial Consortium (OGC) aplicando servicios como por ejemplo Web Map Server (WMS) o Web Coverage Services (WCS para Imágenes Ráster)

La siguiente comparativa mostrada en **(Tabla 2)** se la realiza contra el servidor de mapas conocido como GeoServer el cual es un servidor de mapas que permite mostrar mapas y datos con diversos formatos.

Su interfaz gráfica **(Fig. 9)** se encuentra basada en los estándares OGC como WFS, WCS. para el acceso a los mapas, se encuentra conectada a diversas fuentes de datos, ya que se encuentra diseñado para ofrecer una operabilidad estable. (García, Fonts, & Gonzales, 2013)

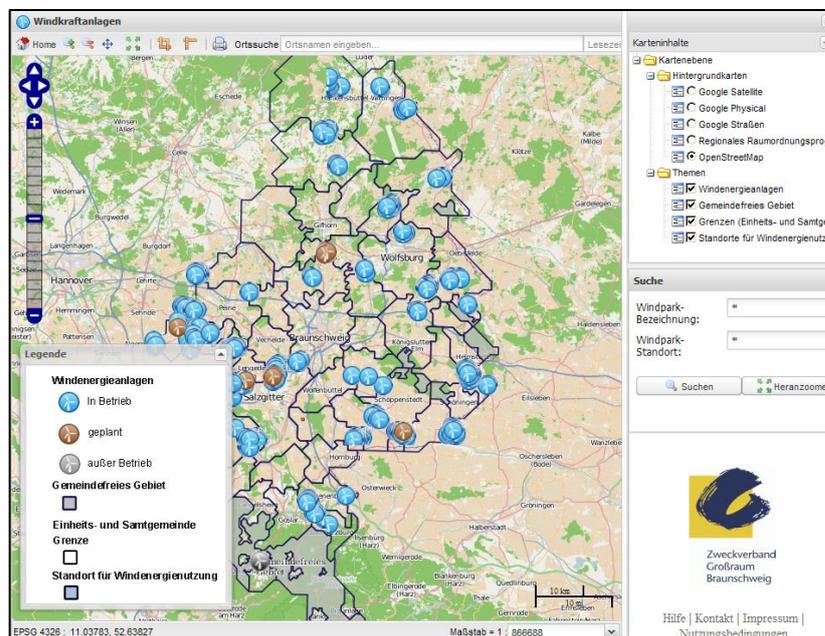


Fig. 9: Interfaz de Geo Server

Fuente: (OSGeoLive, 2015)

GeoServer actúa como nodo intermedio de una IDE, y responde a diversas solicitudes espaciales realizadas por un usuario, de carácter libre para la publicación de HTML

TABLA 2
COMPARATIVA ENTRE SERVIDORES DE MAPAS

MapServer	GeoServer
○ Maneja de mejor manera el servicio Web Map Service (WMS).	○ Es más eficaz con el servicio Web Feature Service (WFS).
○ No soporta servicio Web Feature Service Transaccional	○ Permite la aplicación correcta del servicio WFS-T
○ Es un proyecto mucho más antiguo, por ende, más usado (desde 1996)	○ Se lo realizó en el año 2002 de modo que es un proyecto más reciente
○ No proporciona una web para la administración de la configuración.	○ Proporciona una web administrativa de la configuración
○ Dispone de un sistema de cartografía de mayor calidad	○ Baja calidad del sistema de cartografía del cual se obtiene los mapas
○ El coste de procesamiento de imágenes es menor siempre y cuando se tenga acceso fácil a la información, si se utiliza cache para el acceso, es eficiente Map Server	○ Para el procesamiento de las imágenes, Geo Server es efectivo cuando la información se almacena en la base de datos, de modo que accede más eficientemente.
○ Necesita editar el archivo mapfile, de modo que se usa código	○ Posee una interfaz gráfica más sencilla y amigable al usuario.
○ MapServer es superior en el uso de procesador y hardware gracias a su codificación realizada en C.	○ GeoServer es una herramienta de carácter portable y aplicable a todos los sistemas operativos comunes

Fuente: (Junta de Andalucía, 2015)

1.3.4. Visor de Mapas: P. mapper

- **P. mapper**

P. mapper proporciona un visor de mapas personalizable y un menú para la gestión de capas (Fig. 9) que van a ser accesibles (visualizadas) por el usuario

“La herramienta p. mapper es un complemento para MapServer desarrollado en PHP/MapScript que permite la visualización de las capas que se encuentran dentro de una Infraestructura de Datos Espaciales (IDE), al igual que proporciona herramientas de visualización e información sobre el mapa al cual se accede.”
(Avila, Delgado, & Pacheco, 2013)

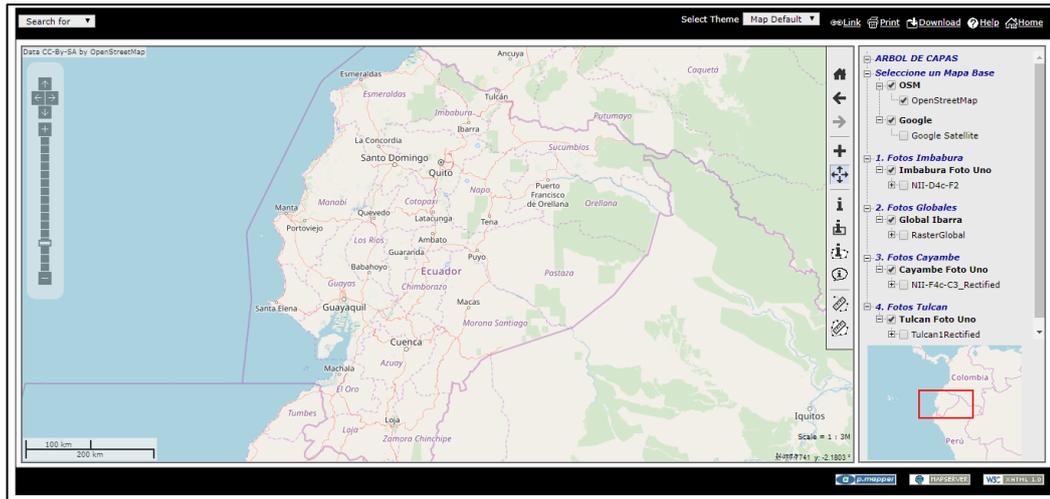


Fig. 10: Visor de mapas p. mapper (menú y visor)

Fuente: Propia

Conformado por varios scripts, el visor ofrece diversas funciones, que, según la documentación oficial de la herramienta, depende del tipo de mapa que se quiera implementar, entre esas funciones se puede encontrar las siguientes:

- a. Interfaz Gráfica DHTML con panel zoom/pan
- b. Soporte para diversos navegadores (Mozilla, Explorer, Opera, etc.)
- c. Consultas: Identificación, Selección, Búsqueda de una capa
- d. Resultados de consultas enlazados a repositorios externos
- e. Código JavaScript en las consultas para la obtención de información geográfica
- f. Funciones para medir áreas o distancias y soporte de API Google u otros
- g. Información de objetos espaciales y funciones XHTML (Ej. Menú)
- h. Soporte de diversos formatos (Fig. 10)

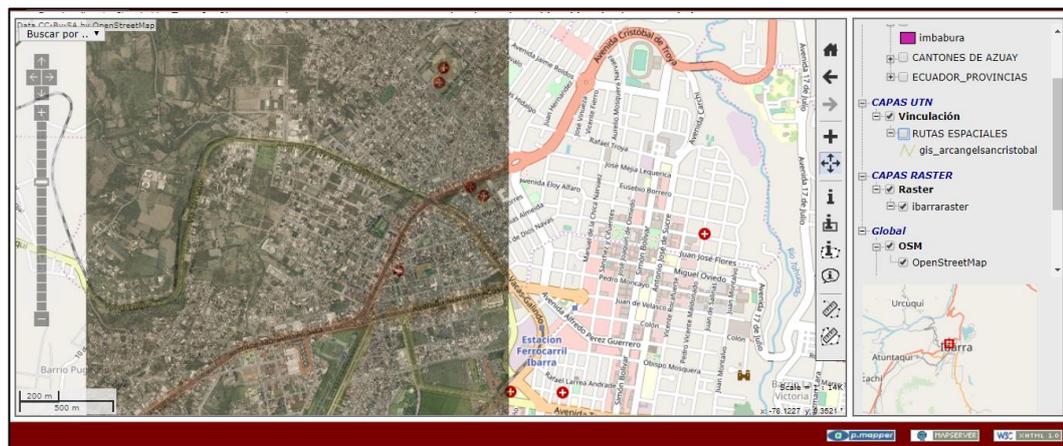


Fig. 11: Ejemplo de capa ráster

Fuente: Propia

1.3.5. Datos GIS: Vectorial vs Raster

PostgreSQL / PostGIS maneja datos de carácter vectorial y ráster (**Fig. 11**), que proporcionan ventajas dependiendo de la elección del diseñador y el tipo de propiedades topológicas con las que se está trabajando

“La mayoría de los elementos que existen en la naturaleza pueden ser representados mediante formas geométricas (puntos, líneas o polígonos, esto es, vectores) o mediante celdillas con información (ráster). Son formas de ilustrar el espacio intuitiva y versátilmente, y ayudan a comprender mejor los elementos objeto de estudio según su naturaleza.” (Osorio, 2010)

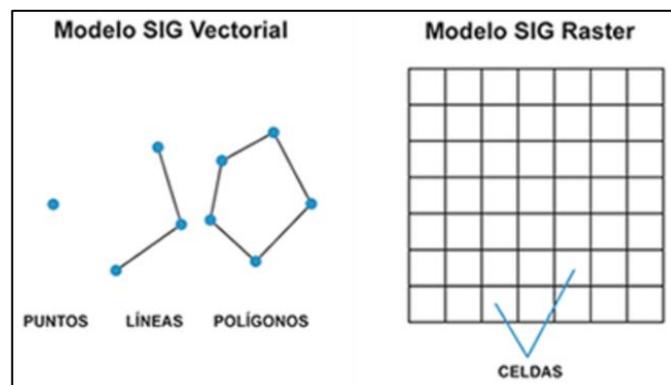


Fig. 12: Tipos de datos que se procesan en un GIS

Fuente: (Junta de Andalucía, 2015)

Según un artículo de la página web de (QGIS, 2013), los datos vectoriales representan un objeto del entorno real, constan de una geometría propia definida, representada en tres formas:

- Puntos.** - Permiten la representación digital de objetos que se encuentran en una única coordenada como por ejemplo señales viales, puntos de encuentro, edificios especiales, etc.
- Líneas.** - Facilitan la representación de objetos de tipo lineal (continuo en coordenadas), como, por ejemplo, ríos, avenidas, oleoductos etc.
- Polígonos.** - Formados por un conjunto de vértices (**Fig. 12**) este tipo de dato permite representar una zona o área cerrada específica como por ejemplo edificios especiales (escuelas, iglesias, parques etc.).

Todos los tipos de datos definidos anteriormente poseen atributos especiales, que se generan al momento de ser creados de modo que se puede definir diversos detalles de cada uno de los objetos espaciales.

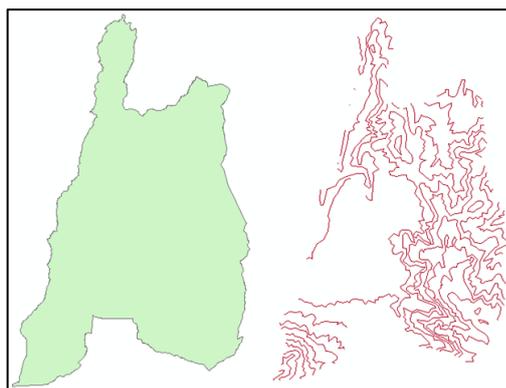


Fig. 13: Ejemplos de capas vectoriales

Fuente: Propia

“Algunos objetos espaciales en un paisaje son fácilmente representados como puntos, líneas y polígonos (por ejemplo, árboles, carreteras, casas). En otros casos puede ser difícil. Por ejemplo, ¿Cómo se representaría los pastizales? ¿Cómo polígonos? ¿Qué pasa con las variaciones de color que se puede ver en la hierba?” (Steinberg, 2012)

Los datos ráster dan solución a este tipo de problemas ya que permite representar valores con variaciones continuas, identificadas por las celdas de la matriz con la cual trabaja, a pesar de todo entre datos ráster y vectoriales existen diversos pros y contras que se muestran en la **(Tabla 3)**, mostrando su funcionalidad:

TABLA 3
COMPARATIVA ENTRE TIPO DE DATOS

Datos Vectoriales		Datos Ráster	
Ventajas	Desventajas	Ventajas	Desventajas
Geometría: Poseen geometría propia (línea, polígono, punto) con atributos editables.	Compleja: Su estructura de datos es imprecisa para irregularidades en el terreno	Matriz: Estructura de datos simple, con una tabla de atributos, cada celda es una coordenada.	Poco compacta: Su estructura de datos de celdillas genera archivos muy pesados
Sencilla: Las topologías son más completas y fáciles de diferenciar	Imprecisa: Superpone objetos con el riesgo de generar polígonos falsos e inservibles	Eficiente: Permite la obtención de resultados predictivos precisos según un conjunto de datos ráster.	Inconvenientes: Existen topologías que no se pueden representar adecuadamente
Agrupación: Pueden agruparse datos de un mismo tipo en una capa	Ineficiencia: Poco eficiente para procesar casos de variación espacio-tiempo.	Almacenamiento: Puede abarcar en su estructura, datos vectoriales.	Poca calidad gráfica: Al ser mapa de bits, su resolución es baja
Estructuración: El MDT del tipo TIN (Triangle Irregular Network) adapta las líneas de relieve en nodos y triángulos	Problemas satelitales: Las imágenes aéreas son difíciles de tratar a nivel vectorial	Eficaz: Permite realizar análisis estadísticos y espaciales a buen nivel	Problemas con el MDT: no se alinea fácilmente a una topografía cambiante, por ser malla pixelada

Fuente: (Gutierrez, 2015)

1.3.6. Codificación: Yii Framework

Todo framework tiene la función de actuar como biblioteca de código reutilizable para acelerar el desarrollo de software, Yii Framework es una herramienta enfocada en la estructura Modelo, Vista, Controlador (**Fig. 14**), creado por Qiang Yue, su nombre representa sencillez y maniobrabilidad, basado en lenguaje PHP implementa Programación Orientada a Objetos (**Tabla 4**) y reduce las complicaciones que representa MVC-PHP.

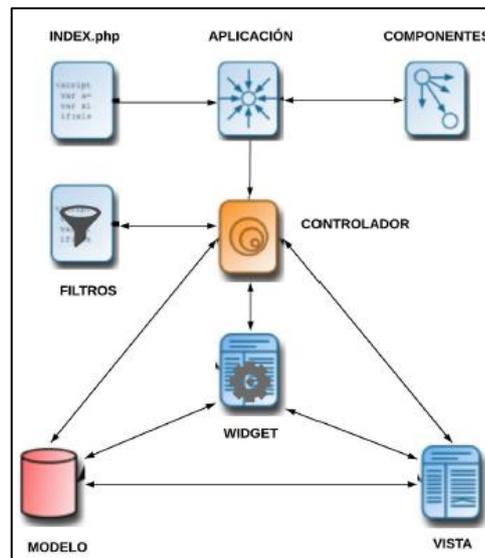


Fig. 14: Estructura de una aplicación con Yii

Fuente: (Quilca, 2019)

TABLA 4
OBJETOS DEL FRAMEWORK YII

Índice	Archivo que php que procesa la solicitud de un usuario, compila en base al archivo yii.php del framework, instanciando la aplicación, con su respectiva configuración
Aplicación	Se encarga de verificar el pedido de un usuario para que el controlador se encargue de dar una respuesta adecuada
Controlador	Archivo que se ejecuta cuando el cliente realiza una petición al mismo, mostrando los resultados de la petición, a través de una vista
Modelo	Representa a una tabla de la base datos o a un formulario e ingreso, con sus respectivas etiquetas personalizas y reglas según lo especificado
Vista	Forma en la como se presenta al usuario, es decir la ventana o menú principal
Componente	Controla las propiedades de la aplicación en sí y sus respectivos eventos
Widget	Permiten la creación de interfaces de usuario complejas y una mayor reusabilidad de estos
Filtros	Configuraciones de código que se ejecutan antes o después de una acción del controlador
Modulo	Parte del código que agrupa modelos vistas y controladores en uno solo. Útiles para la organización de la arquitectura y desarrollo organizado

Fuente: (Quilca, 2019)

1.4. Imágenes Raster

Un dato ráster, permite representar fenómenos espaciales, continuos (espacio-tiempo), y variables que pertenecen a un área o áreas específicas, consta de una matriz pixelada dividida en varias celdas las cuales contienen cierta cantidad de información, representada de la forma más precisa posible gracias a las celdillas de la matriz (Tuesta, Rengifo, & Bravo, 2011)

Cuando se intenta representar variaciones en áreas grandes, los datos ráster juegan un papel importante ya que se adaptan a una imagen para poder construir ideas predictivas o abstractas, es decir, factores cambiantes.

Al decir que representan fenómenos cambiantes en el mundo real la página de la herramienta ArcMap (ESRI, 2016) los clasifica en tres tipos:

- Datos temáticos que permiten la representación de objetos terrestres (Ej. Erosión)
- Datos continuos, encargados de representar fenómenos abstractos como lo son temperaturas, niveles de elevación. Entran en este aspecto las imágenes satelitales, fotografías aéreas rectificadas o imágenes de objetos

1.4.1. Usos Espaciales

“Si bien la estructura de datos ráster es simple, es excepcionalmente útil para una amplia variedad de aplicaciones” (ESRI, 2019), los usos especiales para los datos ráster, según la empresa Environmental Systems Research Institute (ESRI), son los siguientes:

- **Mapas Base:** Como fotografía georreferenciada (**Fig. 15**) que puede abarcar objetos vectoriales reales que pertenecen a una capa aparte



Fig. 15: Ortofoto base con capa de puntos

Fuente: Propia

- **Mapas Superficiales:** Representan datos continuamente cambiantes en un entorno real, es decir datos irregulares (superficies, elevaciones)

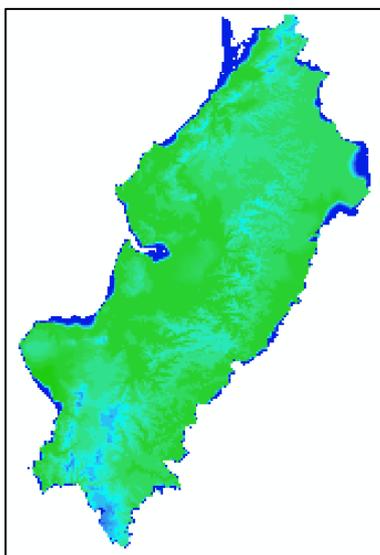


Fig. 16: Modelo Digital del Terreno
Fuente: Propia

- **Mapas Temáticos:** Formados por datos de carácter temático derivan del análisis de las fuentes de datos, se encargan de agrupar los resultados en diversas categorías (Ej. Niveles de Erosión de suelos)

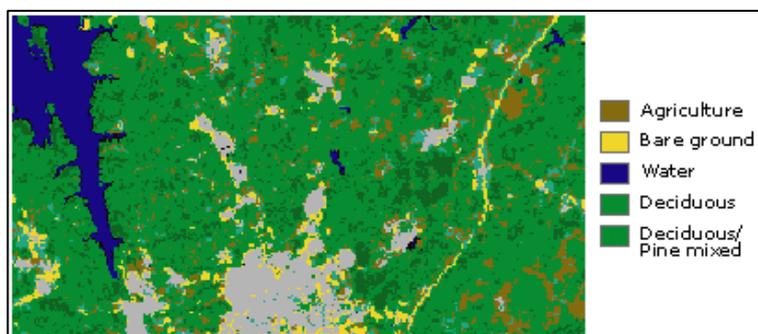


Fig. 17: Digitalización de un mapa post análisis
Fuente: (ESRI, 2016)

- **Objetos con atributos:** Hace referencia a fotografías digitales o archivos de escaneo sobre un objeto específico, dichos archivos poseen una ubicación exacta (coordenadas), representando un objeto estático real.

1.4.2. Georreferenciación

Para conocer la localización de una fotografía aérea que se haya conseguido, es necesario realizar el proceso de georreferenciación el cual proporciona la información exacta de su ubicación.

“La geolocalización permite situar geográficamente las fotografías antiguas y optimizar los sistemas de búsqueda y facilita los estudios de evolución de un lugar o de zonas de una ciudad” (Cascón-Katchadourian, Ruiz-Rodríguez, & Alberich-Pascual, 2018)

El proceso se realiza obteniendo sus respectivas coordenadas, las mismas que se encuentran alineadas con los datum (conjunto de coordenadas definidas a nivel general).

1.5. Estándar: Open Geospatial Consortium

1.5.1. Definición

Consortio internacional formado por diversas entidades empresariales gubernamentales que se enfocan en la implementación de buenas prácticas para el desarrollo de GIS, con requerimientos de alto y bajo nivel como por ejemplo información satelital, cartografía digital, codificación de información etc. (Masó, Núria, & Pons, 2010)

Los estándares de la OGC permiten la aplicación de estándares de carácter libre enfocados al aspecto de calidad, abarcando diversos campos de las Geociencias. Al ser libre y sin ninguna restricción, el estándar incentivó a la participación a nivel mundial de proveedores y desarrolladores para tecnologías GIS.

1.5.2. Servicios Geoespaciales

La página de la (OGC, 2017) da a entender que los servicios basados en los estándares son los siguientes:

- **Web Mapping Service.** - Muestra los mapas cartográficos como servicios mediante una consulta al servidor
- **Web Feature Service.** - Permite la interacción del usuario con los objetos geográficos (consulta, edición, descarga)
- **Web Coverage Service.** - Realiza peticiones de coberturas (imágenes) con su respectiva información, sobre un área específica

- **Catalog Service for the Web.** - Permite realizar consultas sobre metadatos de carácter geográfico
- **Gazetter WFS-G.** - Para la definición de nomenclaturas de fenómenos geográficos
- **Keyhole Markup Lenguaje (KML).** - Representa los datos geográficos de las API de Google, permitiendo su lectura y manipulación.

1.5.3. Web Coverage Services

Es un servicio definido por la Open Geospatial Consortium (OGC), cuyo objetivo es el de permitir a un usuario obtener datos de carácter espacial en forma coberturas digitales (Imágenes) junto con su respectiva información mediante consultas a un servidor de mapas por medio de protocolos HTTP. (Infraestructura de Datos Espaciales de España, 2018)

Al igual que un servicio Web Feature Service, el servicio WCS permite a un usuario que ha realizado la petición, el manejo de los datos espaciales de un objeto geográfico (CRUD de objetos de un GIS).

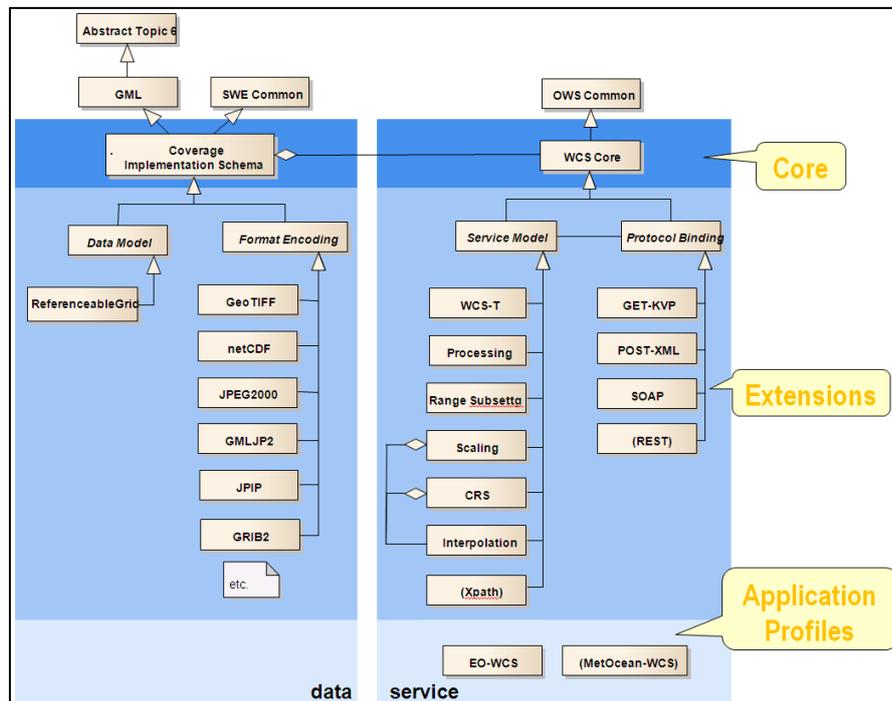


Fig. 18: Arquitectura Web Coverage Service

Fuente: (OGC, 2017)

1.6. Metodología de Desarrollo: XP

Es una metodología ágil para el desarrollo de software en periodos de tiempo cortos y efectivos, con reglas especificadas, enfocadas a fortalecer relaciones y valores de los miembros del equipo programador como clave del éxito, además de promover la comunicación entre el cliente y el equipo desarrollador (Quilca, 2019)

1.6.1. Componentes XP

Las historias de usuario (HU) en toda metodología proporcionan información entendible para el cliente y el desarrollador, ya que especifica los requisitos del software, se realiza un documento rápido y preciso, con características de los requisitos funcionales o no funcionales que el sistema debe contener, son de carácter variable ya que se puede añadir, modificar o reducir, va en conjunto con los roles XP (Borja Lopez, 2013) que se asignan a los miembros del equipo, entre los que se encuentran; programador, cliente, probador, encargado de seguimiento, coach, consultor, gestor, cada uno con su función clave donde:

- **Programador:** Desarrolla el código del sistema
- **Cliente:** Especifica requisitos del sistema
- **Testar:** Realiza pruebas regulares
- **Tracker:** Verifica los cumplimientos por tiempos y objetivos
- **Coach:** Controla la aplicación de las practicas XP
- **Consultor:** Resuelve problemas externos relacionados con el proyecto
- **Gestor:** Intermediario entre clientes y programadores, actúa de coordinador

1.6.2. Ciclo de vida XP

El ciclo de vida de XP (**Fig. 18**) permite verificar la finalización del proyecto en los tiempos determinados, de acuerdo con las habilidades de los miembros del equipo desarrollador, sin la necesidad de presionar al equipo a realizar más trabajo del especificado. Según los autores (Canós, Letelier, & Penadés, 2016) el proceso XP consta de 4 fases principales:

- **Exploración:** En esta fase se plantean los requisitos en las HU, se analiza las herramientas tecnológicas a aplicarse y de ser el caso, se diseña inicialmente la arquitectura del software y un prototipo.
- **Planificación:** Se establece la prioridad sobre cada una de las HU, el esfuerzo a aplicar y tiempos de entrega en base a la velocidad de desarrollo del equipo.

- **Iteraciones:** Se escoge las HU a ser aplicadas en cada iteración, es decir se organizan subtarefas en base a una HU, asignando responsables, todo en una planificación de entrega
- **Producción:** Fase en la que se realizan pruebas y revisiones de rendimiento antes de ser ingresado en un servidor principal (el del cliente)

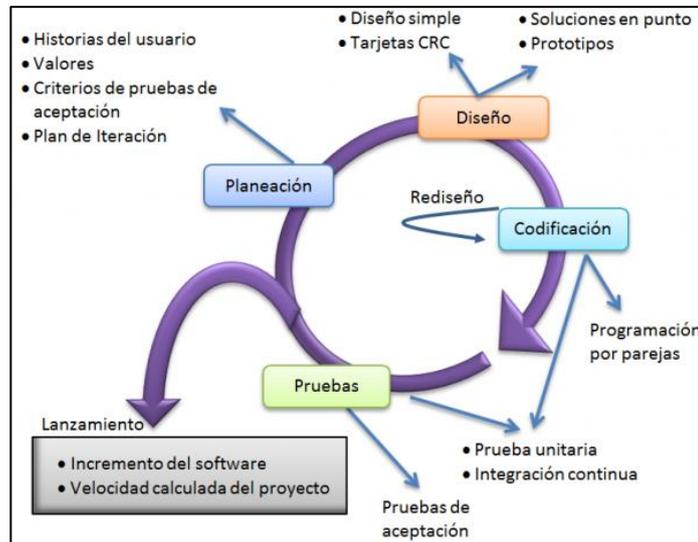


Fig. 19: Ciclo de vida de la metodología XP

Fuente: (Calvo, 2018)

CAPÍTULO II

2. DESARROLLO

Siguiendo las especificaciones de la metodología de desarrollo, como paso inicial se realiza una planificación en la cual se detallará a breves rasgos las personas implicadas en el proyecto, sus etapas a nivel general, y las fases XP a profundidad. Dos de los usuarios implicados, cumplen los siguientes roles dentro del desarrollo del módulo ráster:

- **Ciente**

Persona que se encarga de especificar los requerimientos del sistema, encargada de especificar los resultados y cambios a nivel de requisitos

- **Desarrollador**

Persona que se encarga de la manipulación del código fuente, realiza pruebas constantes de las funcionalidades del prototipo y lo implementa en el servidor

El desarrollo de la parte práctica se la divide en tres etapas principales que conforman el módulo ráster a implementarse en Geo Portal:

- **Primera Etapa**

Desarrollo de un módulo de archivos (zip) que permita la carga de los archivos que comprenden una imagen ráster y sus complementos a un repositorio de datos (local o servidor).

- **Segunda Etapa**

Creación automática de un menú de control para la activar la visualización de las capas ráster del directorio local, del visor de mapas del Geo portal.

- **Tercera Etapa**

Crear funciones que permitan manipular al usuario la creación y eliminación de los archivos (zip) dentro de su equipo local, además de la instalación del visor de mapas p. mapper para las pruebas del módulo de carga.

Al disponer de un módulo de objetos se mejora el servicio del visor de mapas y enriquece la información que se almacena en el repositorio de datos geográfico, la gestión de los archivos que conforman una capa ráster, pasa a ser de modo automatizado, sin la necesidad de acceder manualmente a los archivos de configuración del visor de mapas para la creación y asignación de capas a un menú.

Los procesos automatizados mejoran la eficacia en la carga de información y facilitan la gestión local de los objetos geográficos para los administradores del geo portal.

2.1. Fase de Exploración: Análisis

2.1.1. Especificación de Tecnologías

➤ Lenguaje de Programación: PHP

Es un lenguaje de programación de alto nivel para la creación de sitios web, se encuentra enfocado al servidor y soporta diversas librerías para mejorar su desempeño sea en diseño de interfaces o código interno, es gratuito adaptable y funcional en la mayoría de los sistemas operativos sus siglas PHP hacen referencia a PHP: Hipertext Process. (Quilca, 2019)

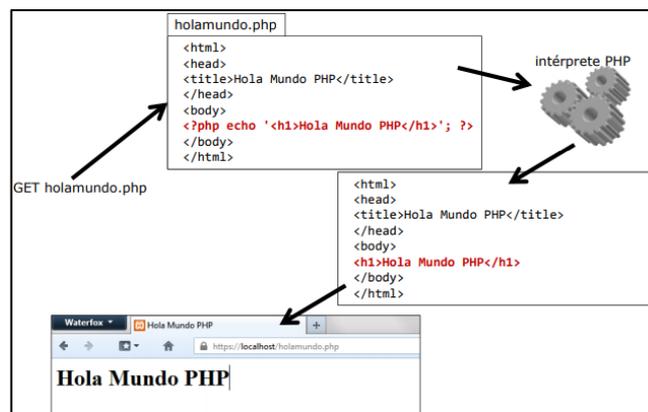


Fig. 20: Arquitectura de funcionamiento de PHP

Fuente: (Pavón, 2016)

Según el autor (Rodríguez M. , 2018) PHP contiene diversos factores que lo caracterizan como el lenguaje de programación más comprensible y sencillo de manipular:

- Es multiplataforma, a nivel de sistema operativo y conexión a base de datos
- Manipula datos de varias fuentes ingresadas por los usuarios desde las vistas
- Es escalable y con diversas extensiones y librerías para mejorar su potencial
- Extensa información a nivel de la web, documentación y foros oficiales
- Soporte de POO (Programación Orientada a Objetos)

➤ Servidor de Pruebas: XAMPP

La herramienta XAMPP actúa como un servidor local que permite alojar y testear los prototipos web base, antes de ser integrados en el servidor principal de la aplicación y con ciertas modificaciones en su configuración principal, se puede lograr que el servidor XAMPP funcione como un servidor de aplicaciones robusto, seguro y confiable, que permita el alojamiento de aplicaciones web profesionales.

2.1.2. Cartillas de Historias de Usuario

TABLA 5
HISTORIA DE USUARIO N° 1

HISTORIA DE USUARIO	
Numero: 1	Usuario: Administrador
Nombre HU: Carga de Capas Comprimidas	
Estimación: 5	Iteración: 1
Nivel Prioridad: Alta	Nivel Riesgo: Alta
Desarrollador: Cristoper Toapanta	
Descripción: Se requiere una función que permita la extracción de los datos de un archivo comprimido en una carpeta del equipo/servidor, además debe permitir la obtención de la dirección en la que se guarda el archivo para acceder a la imagen ráster.	
Procedimiento:	
<ul style="list-style-type: none"> • El usuario selecciona los archivos a comprimirse • Carga un archivo comprimido y presiona el botón de carga • Internamente, se crea una carpeta en la cual se descomprimen los archivos 	
Observaciones: <i>La compresión se realiza agrupando los archivos en una sola selección, no se debe comprimir carpetas</i>	
Fuente: Propia	

TABLA 6
HISTORIA DE USUARIO N° 2

HISTORIA DE USUARIO	
Numero: 2	Usuario: Administrador
Nombre HU: Carga de la información del comprimido	
Estimación: 3	Iteración: 1
Nivel Prioridad: Alta	Nivel Riesgo: Alta
Desarrollador: Cristoper Toapanta	
Descripción: Se requiere almacenar la información del archivo zip ingresada por el usuario en un repositorio de datos, con los campos, código, tipo de esquema, nombre del archivo, fecha de carga, dirección local de la imagen.	
Procedimiento:	
<ul style="list-style-type: none"> • Al presionar el botón de carga, se ingresa la información en el repositorio de datos, con los campos establecidos 	
Observaciones: <i>Los datos que ingresa son de carácter informativo, debe validar que no se ingrese el mismo archivo</i>	
Fuente: Propia	

TABLA 7
HISTORIA DE USUARIO N° 3

HISTORIA DE USUARIO	
Numero: 3	Usuario: Administrador
Nombre HU: Listado de archivos comprimidos	
Estimación: 5	Iteración: 1
Nivel Prioridad: Alta	Nivel Riesgo: Alta
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita mostrar los archivos zip comprimidos que se han cargado, con su respectiva información, en una tabla la cual debe estar visible para el usuario en el formulario.	
Procedimiento: <ul style="list-style-type: none">• Al principio la tabla se encuentra sin datos• Al presionar el botón de carga, debe aparecer en la tabla el archivo cargado	
Observaciones: <i>Los datos se mostrarán en la parte inferior de la ventana</i>	
Fuente: Propia	

TABLA 8
HISTORIA DE USUARIO N° 4

HISTORIA DE USUARIO	
Numero: 4	Usuario: Administrador
Nombre HU: Edición de los datos informativos	
Estimación: 2	Iteración: 1
Nivel Prioridad: Media	Nivel Riesgo: Bajo
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita una vista con las características informativas del comprimido que permita la edición de la etiqueta mostrada en el visor de mapas	
Proceso: <ul style="list-style-type: none">• El usuario puede ingresar desde la pantalla principal de carga de archivos• La etiqueta descrita que se muestra en el visor de mapas	
Observaciones: <i>La función se la puede desarrollar en la ventana del menú de capas</i>	
Fuente: Propia	

TABLA 9
HISTORIA DE USUARIO N° 5

HISTORIA DE USUARIO	
Numero: 5	Usuario: Administrador
Nombre HU: Eliminación de archivos ráster comprimidos	
Estimación: 5	Iteración: 1
Nivel Prioridad: Alta	Nivel Riesgo: Alto
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita una función que elimine un comprimido específico del equipo local y además debe eliminar la información de la base de datos	
Proceso por Pasos:	
<ul style="list-style-type: none"> • Al dar clic en un botón eliminar, debe borrarse del repositorio de datos • Debe encontrar el archivo a nivel local y eliminarlo 	
Observaciones: Verificar si la capa se utiliza en otros componentes, generar mensajes de advertencia si el archivo se usa en otros componentes	
Fuente: Propia	

TABLA 10
HISTORIA DE USUARIO N° 6

HISTORIA DE USUARIO	
Numero: 6	Usuario: Administrador
Nombre HU: Creación y gestión de menús	
Estimación: 5	Iteración: 2
Nivel Prioridad: Alta	Nivel Riesgo: Alto
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita un formulario que permita crear menús principales y submenús para el visor, además debe contar con los siguientes parámetros de ingreso:	
<ul style="list-style-type: none"> • Debe contener una lista desplegable con los nodos principales y con las capas disponibles • Un campo de caracteres para ingresar el número del menú y el nombre 	
Procedimiento:	
<ul style="list-style-type: none"> • Se ingresa la información del nodo a crear • Se debe asignar un nodo principal en caso de ser submenú • Se debe asignar una capa para los que son nodos hijos • Al presionar el botón debe cargarse la información en el repositorio de datos 	
Observaciones: Validar la creación de menús, la existencia de un menú padre para la creación de un submenú y que el nivel del nodo sea menor a 3 para asignar una capa	
Fuente: Propia	

TABLA 11
HISTORIA DE USUARIO N° 7

HISTORIA DE USUARIO	
Numero: 7	Usuario: Administrador
Nombre HU: Lista de menús disponibles	
Estimación: 5	Iteración: 2
Nivel Prioridad: Alta	Nivel Riesgo: Alta
Desarrollador: Cristoper Toapanta	
<p>Descripción: Se necesita una pantalla que muestre el estado en el que se encuentra el menú ráster, es decir, que liste todos los nodos creados con sus respectivas capas asignadas además debe contar con las siguientes funciones:</p> <ul style="list-style-type: none"> • Un botón para eliminar la información de un nodo • Un botón que permita acceder a la ventana de edición de los nodos 	
<p>Procedimiento:</p> <ul style="list-style-type: none"> • Al concluir la creación de un nodo, debe listarse en la parte inferior del formulario • En caso de eliminarse un nodo, este, debe desaparecer del visor de mapas • Al presionar de edición, debe ir a la ventana para editar los datos de menú 	
<p>Observaciones: Al momento de eliminar un nodo, debe advertir al usuario que dicho nodo tiene una capa asignada</p>	

Fuente: Propia

TABLA 12
HISTORIA DE USUARIO N° 8

HISTORIA DE USUARIO	
Numero: 8	Usuario: Administrador
Nombre HU: Servidor de mapas local y visualizador	
Estimación: 4	Iteración: 3
Nivel Prioridad: Alta	Nivel Riesgo: Alta
Desarrollador: Cristoper Toapanta	
<p>Descripción: Se necesita instalar un visor de mapas para realizar las pruebas de carga de capas, menús y para la manipulación de los archivos planos MAP del visor de mapas y XML de configuraciones</p>	
<p>Procedimiento:</p> <ul style="list-style-type: none"> • Se instala el servidor de mapas MapServer de forma local para las pruebas • Se instala el cliente ligero (visor de mapas) p. mapper para la visualización 	
<p>Observaciones: Sin Observaciones</p>	

Fuente: Propia

TABLA 13
HISTORIA DE USUARIO N° 9

HISTORIA DE USUARIO	
Numero: 9	Usuario: Administrador
Nombre HU: Muestra de los menús creados en el visor	
Estimación: 4	Iteración: 3
Nivel Prioridad: Alta	Nivel Riesgo: Medio
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita una función que permita la escritura de códigos para menús principales, submenús y asignación de capas en el archivo XML del visor de mapas	
Procedimiento:	
<ul style="list-style-type: none"> • El usuario crea un menú o submenú en formulario de menús • Dependiendo del tipo de nodo, se crea el código en el archivo XML • Al cargar el visor de mapas, se muestran los nodos creados en la parte derecha 	
Observaciones: <i>Ya existen ejemplos de menús en el archivo XML de configuración</i>	

Fuente: Propia

TABLA 14
HISTORIA DE USUARIO N° 10

HISTORIA DE USUARIO	
Numero: 10	Usuario: Administrador
Nombre HU: Gestión de archivo plano MAP (Capas)	
Estimación: 3	Iteración: 3
Nivel Prioridad: Media	Nivel Riesgo: Medio
Desarrollador: Cristoper Toapanta	
Descripción: Se necesita una función que permita encontrar una determinada capa (línea de código) en el archivo MAP con el fin de antes de eliminarla del visor de mapas.	
Procedimiento:	
<ul style="list-style-type: none"> • El usuario ejecuta la opción eliminar en la lista de archivos comprimidos • Se busca la capa solicitada y se elimina la línea de código a nivel del archivo MAP 	
Observaciones: <i>Los cambios en los archivos se guardan automáticamente</i>	

Fuente: Propia

TABLA 15
HISTORIA DE USUARIO N° 11

HISTORIA DE USUARIO	
Numero: 11	Usuario: Administrador
Nombre HU: Gestión de archivos de configuración XML (Menús)	
Estimación: 4	Iteración: 3
Nivel Prioridad: Alta	Nivel Riesgo: Alto
Desarrollador: Cristoper Toapanta	
<p>Descripción: Se necesita una función que permita encontrar un determinado menú (línea de código) en el archivo XML de configuraciones con el fin de editar el nombre del menú o eliminarlo del árbol de capas en el visor.</p>	
<p>Procedimiento:</p> <ul style="list-style-type: none"> • El usuario ejecuta la opción editar o eliminar en la lista de menús • En caso de edición: Ingresa los nuevos parámetros y guarda los cambios • En caso de eliminación: Se busca el menú solicitado y se elimina la capa 	
<p>Observaciones: <i>Los cambios en los archivos se guardan automáticamente</i></p>	

Fuente: Propia

TABLA 16
HISTORIA DE USUARIO N° 12

HISTORIA DE USUARIO	
Numero: 12	Usuario: Administrador
Nombre HU: Visualización de las capas en el mapa	
Estimación: 4	Iteración: 3
Nivel Prioridad: Media	Nivel Riesgo: Alto
Desarrollador: Cristoper Toapanta	
<p>Descripción: Se necesita una función que permita escribir código de forma automática en el archivo de configuración del visor de mapas para el reconocimiento de las capas</p>	
<p>Procedimiento:</p> <ul style="list-style-type: none"> • El usuario sube un archivo zip comprimido al cargador • Durante la carga del archivo se escribe un código en el archivo MAP • Finaliza la carga y el archivo MAP se modifica con nuevo código 	
<p>Observaciones: <i>El código para capas ráster se encuentra definido según la documentación oficial del visor de mapas.</i></p>	

Fuente: Propia

2.2. Fase de Planificación: Diseño

2.2.1. Planificación de Iteraciones

Se detallan a continuación los trabajos a realizarse en cada una de las iteraciones, basadas en sus historias de usuario:

TABLA 17
PLANIFICACIÓN INICIAL DE ITERACIONES

ITERACIÓN	DESCRIPCIÓN
Iteración N° 1	Manejo de los archivos ráster comprimidos
Historia de Usuario N° 1	Carga de capas comprimidas
Historia de Usuario N° 2	Carga de la información del comprimido
Historia de Usuario N° 3	Listado de archivos comprimidos cargados
Historia de Usuario N° 4	Edición de los datos informativos de la capa
Historia de Usuario N° 5	Eliminación de archivos ráster comprimidos
Iteración N° 2	Manejo del menú ráster del visor de mapas
Historia de Usuario N° 6	Formulario para la creación de menús
Historia de Usuario N° 7	Listado del menú jerárquico de capas
Iteración N° 3	Manejo de los archivos de configuración del visor
Historia de Usuario N° 8	Servidor de mapas local y visualizador
Historia de Usuario N° 9	Muestra de los menús creados en el visor
Historia de Usuario N° 10	Gestión de archivo plano MAP (Capas)
Historia de Usuario N° 11	Gestión de archivos de configuración XML (Menús)
Historia de Usuario N° 12	Visualización de las capas en el mapa

Fuente: Propia

2.2.2. Tarjetas CRC

Las tarjetas Clase Responsabilidad Colaborador, permiten una representación rápida sobre cuál de las clases que componen el prototipo serán creadas.

TABLA 18
TARJETA CRC DE LOS OBJETOS CAPAS RÁSTER

Capas Raster	
Cargar Archivo	<ul style="list-style-type: none"> • Modelo: Capas Raster • Vista: Crear Raster • Vista: Actualizar Raster • Vista: Mostrar Raster • Controlador: Capas Raster
Guardar Información	
Editar Etiqueta	
Mostrar Lista	
Eliminar Capa	

Fuente: Propia

TABLA 19
TARJETA CRC DE LOS OBJETOS MENÚ RÁSTER

Menú Raster	
Crear Menú Principal	<ul style="list-style-type: none"> • Modelo: Menú Raster • Vista: Crear Menú • Vista: Crear Categoría • Vista: Mostrar Menú • Vista: Update • Vista: Update Padre • Controlador: Capas Raster
Crear Sub Menú	
Editar Menú Principal	
Editar Sub Menú	
Mostrar Menú	
Eliminar Menú	

Fuente: Propia

2.2.3. Diagramas UML: Casos de Uso

La (Fig. 21) representa un caso de uso general del módulo y las opciones principales con las que cuenta el usuario

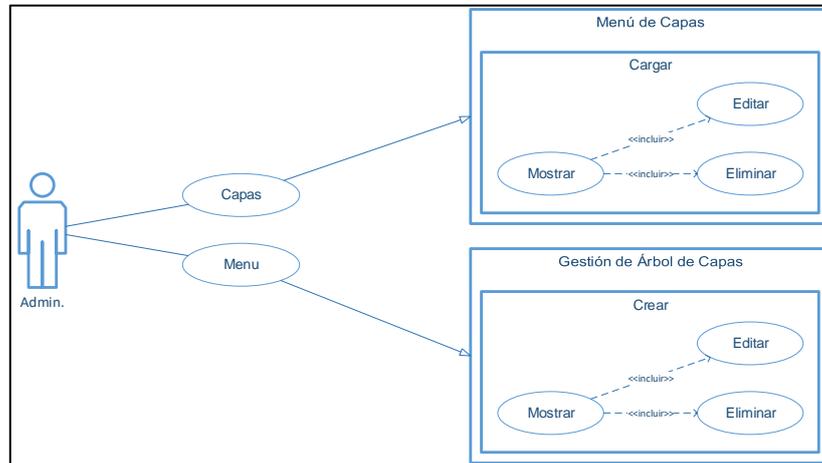


Fig. 21: Caso de Uso 1 – Funciones del Módulo de Carga

Fuente: Propia

La (Fig. 22) representa el escenario de carga de capas (comprimidos) con las funciones internas de almacenamiento de información, escritura en el archivo MAP y descompresión en el servidor.

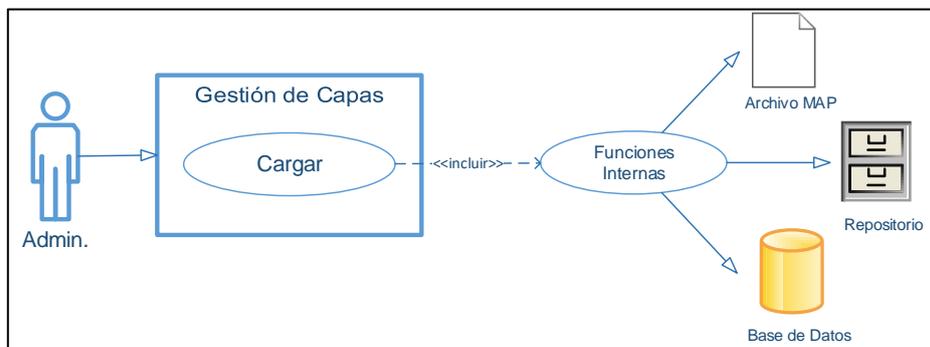


Fig. 22: Caso de Uso 2 – Escenario: Carga de Capas

Fuente: Propia

Para la (Fig. 23) el escenario de edición de la información resulta básico ya que los datos del archivo zip son en su mayoría privados (de servidor), siendo editable solo su 'alias' o etiqueta.

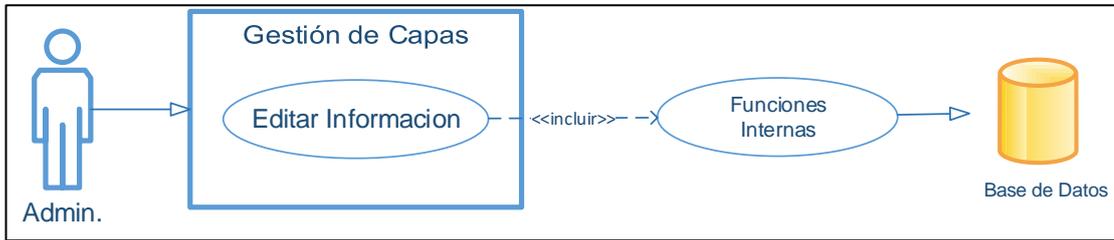


Fig. 23: Caso de Uso 2 – Escenario: Edición de Información

Fuente: Propia

El escenario de asignación de menú (Fig. 24) permite al usuario, crear un menú (nodos) para el árbol de capas del visor sin necesidad de acceder al archivo de configuración para la muestra del menú en el visor de mapas.

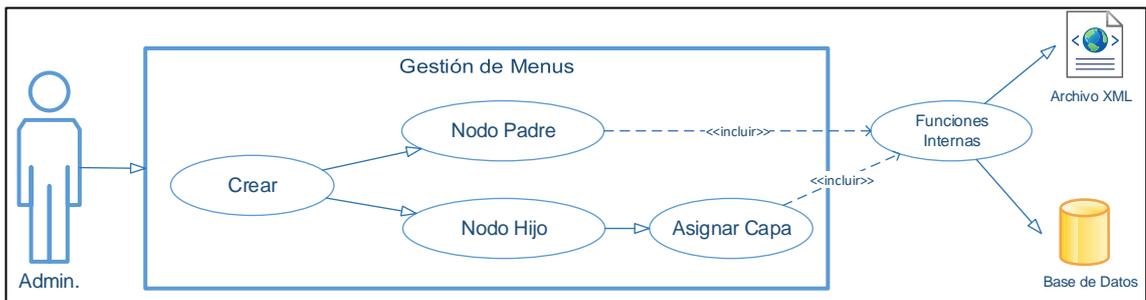


Fig. 24: Caso de Uso 3 – Escenario: Creación de Menús

Fuente: Propia

La (Fig. 25) representa el caso de uso para la edición de menús del visor de mapas, los cambios se realizan a nivel del archivo XML de configuración y de la base de datos

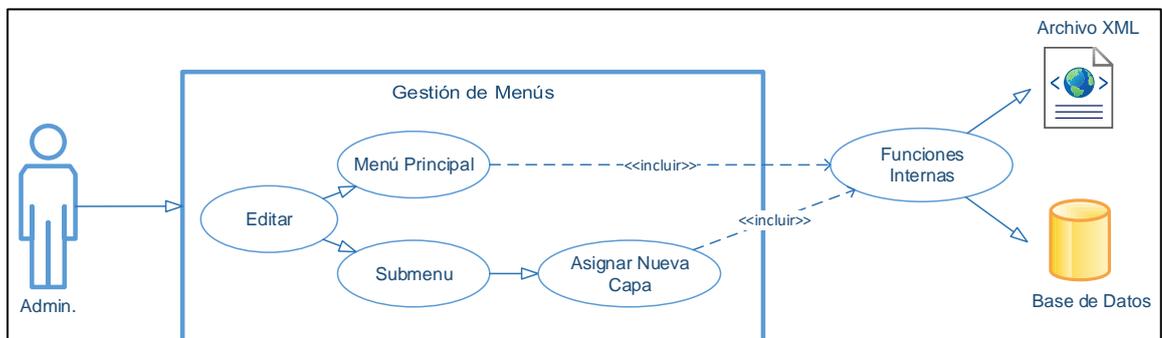


Fig. 25: Caso de Uso 4 – Escenario: Edición de un Menú

Fuente: Propia

El escenario presentado en la (Fig. 26) es el caso de uso para las funciones de eliminación para capas y árbol de menú, donde la capa se elimina a nivel del archivo MAP, del equipo local / servidor y la información de la base de datos, mientras que el menú se borra a nivel de la base de datos y de su archivo de configuración XML.

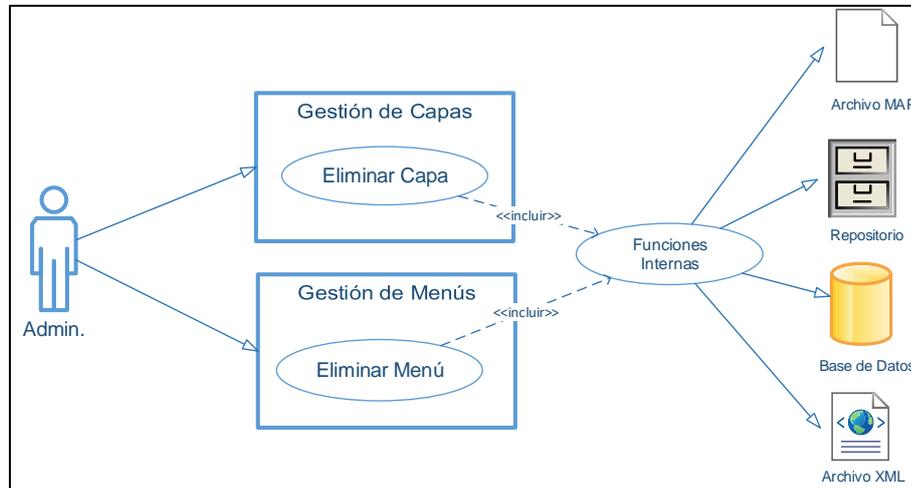


Fig. 26: Caso de Uso 5 – Escenario: Eliminación

Fuente: Propia

2.2.4. Diseño Simple: Arquitectura

El prototipo web de carga se encuentra realizado en un lenguaje PHP con base en el framework Yii 1.0 que permite la aplicación de librerías Bootstrap / CSS para el diseño y JavaScript / jQuery.

La arquitectura para visualización de capas está compuesta por un servidor de mapas Map Server para el almacenamiento de estas y el cliente ligero p. mapper aplicable para Map Server que permite modelizar el visor de mapas y añadir un menú de capas

Los archivos de control principales de Map Server y p. mapper que se modifican automáticamente son:

- Archivo MAP: Es un contenedor de objetos e información geográfica pertenecientes a un mapa.
- Archivo XML: Permite el control de los objetos que conforman el menú del visor de mapas

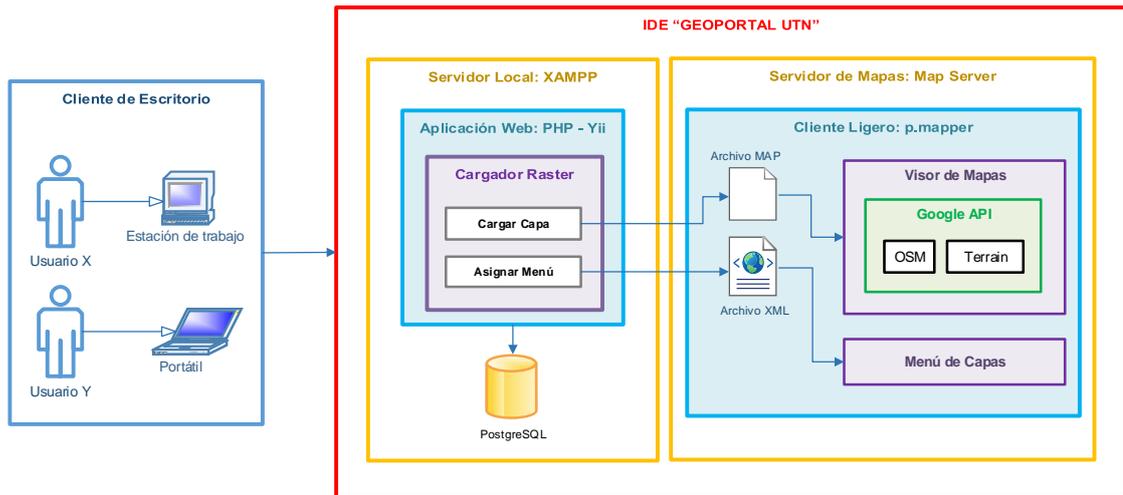


Fig. 27: Arquitectura del Prototipo

Fuente: Propia

2.3. Fase de Iteraciones: Desarrollo

Al ser Yii un framework basado en la estructura Modelo, Vista y Controlador, permite la ejecución de Gii (Fig. 28), un complemento que realiza la codificación automática de controladores y vistas según el modelo de datos de un aplicativo, de modo que permite obtener la una arquitectura de carpetas base para el prototipo.

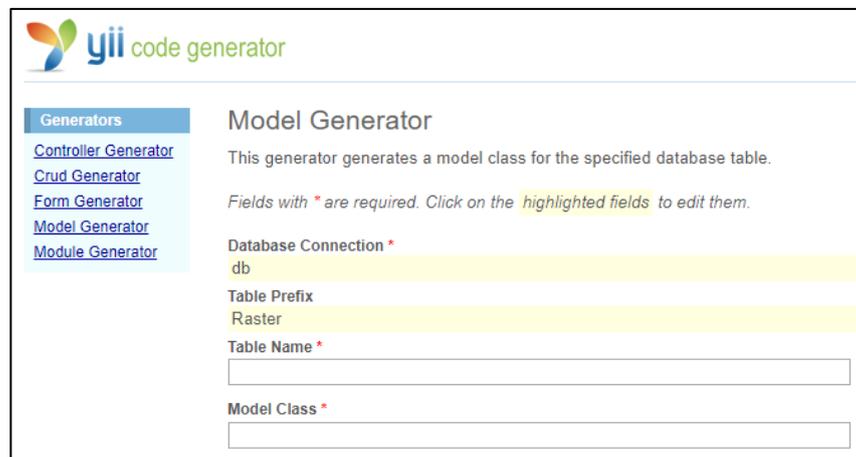


Fig. 28: Pantalla principal del generador Gii

Fuente: Propia

Para activar el generador se debe modificar el archivo de la dirección siguiente: `\protected\config\main.php` y añadir `"?r=gii/login/default"` a la url principal.

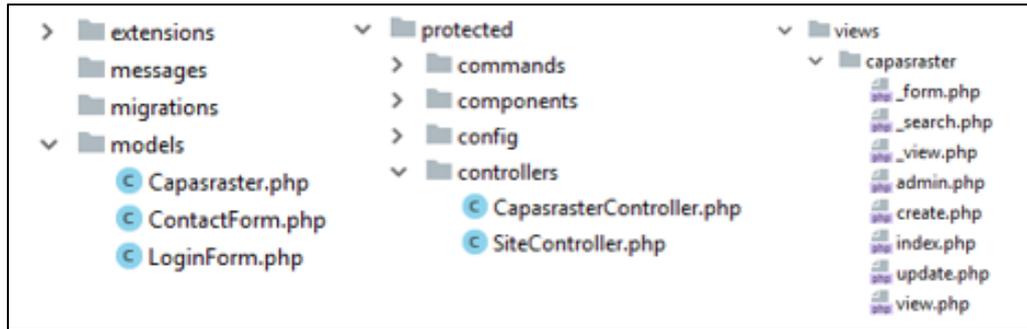


Fig. 29: Arquitectura CRUD generada por Gii

Fuente: Propia

La arquitectura generada por el complemento (Fig. 29) es de estilo MVC, siendo esta una de las facilidades del framework Yii, de modo que permite tener un esquema base para el prototipo que se desarrolla. Muchas de las clases especificadas en las Tarjetas CRC se van creando según las necesidades especificadas en las historias de usuario.

2.3.1. Iteración N° 1

- **Desarrollo de las Historias de Usuario 1 a 5**



Fig. 30: Formulario de Carga y sus componentes

Fuente: Propia

Como se muestra en la (Fig. 30), la pantalla principal para la carga de archivos consta de los siguientes componentes para su funcionamiento:

- a. Un campo que permite escoger un archivo comprimido del equipo para la subida a los diferentes servidores

- b. Un botón para ejecutar las funciones realizadas para la carga de la ráster en el repositorio de imágenes (Fig. 31) y su información en la base de datos
- c. Una lista con los archivos comprimidos que se van cargando y sus funciones de edición (Fig. 32) y eliminación respectivamente
- d. Una tabla con los botones funcionales que permiten ejecutar la eliminación, acceso a la pantalla de edición y vista.

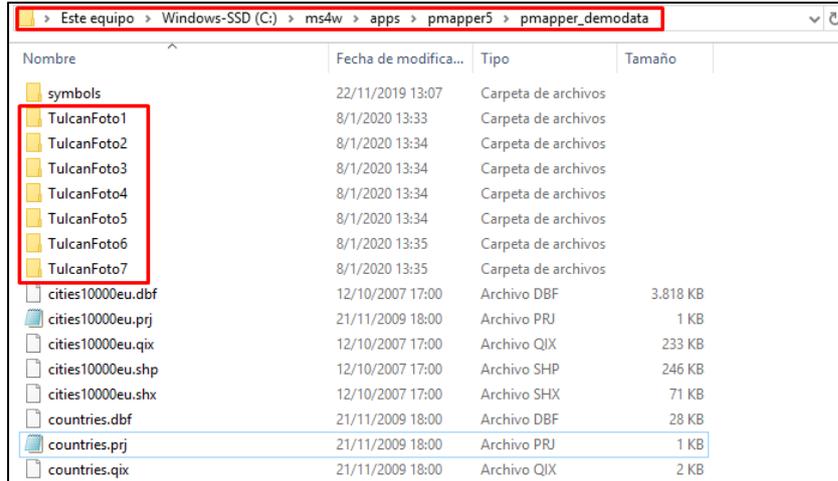


Fig. 31: Repositorio de Capas de MapServer

Fuente: Propia

CARACTERISTICAS LOCALES

ID Raster	Base de Datos
759	moduloRaster
Nombre de la Capa	Nombre en el Menu
Tulcan1Rectified	TulcanFoto1
Esquema de BDD	Fecha de Ingreso
public	2020-01-08
Direccion Local	
C:/ms4w/apps/pmapper5/pmapper_demodata/TulcanFoto1/Tulcan1Rectified.ti	
<input type="button" value="Save"/>	

Fig. 32: Pantalla de Edición de Etiqueta

Fuente: Propia

2.3.2. Iteración N° 2

- Desarrollo de las Historias de Usuario 6 y 7

IDE - GEOPORTAL UTN Pruebas Locales: Modulo Raster

Crear Menús & Categorías

Crear Menu Geografico Seleccione Categoria

Numero del Menu *

Menu Padre -- Seleccione Nodo Padre --

Nombre a Visualizar

Capa Asignada --- Asigne Imagen Raster ---

Create

Displaying 1-2 of 2 results.

Numero del Menu	Menu Padre	Capa Asignada	Nombre a Visualizar	Nivel	Es ultimo menu	
1.			Fotos Globales	1	S	
1.1	1.	759	Tulcan Foto Uno	2	N	

Fig. 33: Pantalla para la creación de Menús

Fuente: Propia

La pantalla principal para la creación de un menú (Fig. 33) que se muestra en el visor de mapas consta de los siguientes elementos:

- a. Campos para el ingreso del número y etiqueta del menú
- b. Lista desplegable con todos los nodos disponibles para la asignación de nodos padre a un nodo hijo
- c. Lista desplegable con capas disponibles para asignar a un nodo hijo
- d. Botón para la creación de los menús y carga de información en la BDD
- e. Una lista con la estructura del menú y sus opciones de edición (Fig. 34) y borrado de datos.

EDITAR MENÚ: "1. Fotos Globales"

Numero del Menu *

Nombre a Visualizar

Menu Padre

Nivel

Capa Asignada

Es ultimo menu

Save

EDITAR MENÚ: "1.1 Tulcan Foto Uno"

Numero del Menu *

Nombre a Visualizar

Menu Padre

Nivel

Nombre de la Capa

Es ultimo menu

Capa Asignada --- Asigne Nueva Capa ---

Save

Fig. 34: Pantallas de Edición para nodos padre e hijo

Fuente: Propia

2.3.3. Iteración N° 3

- Desarrollo de las Historias de Usuario 8 a 12

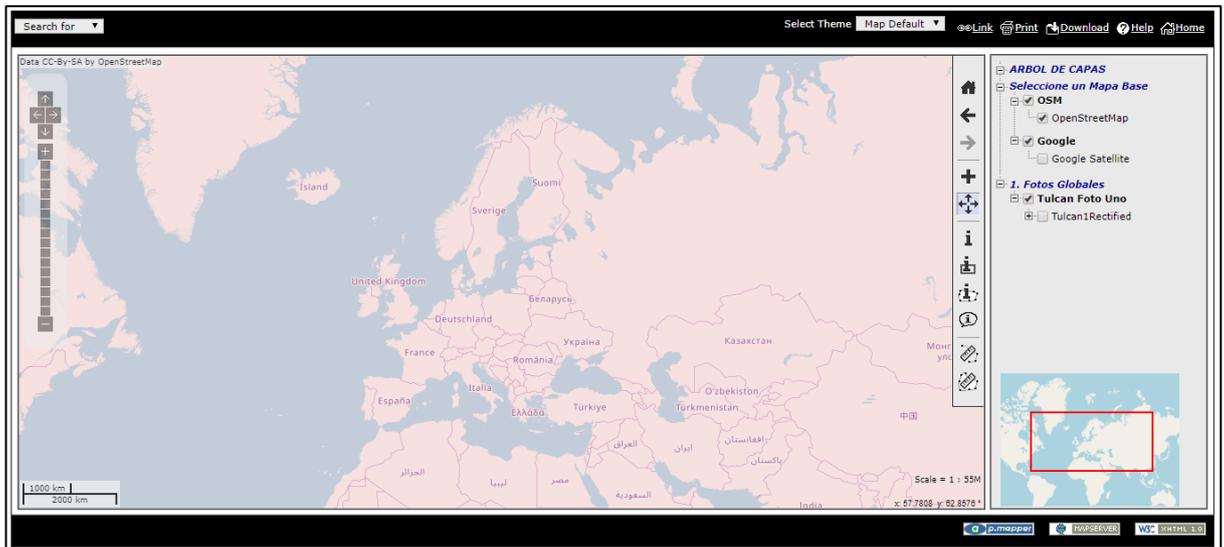


Fig. 35: Pantalla inicial del visor de mapas

Fuente: Propia

El cliente ligero p. mapper (Fig. 35) permite la publicación de un mapa y un menú de control para la activación de capas, de modo que permite al prototipo la manipulación de dicho visor en forma automática sin la necesidad de acceder a los archivos de configuración que lo conforman.

El archivo plano de formato. MAP "p.mapper_demo.map" permite el reconocimiento de las capas dentro del mapa mostrado, mientras que el archivo de formato XML "config_default.xml" permite el control del árbol de capas que se encuentra junto al mapa.

Durante el proceso de carga del archivo zip comprimido, se accede a los archivos que contiene el zip para la escritura del código necesario en el archivo MAP. (Fig. 36) del visor de mapas.

```
#===== START OF LAYER SECTION =====#  
LAYER  
NAME "Tulcan1Rectified"  
DATA "C:/ms4w/apps/pmapper5/pmapper_demodata/TulcanFotol/Tulcan1Rectified.tif"  
TYPE RASTER  
STATUS DEFAULT  
PROJECTION AUTO  
END  
END
```

Fig. 36: Ejemplo de capa en el archivo MAP

Fuente: Propia

Por otro lado, durante el proceso de creación de menús, dentro del archivo `config_default.xml` para el control del árbol en el visor, se realizan tres escrituras clave:

1. La etiqueta `<domain>` (Fig. 37) que permite la escritura de un nodo padre y una etiqueta de pista `<category>` junto a la 'id' del padre para que el nodo hijo sepa a qué padre enlazarse.

```
<domain id="domain1." description="1. Fotos Globales">
  <category>cat_Tulcan Foto Uno</category>
  <category>1.</category>
  <category></category>
</domain>
```

Fig. 37: Nodo padre con un nodo hijo asignado (submenú)

Fuente: Propia

2. La etiqueta `<category>` (Fig. 38) que permite especificar un nodo hijo y una `<group>` que abarca la capa asignada.

```
<category name="cat_Tulcan Foto Uno"
  description="Tulcan Foto Uno">
  <group
    olTheme="Tulcan1Rectified" imgFormat="png8"
    name="Tulcan1Rectified" description="Tulcan1Rectified"
    buffer="1" themeType="PMfile">
  </group>
</category>
```

Fig. 38: Nodo hijo con una capa asignada

Fuente: Propia

3. La etiqueta `<theme>` que representa la capa subida para ser asignada al nodo hijo especificado en el formulario (Fig. 39).

```
<olThemes>
  <theme>gsat</theme>
  <theme>osmmapnik</theme>
  <theme>Tulcan1Rectified</theme>
</olThemes>
```

Fig. 39: Escritura para una capa subida

Fuente: Propia

2.4. Generación de Servicios WCS

El servidor de mapas MapServer permite el soporte de diversos estándares de servicio de capas. Para Web Coverage Services se especifican tres funciones clave para permitir a un usuario controlar la información de una Ráster:

- **Get Capabilities:** Un archivo XML que permite conocer al usuario la información del servidor de mapas en el que se encuentra la capa.
- **Describe Coverage:** Un archivo XML que muestra al usuario los metadatos pertenecientes a una capa cargada.
- **Get Coverage:** Permite al usuario descargar la fotografía ráster solicitada

Con las tres funcionalidades, un usuario puede obtener la capa desde otro servidor y solicitar la información de cabecera mediante las funciones de descripción y obtención de datos de servidor para posteriormente procesarlas en su aplicación web o en una herramienta GIS de escritorio.

2.5. Fase de Pruebas

En la etapa de pruebas se ha planteado como objetivo la verificación del funcionamiento del módulo de carga según los casos de uso establecidos por cada etapa de desarrollo planificada, cumpliendo cada una de las historias de usuario establecidas con el fin de definir un prototipo entregable y corregir defectos encontrados.

2.5.1. Pruebas: Escenario de Carga y Gestión de Archivos Raster

TABLA 20
PRUEBA DE FUNCIONAMIENTO DE CARGA DE CAPAS

Objetivo:	Cargar una Capa Ráster
Estrategia:	Mediante codificación se realizaron diversas funciones que permiten la carga de un comprimido ráster en el equipo con su información local y geográfica en la base de datos, además consta con funciones que controlan el archivo MAP de visor habilitar la visualización de esta.
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm, Map Server
Resultados	Se realiza la carga de un comprimido ráster que se almacena en el equipo y en la base de datos
Observaciones	Se realizan las siguientes validaciones: <ul style="list-style-type: none"> • Formatos de Imagen: JP2, JPG, PNG, TIF, BMP, DAT, BIL, BIP, BSQ, GIF, GRID, IMG • Existencia de las imágenes en el comprimido • Formatos de Archivos Mundiales: J2W, JGW, PGW, TFW, GFW, BLW, BPW, BQW, BTW, HDR, GFW, RCW, SDW • Existencia de los archivos mundiales en el comprimido • Formato del Comprimido: ZIP • Forma de Compresión: Directa sin carpetas internas • Existencia del archivo SQL para la carga de la capa

Fuente: Propia

TABLA 21
PRUEBA DE FUNCIONAMIENTO DE ELIMINACIÓN DE CAPAS

Objetivo:	Eliminar una Capa Raster
Estrategia:	La generación de código de la herramienta Gii del Framework Yii permitió la creación de un método de eliminado el cual se editó con el fin de lograr la eliminación archivos a nivel de la base de datos y a nivel del equipo. El método permite quitar la capa directamente del visor de mapas controlando los archivos de configuración de este.
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm, p. mapper
Resultados	El usuario administrador elimina un archivo comprimido ráster del equipo y de la base de datos
Observaciones	Se valida que la capa a borrarse no tenga un menú asignado, en caso de darse, se enviara un mensaje de error

Fuente: Propia

2.5.2. Pruebas: Escenario de Creación y Gestión de Menú Raster

TABLA 22
PRUEBA DE FUNCIONAMIENTO DE CREACIÓN DE MENÚS

Objetivo:	Crear un menú y asignar una capa
Estrategia:	Los métodos codificados para la creación de menús se los ejecuta en una sola vista, ya que la programación permite validar cuando la capa es un nodo padre o un nodo hijo, además los nodos creados se visualizan inmediatamente en el visor del menú
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm, p. mapper
Resultados	El usuario administrador puede crear un menú y asignarle una de las capas que se encuentra en el repositorio de datos. La información del nodo se almacena en una tabla de PostgreSQL
Observaciones	Se realizan las siguientes validaciones: <ul style="list-style-type: none"> • Se valida que al nodo padre no se le asigne una capa (el menú se crea, pero la asignación de capa no se da) • Se valida que un nodo hijo pertenezca a un nodo padre • Se valida que un nodo hijo tenga una capa asignada • Se crea un mensaje de error en caso de no coincidir los nodos padre-hijo • Se crea un mensaje de error en caso de que un nodo hijo no tenga asignada una capa

Fuente: Propia

TABLA 23
PRUEBA DE FUNCIONAMIENTO DE EDICIÓN DE MENÚS

Objetivo:	Edición de Menú Padre e Hijo
Estrategia:	Se codificaron métodos que cambian la información de un nodo cualquiera. Para casos de edición a un nodo padre, se proporciona acceso a los campos informativos, mas no a sus claves, de la misma manera para un nodo hijo, siendo la única diferencia que a un nodo hijo se le puede volver a asignar cualquier otra capa cargada
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm, p. mapper
Resultados	El usuario administrador altera la información de un nodo padre o hijo en la base de datos y en el visor de mapas
Observaciones	Se modifica el archivo de configuraciones XML de p. mapper según los valores que ingresa el administrador, de modo que los cambios se reflejan inmediatamente en visor de mapas

Fuente: Propia

TABLA 24
PRUEBA DE FUNCIONAMIENTO PARA ELIMINAR MENÚ

Objetivo:	Eliminar un Menú Padre o Hijo
Estrategia:	El código generado por el complemento Gii del framework se lo modificó con el fin de permitir la eliminación de un nodo padre de la base de datos y del visor de mapas
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm, p. mapper
Resultados	El usuario administrador puede eliminar un nodo padre o nodo hijo de manera automática sin necesidad de acceder al archivo de XML que controla el árbol – menú del visor de mapa de modo que los cambios se reflejan inmediatamente.
Observaciones	Se realizan las siguientes validaciones: <ul style="list-style-type: none"> • Mensaje de error al eliminar un nodo padre que contenga nodos con capas asignadas.

Fuente: Propia

2.5.3. Pruebas: Escenario de Muestra de la Información

TABLA 25
PRUEBA DE FUNCIONAMIENTO - DATOS GEOGRÁFICOS

Objetivo:	Obtener datos geográficos de una capa
Estrategia:	Se codifico una función asociada a la creación de la capa que permite duplicar los datos de la tabla "raster_columns" y almacenarlos en una tabla diferente, además permite obtener las coordenadas de georreferencia del archivo mundial generado durante la rectificación.
Herramientas	Servidor XAMPP, PostgreSQL, IDE PhpStorm
Resultados	El usuario tiene acceso a una vista que en la que se visualiza la información geográfica de una capa cargada en la base de datos, además permite visualizar los datos obtenidos del archivo mundial de cabecera de una imagen ráster
Observaciones	Los valores de la tabla geográfica son solo de carácter informativo: no se puede editar o crear ya que la función de subida de capas hace la mayoría del trabajo

Fuente: Propia

CAPÍTULO III

3. RESULTADOS

3.1. Validación de Resultados

Para la validación de resultados, no se cuenta con una estándar que permita la especificación y organización de métricas, ya que la normativa establecida para el proyecto va enfocada al diseño de los servicios WCS para las fotografías ráster, es decir un servicio como normativa implementada para el módulo.

La IDE “Geo portal UTN” cuenta con diferentes servicios administrativos para el control de metadatos, simbologías, gestión de capas vectoriales e indicadores numéricos, además de servicios de mapas WMS, por lo que con la adición de un módulo ráster se busca la mejorar la proporción de servicios de información geográfica a un usuario.

Para la gestión de los componentes de una imagen aérea es necesario conocer la reacción del prototipo local diseñado durante la carga de capas ráster en diversos tamaños para uno de los formatos más conocidos: TIFF.

3.2.1. Pruebas de Rendimiento: J-Meter

La herramienta Apache J-Meter es una herramienta basada en código JAVA que permite realizar el análisis de rendimiento de una aplicación, mediante simulaciones sobre los recursos del servidor web o local. J-Meter permite realizar pruebas sobre peticiones enfocadas a los principales servicios como, por ejemplo: HTTP, HTTPS, WEB, SOAP.

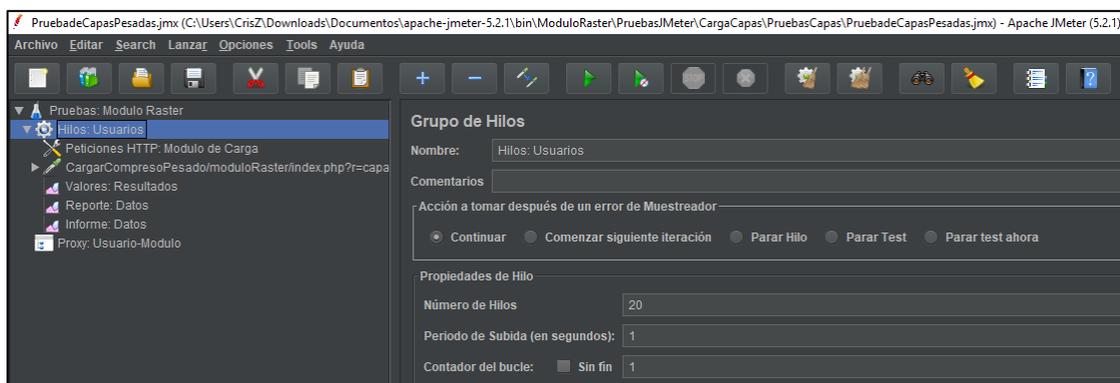


Fig. 40: Arquitectura inicial de un Proyecto J-Meter

Fuente: Propia

Para la correcta simulación es necesario establecer la cantidad de usuarios (Hilos) que realizaran la petición a la función que se ha especificado.

3.2.2. Escenarios para el análisis

La obtención de los datos realizada mediante simulaciones en J-Meter permitiendo comprobar el rendimiento del prototipo local de carga. Se diseñaron manualmente 3 peticiones del tipo Web – HTTP en el programa para tres escenarios divididos en:

- **Petición HTTP de Carga para una Capa de 2 MB.**

Al ser capas ligeras se estableció una muestra de 8 con los siguientes valores 55, 80, 105, 130, 155, 180, 205, 230 Usuarios, es decir un patrón de aumento de +25:

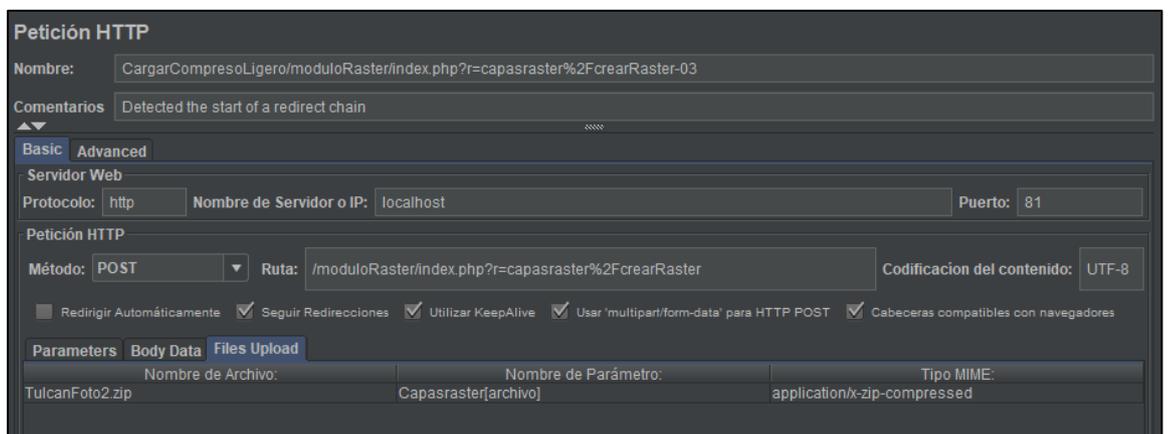


Fig. 41: Petición HTTP para Capas Ligeras

Fuente: Propia

TABLA 26
PARÁMETROS DE PRUEBA PARA LA PETICIÓN N° 1

Valores	Especificaciones
Protocolo:	http
Nombre del Servidor o IP:	Localhost
Puerto:	81
Método:	POST
Ruta:	/moduloRaster/index.php?r=capasraster%2FcrearRaster
Nombre del Archivo:	TulcanFoto2
Peso:	2.510 KB
Extensión:	Archivo con extensión ZIP

Fuente: Propia

- **Petición HTTP de Carga para una Capa de 100 MB.**

Para las capas medianas se decidió realizar 10 muestras con valores de usuarios para 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, en un patrón de aumento de +2.

Con el fin de esquivar las limitaciones del equipo personal que actúa como usuario y servidor, se decidieron valores cortos que permitan a la herramienta la correcta realización de la petición.

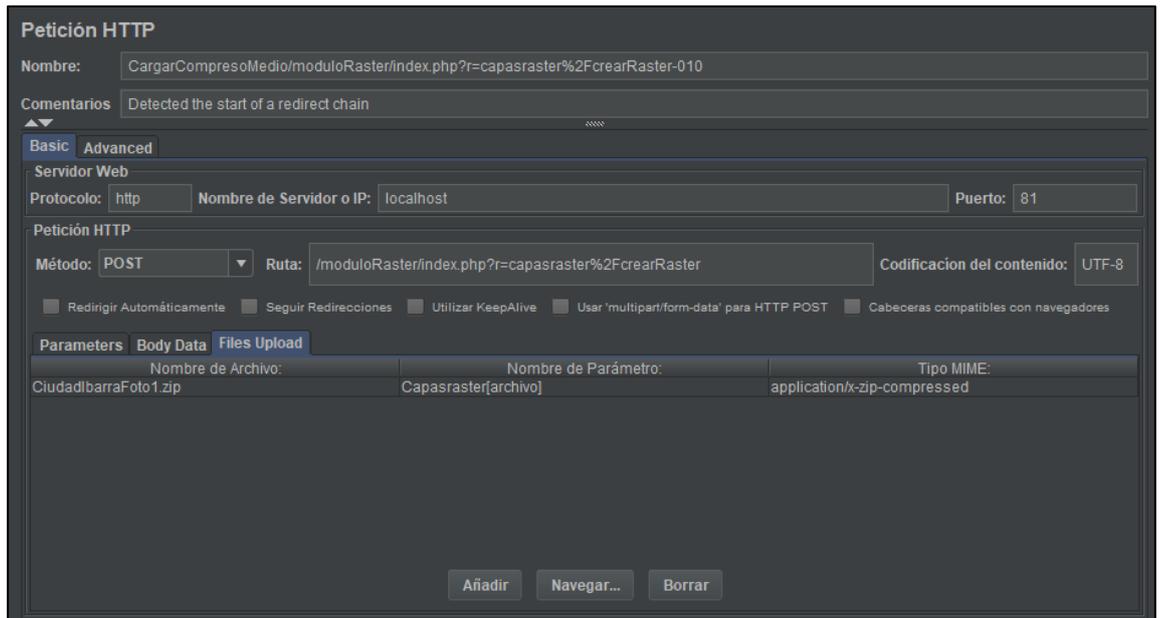


Fig. 42: Petición HTTP para Capas Medianas

Fuente: Propia

TABLA 27
PARÁMETROS DE PRUEBA DE LA PETICIÓN N° 2

Valores	Especificaciones
Protocolo:	http
Nombre del Servidor o IP:	Localhost
Puerto:	81
Método:	POST
Ruta:	/moduloRaster/index.php?r=capasraster%2FcrearRaster
Nombre del Archivo:	CiudadIbarraFoto1
Peso:	166.280 KB
Extensión:	Archivo con extensión ZIP

Fuente: Propia

- **Petición HTTP de Carga para una Capa de 200 MB o Superior**

Se estableció el mismo patrón de aumento de +2 para las capas pesadas: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 usuarios

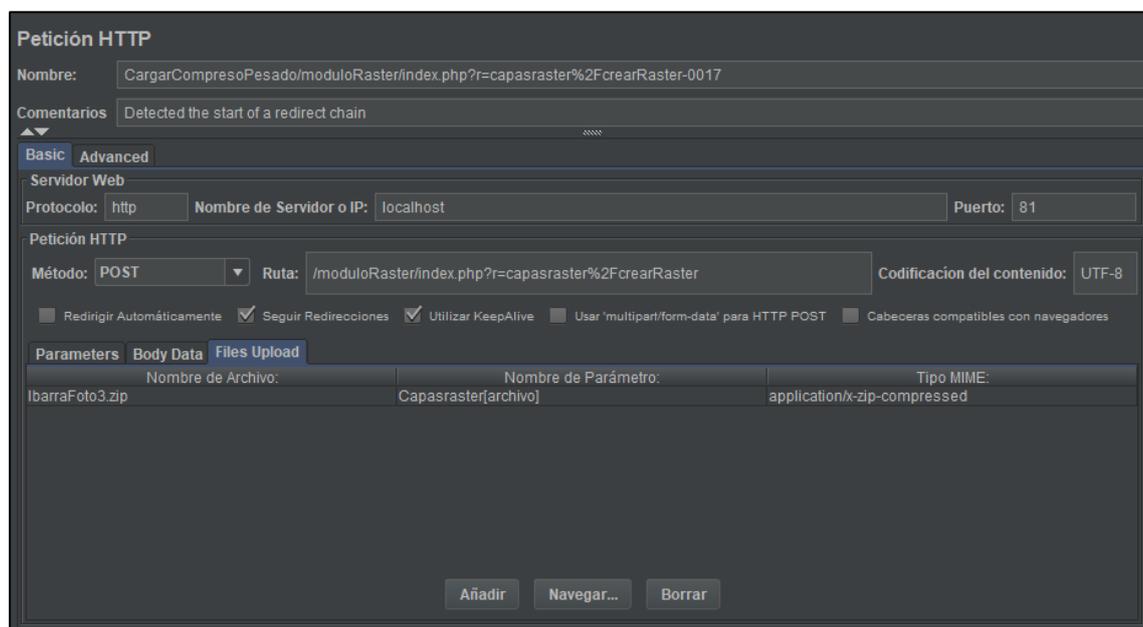


Fig. 43: Petición HTTP para Capas Pesadas

Fuente: Propia

TABLA 28
PARAMETROS DE PRUEBA DE LA PETICIÓN N° 3

Valores	Especificaciones
Protocolo:	http
Nombre del Servidor o IP:	Localhost
Puerto:	81
Método:	POST
Ruta:	/moduloRaster/index.php?r=capasraster%2FcrearRaster
Nombre del Archivo:	IbarraFoto3
Peso:	203.480 KB
Extensión:	Archivo con extensión ZIP

Fuente: Propia

Cabe recalcar que las “capas” no son solo el archivo TIFF principal, ya que una ráster puede contar con varios archivos planos como: Archivo de datos mundiales, de

proyección y XML auxiliares que contienen la información geográfica de la imagen procesada en una herramienta de escritorio.

2.2.3. Datos Obtenidos

Después de realizar la simulación de las peticiones HTTP que realizan la subida de las capas para los diferentes tamaños de zip comprimidos en J-Meter, se pudo obtener una tabla de valores para cada uno de los escenarios especificados:

- **Datos del Escenario 1: Capas Ligeras**

TABLA 29
VALORES OBTENIDOS PARA CAPAS LIGERAS

Usuarios	Media	Desviación Estándar	Error %	Rendimiento en Minutos	Rendimiento en Segundos	Kb x Seg	Kb x Seg Enviados	Media en Bytes
55	2411	335,54	0,00	0,276687796	16,60126773	279,48	41689,76	17239,0
80	3854	373,66	0,00	0,275311446	16,51868676	278,10	41482,41	17239,6
105	4463	584,10	0,00	0,301152986	18,06917914	304,21	45376,05	17239,9
130	6219	835,44	3,85	0,276184406	16,57106437	276,43	41613,67	17082,0
155	5945	1168,17	14,84	0,324865862	19,49195171	361,56	48947,92	18994,3
180	6504	1497,12	21,67	0,346300358	20,77802147	399,67	52176,41	19697,0
205	11203	2841,28	26,83	0,227868925	13,67213552	279,13	34332,40	20906,0
230	12531	2840,84	30,00	0,238673391	14,32040346	293,13	35960,21	20960,7
1140	53130	10476,15	97,19	2,267045170	136,022710176	2.471,71	341578,83	149358,5

Fuente: Propia

- **Datos del Escenario 2: Capas Medianas**

Los valores de muestra de los escenarios 2 y 3 son los mismos ya que el archivo zip comprimido pesa entre 100 kb y 200 kb o mas

TABLA 30
VALORES OBTENIDOS PARA CAPAS MEDIANAS

Usuarios	Media	Desviación Estándar	Error %	Rendimiento en Minutos	Rendimiento en Segundos	Kb x Seg	Kb x Seg Enviados	Media en Bytes
2	3371	112,00	0,00	30,15833124	1809,499874	8,80	102277,68	17923,0
4	5725	293,85	0,00	36,49080128	2189,448077	10,64	123753,34	17923,0
6	6299	576,99	0,00	44,07443683	2644,46621	12,86	149472,17	17923,0
8	9346	437,57	0,00	43,21598992	2592,959395	12,61	146560,87	17923,0
10	10961	701,35	10,00	50,43712172	3026,227303	13,95	171050,27	16989,3
12	14537	863,80	25,00	45,62737643	2737,642586	11,58	154738,64	15588,8
14	14779	1115,58	35,71	51,16335729	3069,801437	12,15	173513,07	14588,4
16	16216	837,69	43,75	53,82070976	3229,242586	12,12	182525,06	13838,1
18	16973	1598,25	44,44	56,32333768	3379,400261	12,63	191012,36	13773,2
20	20412	1323,30	55,00	51,38746146	3083,247688	10,70	174273,00	12787,6
110	118619	7860,38	213,90	462,6989236	27761,93542	118,04	1569176,46	159257,4

Fuente: Propia

- **Datos del Escenario 3: Capas Pesadas**

TABLA 31
VALORES OBTENIDOS PARA CAPAS PESADAS

Muestras	Media	Desviación Estándar	Error %	Rendimiento en Minutos	Rendimiento en Segundos	Kb x Seg	Kb x Seg Enviados	Media en Bytes
2	2955	10,50	0,00	34,88372093	2093,023256	0,97	96675,19	17243,0
4	3574	335,92	0,00	49,24086992	2954,452195	1,38	136463,96	17243,0
6	4724	494,61	0,00	54,95344222	3297,206533	4,20	152295,03	466,0
8	7378	467,97	0,00	53,60732633	3216,43958	4,10	148564,48	1616,1
10	8461	794,59	0,00	57,37234653	3442,340792	4,40	158998,66	469,0
12	11908	680,00	8,33	55,75776349	3345,46581	1,04	154524,08	1145,0
14	13227	921,13	14,29	56,28894994	3377,336997	1,49	155996,19	1628,6
16	14974	1452,08	18,75	55,28361647	3317,016988	1,79	153210,07	1991,3
18	17971	582,60	22,22	57,2124808	3432,748848	2,12	158555,61	2271,8
20	19287	556,15	55,00	58,81776296	3529,065778	4,72	163004,41	4935,3
110	104459	6295,55	118,59	533,4182796	32005,09678	26,21	1478287,68	49009,1

Fuente: Propia

3.2.4. Análisis de Resultados

Con la obtención de datos en los tres escenarios principales, se ha visto la necesidad de comprobar la relación entre los valores obtenidos durante la simulación mediante una función de regresión lineal para el análisis de las variables siguientes:

- Muestra (Usuarios)
- Media (Tiempo)
- Desviación estándar
- Porcentaje de Error
- Kb x Segundo Enviados

Todo con el fin de verificar que los datos obtenidos comprueban la forma de actuar del prototipo local sobre las peticiones de carga.

Es necesario conocer qué tipo de regresión lineal se aplicó para el caso actual, por lo que se realizaron pruebas de normalidad de los datos con el método de Shapiro-Wilk, el cual especifica que un valor cumple la regla de normalidad cuando su:

$$p - value \geq 0,05$$

Las pruebas de Shapiro-Wilk se las realizó con la herramienta estadística R Studio mediante el análisis de los valores obtenidos, dando los resultados mostrados en la tabla a continuación:

TABLA 32
PRUEBAS DE NORMALIDAD

Variables	p-value	Condición	Regla	Cumple
Muestras	0.9332			SI
Media	0.2803			SI
Desviación	0.06972			SI
Error	0.0811			SI
Rend. Minutos	0.7391	>=	0,005	SI
Rend. Segundos	0.7391			SI
KB x Segundo	0.007965			SI
KB x Seg. Enviados	0.7391			SI
Media Bytes	0.03821			SI

Fuente: Propia

Después de las pruebas aplicadas a las variables seleccionadas, se pudo constatar que todas cumplen con la regla de normalidad de datos por lo que es posible la aplicación de una prueba paramétrica de correlación.

El método de Correlación de Pearson verifica la covariación lineal entre dos variables X, Y cuantitativas y permite analizar la relación existente entre ambas variables, según la fórmula aplicada:

$$r_{xy} = \frac{\sum xy}{\sqrt{\sum x^2} \sqrt{\sum y^2}}$$

La regla por cumplir es la siguiente

$$1 \geq r_{xy} \geq -1 \quad \text{donde} \quad r_{xy} = \text{Coeficiente de Correlación}$$

Las relaciones entre variables analizadas con la Correlación de Pearson nos permitieron obtener los siguientes resultados:

	Muestras	Media	Desviación	Error	Rend.Min	Rend.Seg	KbxSeg	KbxSegEnv.	Bytes
Muestras	1.00	0.94	0.94	0.96	-0.23	-0.23	0.30	-0.23	0.92
Media	0.94	1.00	0.98	0.90	-0.55	-0.55	-0.04	-0.55	0.89
Desviación	0.94	0.98	1.00	0.95	-0.50	-0.50	0.05	-0.50	0.96
Error	0.96	0.90	0.95	1.00	-0.25	-0.25	0.32	-0.25	0.99
Rend.Min	-0.23	-0.55	-0.50	-0.25	1.00	1.00	0.83	1.00	-0.30
Rend.Seg	-0.23	-0.55	-0.50	-0.25	1.00	1.00	0.83	1.00	-0.30
KbxSeg	0.30	-0.04	0.05	0.32	0.83	0.83	1.00	0.83	0.28
KbxSegEnv.	-0.23	-0.55	-0.50	-0.25	1.00	1.00	0.83	1.00	-0.30
Bytes	0.92	0.89	0.96	0.99	-0.30	-0.30	0.28	-0.30	1.00

Fig. 44: Matriz de Correlación General (Capas Ligeras)
Fuente: Propia

- **Resultados:**

Los valores obtenidos en la (**Fig. 45**), para la relación entre las variables como Muestra - Media con un coeficiente de 0.9404957 y Muestra - Desviación con 0.9393996 en sus respectivos datos de correlación, permiten interpretar la gráfica y establecer una Correlación Alta Positiva entre el Número de Usuarios (Muestra) y la Media (Tiempo de Respuesta), es decir, a mayor número de usuarios, mayor será el tiempo de respuesta de la petición, estableciendo una relación ascendente entre las variables analizadas.

Operación	Valores	Resultados	
Significación del Coeficiente	t	6.7796	
	df	6,00	
	p-value	0.0005035	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0.6992354	0.9894312
	Valor de Correlación	0.9404957	

Operación	Valores	Resultados	
Significación del Coeficiente	T	6.712	
	df	6,00	
	p-value	0.0005314	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0.693950	0.9892315
	Valor de Correlación	0.9393996	

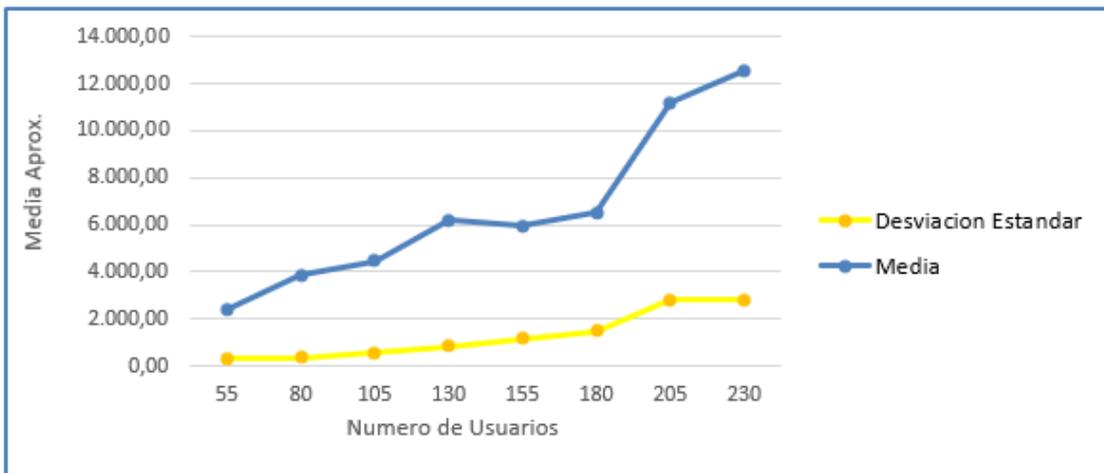


Fig. 45: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)

Fuente: Propia

Para la (**Fig. 46**) la relación entre Muestra - Error con un coeficiente de 0.9584829 se establece que la gráfica permite observar lo siguiente: a mayor número de usuarios que solicitan la petición de carga, el error aumenta con una Correlación Alta Positiva.

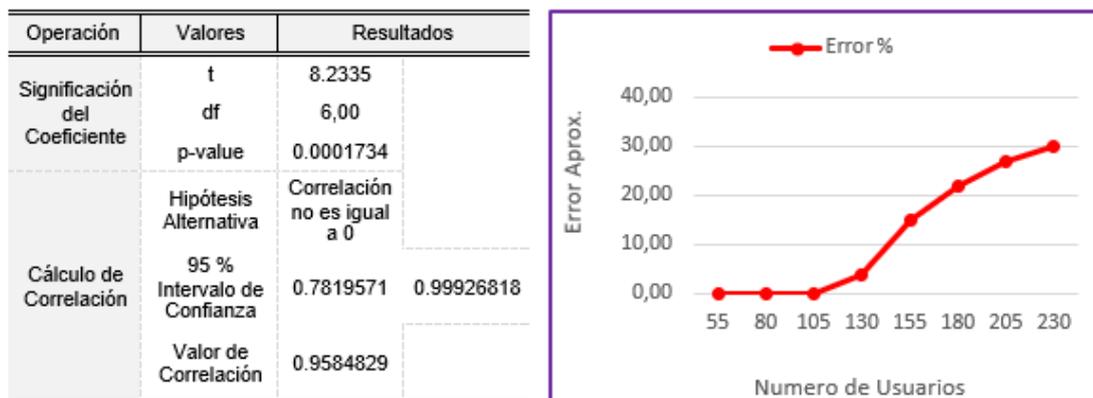


Fig. 46: Resultados y Gráfica de Correlación para Muestra-Error

Fuente: Propia

Los valores de la gráfica (Fig. 47) en la relación entre Muestra – Rendimiento con un coeficiente de $-0,232303$ establece una Correlación Negativa, estrechamente relacionada con error a causa de que, a mayores usuarios, mayor error, pero con un rendimiento decreciente, que depende de las características del servidor.

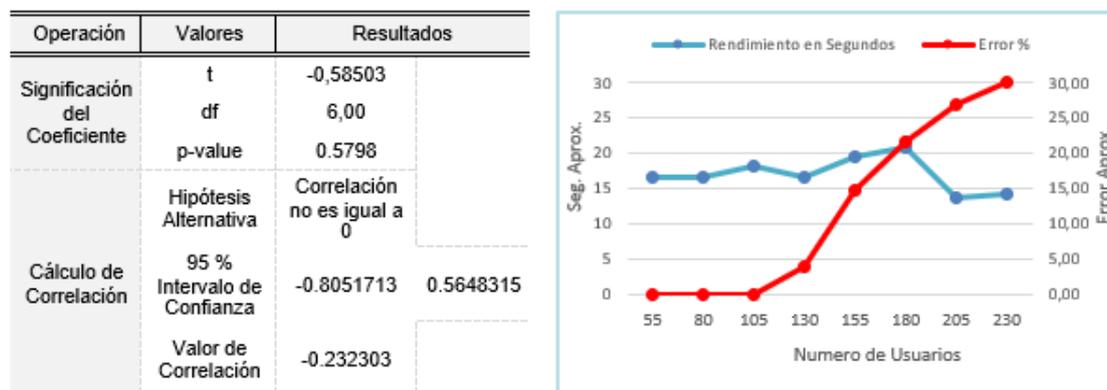


Fig. 47: Resultados y Gráfica de Correlación para Muestra-Rendimiento

Fuente: Propia

• Capas Medianas

Los valores de muestra (Usuarios) para las capas medianas comprimidas que tienen un peso de 100 MB, fueron mínimos por las limitaciones del equipo local, de modo que los resultados que se muestran en la (Fig. 48) para los valores de:

- Muestra – Media: Con un coeficiente de 0,9890859
- Muestra – Desviación: Con un coeficiente de 0,9313714

Operación	Valores	Resultados	
Significación del Coeficiente	t	18,897	
	df	8	
	p-value	6,127	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0,952853	0,997509
	Valor de Correlación	0,9890859	

Operación	Valores	Resultados	
Significación del Coeficiente	t	7,357	
	df	8	
	p-value	8,928	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0,7295858	0,9839776
	Valor de Correlación	0,9313714	

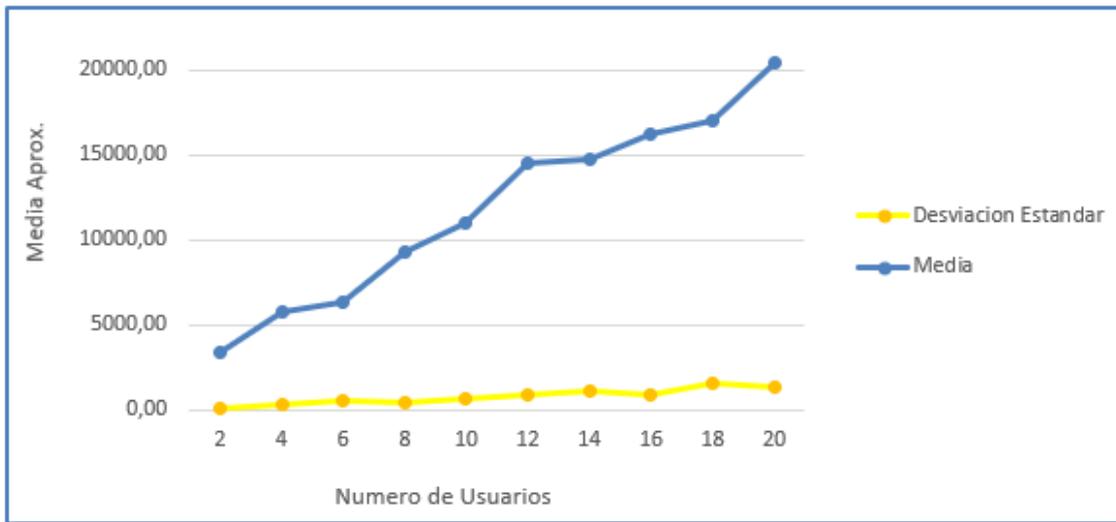


Fig. 48: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)

Fuente: Propia

La correlación es del tipo Alta Positiva para ambas variables, indicando que a mayor número de usuarios (Muestra), los valores de Media y Desviación, además del error con un coeficiente de 0,9585252 (Fig. 49) y su rendimiento con 0,8971348 (Fig. 50) aumentan.

Operación	Valores	Resultados	
Significación del Coeficiente	t	9,5124	
	df	8	
	p-value	1,231	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0,8295318	0,9904203
	Valor de Correlación	0,9585252	

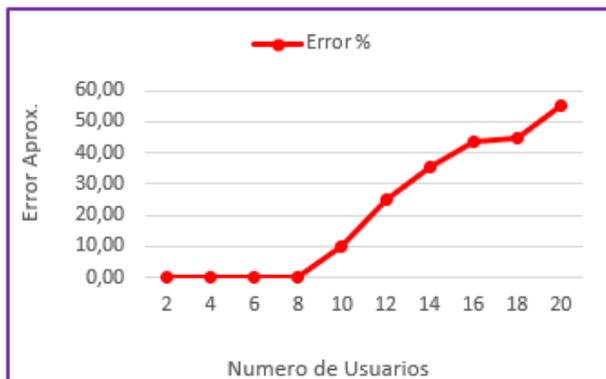


Fig. 49: Resultados y Gráfica de Correlación para Muestra - Error

Fuente: Propia

Operación	Valores	Resultados	
Significación del Coeficiente	t	5,7441	
	df	8	
	p-value	0,00041319	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0,6147636	0,9756537
	Valor de Correlación	0,8971348	

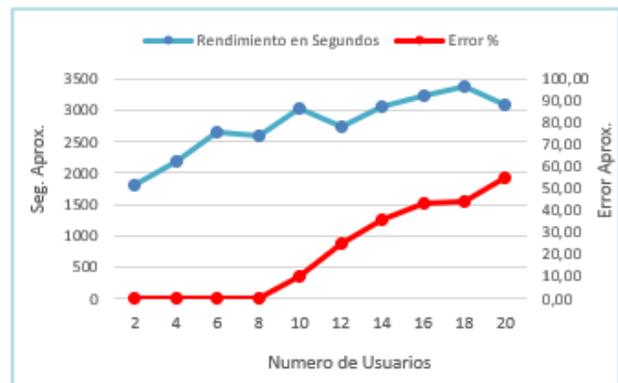


Fig. 50: Resultados y Gráfica de Correlación para Usuarios-Rendimiento

Fuente: Propia

- **Capas Pesadas**

Los resultados de eficiencia entre las capas pesadas y medidas son diferentes, ya que las peticiones de carga varían, debido al peso del comprimido zip, por lo que los valores obtenidos para la relación entre Muestra-Media con 0,9931866 y Muestra-Desviación para 0,608738 demuestran que si el número de Usuarios aumenta (Fig. 51), los Tiempos de Respuesta igualmente, demostrando una relación entre variables.

Operación	Valores	Resultados	
Significación del Coeficiente	T	24,106	
	df	8	
	p-value	9,351	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	0,9703647	0,984474
	Valor de Correlación	0,9931866	

Operación	Valores	Resultados	
Significación del Coeficiente	t	2,1702	
	df	8	
	p-value	0,0618	
Cálculo de Correlación	Hipótesis Alternativa	Correlación no es igual a 0	
	95 % Intervalo de Confianza	-0,03386985	0,997509
	Valor de Correlación	0,608738	

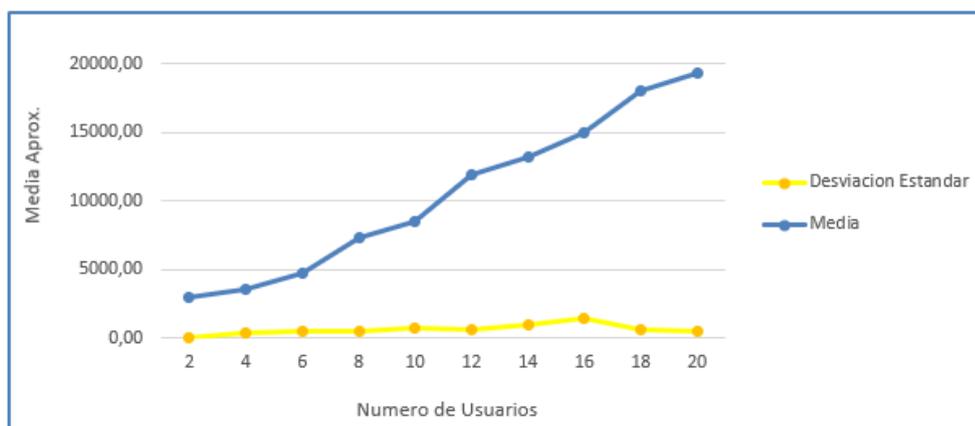


Fig. 51: Resultados y Gráfica de Correlación para (Muestra - Media) y (Muestra - Desviación)

Fuente: Propia

Las relaciones de Muestra – Error con valores de 0,7331978 (Fig. 52) y para los de Muestra – Rendimiento con 0,8369484, dan a entender que a mayor número de Muestras el porcentaje de error del prototipo aumenta y el ritmo de rendimiento del servidor es linealmente ascendente (Fig. 53).

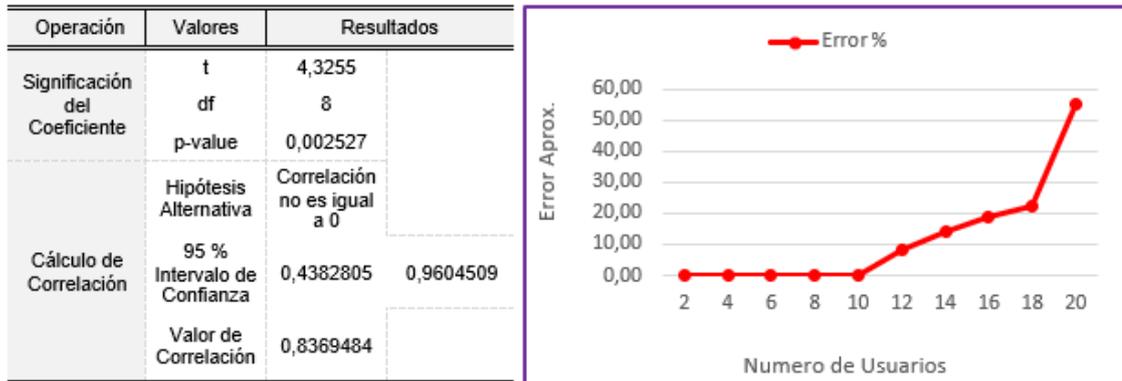


Fig. 52: Resultados y Gráfica de Correlación para Muestra - Error

Fuente: Propia

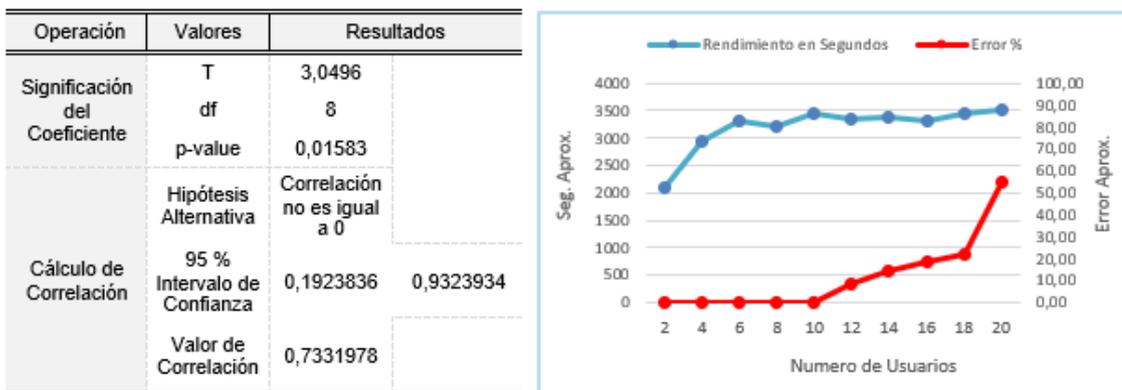


Fig. 53: Resultados y Gráfica de Correlación para Muestra - Rendimiento

Fuente: Propia

CONCLUSIONES

- Por medio de la investigación se pudo comprender de mejor forma, la arquitectura general con la que trabaja una aplicación web SIG y sus diversos componentes geográficos, además de sus servicios y elementos internos que controlan la visualización de los resultados.
- El módulo de carga de capas ráster se encuentra enfocado a controlar los archivos de configuración del visor de mapas P.mapper con el fin de automatizar el proceso de escritura de capas en el archivo plano y que las mismas se visualicen de manera rápida sin que existan problemas de proyección.
- Los formatos admitidos de imágenes para la carga proporcionan una mayor variabilidad de datos para no limitarse a admitir un solo formato de capas
- La función de verificación de los archivos planos geográficos, generados por las herramientas GIS de escritorio, facilita la obtención de datos, mientras que las librerías GDAL de MapServer mejora la visualización de una capa en el visor de mapas.
- Los servicios de capas web implementados para objetos ráster, permiten a un usuario obtener más información sobre los valores de cabecera de una capa, la información del servidor y la imagen visual, para tenerlas como respaldo, u ofrecer un servicio para un cliente que acceda al sistema web de gestión del geo portal.
- La herramienta J-Meter permitió conocer el rendimiento del módulo de carga de capas en un entorno simulado, con resultados positivos, que, a nivel de aplicación, demuestran que el módulo controla el cumplimiento de las peticiones solicitadas
- La Correlación de Pearson, permitió conocer la relación entre los datos obtenidos en la simulación, con el comportamiento de la función de carga del módulo, en diferentes pesos de capa (archivos comprimidos).

RECOMENDACIONES

- Hay que tomar en cuenta la arquitectura, herramientas y metodologías de desarrollo que se establecen para el diseño de una aplicación SIG web, para evitar complicaciones en el diseño de funciones geográficas y para facilitar el uso de las librerías
- Es importante contar con diversas fuentes de información sobre el funcionamiento de herramientas de escritorio GIS cuyas funciones puedan implementarse dentro de un Geo portal
- Para el diseño de una arquitectura GIS versátil, es recomendable usar herramientas de carácter abierto, que permitan aplicar bases de datos No SQL conectadas a un servidor web o local de mapas que cuente con funciones manipulables desde la codificación personal.
- Es necesario tomar en cuenta las funciones geográficas diseñadas en el lenguaje de programación PHP con el fin de poderlas mejorar constantemente y aplicarlas para la mejora de las funciones vectoriales

REFERENCIAS

- Apache Forums. (2017). *XAMPP*. Obtenido de XAMPP: <https://www.apachefriends.org/es/index.html>
- Arce, A. (2018). Programación PHP. *Programación PHP*, 52.
- Avila, L., Delgado, O., & Pacheco, D. (2013). Geopublicacion de estadísticas académicas de instituciones educativas del canton Cuenca. *Universidad del Azuay Facultad de Ciencias de la Administración*, 103.
- Avila, Luis - Delgado, Omar - Pacheco, Diego. (2013). Geopublicacion de estadísticas académicas de instituciones educativas del canton Cuenca. *Universidad del Azuay Facultad de Ciencias de la Administración*, 103.
- Bermejo, E. (2017). *Geoinnova*. Obtenido de Geoinnova: <https://www.geoinnova.org/blog-territorio/infraestructura-de-datos-espaciales-ide-que-es-y-por-que-surge/>
- Bollas, A. M. (2011). Introducción a PostGIS Bases de datos geográficas. *Open Access*, 31. Obtenido de [http://openaccess.uoc.edu/webapps/o2/bitstream/10609/51482/2/Bases de datos geográficas_Módulo2_Introducción a PostGIS.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/51482/2/Bases%20de%20datos%20geogr%C3%A1ficas_M%C3%B3dulo2_Introducci%C3%B3n%20a%20PostGIS.pdf)
- Borja Lopez, Y. (2013). Metodología Ágil de Desarrollo de Software – XP. *Espe Mevast*, 1-10.
- Calvo, D. (2018). *Diego Calvo*. Obtenido de Metodología XP Programación Extrema (Metodología ágil): <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>
- Canós, J., Letelier, P., & Penadés, C. M. (2016). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica administrativa*, 1-8.
- Cascón-Katchadourian, J., Ruiz-Rodríguez, A., & Alberich-Pascual, J. (2018). Uses and applications of georeferencing and geolocation in old cartographic and photographic document management. *El Profesional de la Información*, 27(1), 202.
- Domínguez Bravo, J. (2017). Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica. *GIS and Cartography: An Introductory Overview*, 1-38.
- ESRI. (2016). *ArcGIS for Desktop*. Obtenido de Que es ArcMap?: <http://desktop.arcgis.com/es/arcmap/10.3/main/map/what-is-arcmap-.htm>
- ESRI. (2019). *¿Qué son los datos ráster?* Recuperado el 2019-05-14, de ArcGIS for Desktop: <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>
- Federación Internacional de la Cruz Roja. (2017). Guía básica de referencia QGIS. *Equipo de Manejo de la Información*, 29.
- Flores, E. (2011). *Ingeniería de Software*. Obtenido de Universidad "Unión Bolivariana": http://ingenieriadesoftware.mex.tl/52753_xp---extreme-programing.html
- García, M., Fonts, O., & Gonzales, V. (2013). Geoserver, mas allá de un servidor {WMS}. *Sigte, red de desarrolladores {SIG}*, 1, 18. Obtenido de http://www.slideshare.net/geomati_co/geoserver-paper

- Gutierrez, H. (2015). *Sistemas de Información Geográficos*. Obtenido de Sistemas de Información Geográficos: <https://sites.google.com/site/sigarcgis/4-modelo-vector--raster/ventajas-y-desventajas>
- Infraestructura de Datos Espaciales de España. (2018). Servicio de descarga de coberturas. *GTIDEE*, 6.
- Junta de Andalucía. (2015). *Recursos naturales - Geodiversidad :: Red de Información Ambiental de Andalucía :: Consejería de Medio Ambiente y Ordenación del Territorio :: Junta de Andalucía* :: Obtenido de Junta de Andalucía: <http://www.juntadeandalucia.es/medioambiente/site/rediam/menuitem.aedc2250f6db83cf8ca78ca731525ea0/?vgnnextoid=d42069793c578310VgnVCM1000001325e50aR CRD>
- Junta de Andalucía. (2015). *Sistemas de Información Geografica, tipos y aplicaciones empresariales*. Obtenido de Sistemas de Información Geografica, tipos y aplicaciones empresariales: http://sig.cea.es/tipos_SIG
- Langle, R. (2012). *Sistemas de Información Geográfica*. (R. Langle, Editor, & L. U. Sur, Productor)
- MapServer. (2017). *MapServer - Open Source Web Mapping*. Obtenido de MapServer - Open Source Web Mapping: <https://mapserver.org/>
- Masó, J., Núria, J., & Pons, X. (2010). Historia y estado actual del futuro estándar Web Map Tiling Service del OGC. *5th Spanish SDI Meeting*. Obtenido de http://www.creaf.uab.es/Miramón/publicat/papers/jidee08/WebMapTilingService_Mas oJuliaPons.pdf
- Moreno Alonso, C. (2016). Desarrollo de un modelo de evaluación de ciudades basado en el concepto de ciudad inteligente (Smart city). *Tesis*, 100. Obtenido de <http://oa.upm.es/39079/>
- Naranjo, A. (2017). EVALUACIÓN DEL RENDIMIENTO DE LOS SERVICIOS WMS DE MAPSERVER Y GEOSERVER PARA LA IMPLEMENTACIÓN IDE. 1-98.
- Noe, Y., Elena, H., Morales, C., & Ledesma, F. (2015). Sistemas de Información Geográfica con Qgis 2.x - Nivel I -. *LABORATORIO DE TELEDETECCION Y SIG GRUPO DE RECURSOS NATURALES INTA EEA SALTA*, 1-174.
- OGC. (2017). Web Coverage Service. <https://www.opengeospatial.org/standards/wcs>.
- Organización de las Naciones Unidas. (2017). Agenda 2030 y los Objetivos de Desarrollo Sostenible. 50.
- OSGeoLive. (2015). *GeoServer*. Obtenido de GeoServer: https://live.osgeo.org/es/overview/geoserver_overview.html
- OSGeoLive. (2015). *Quantum GIS (QGIS): SIG de Escritorio*. Obtenido de Quantum GIS (QGIS): SIG de Escritorio: https://live.osgeo.org/archive/5.0/es/overview/qgis_overview.html
- Osorio, H. (2010). *Sistemas de Información Geografica y aplicaciones empresariales*. Obtenido de http://sig.cea.es/tipos_SIG
- Padrón, D. (2011). Practica General Desarrollo de un Servidor de Mapas con Mapserver. *Desarrollo de Servidores de Mapas con Software Libre*, 24.

- Pavón, J. (2016). Instalación y configuración PHP en un servidor Web. *Php*, 34.
- PostGIS. (2016). *PostGIS: Spatial and Geographic objects for PostgreSQL*. Obtenido de PostGIS: Spatial and Geographic objects for PostgreSQL: <https://postgis.net/>
- QGIS. (2013). *DOCUMENTACIÓN DE QGIS2.14*. Obtenido de DOCUMENTACIÓN DE QGIS2.14: https://docs.qgis.org/2.14/es/docs/gentle_gis_introduction/vector_data.html
- Quilca, J. (2019). "INTELIGENCIA DE NEGOCIOS PARA LA INTERPRETACIÓN DE LA INFORMACIÓN GEO-ESTADÍSTICA CONTENIDA EN EL GEOPORTAL UTN. 1-101.
- República Nacional del Ecuador. (2017). Plan Nacional Toda Una Vida. 148.
- Rodríguez Lloret, J., & Olivella, R. (2019). Introducción a los sistemas de información geográfica : conceptos y operaciones fundamentales. 1-32.
- Rodríguez Lloret, Jesús Olivella, Rosa. (2010). Introducción a los sistemas de información geográfica : conceptos y operaciones fundamentales. (39, Ed.) Obtenido de http://ccuc.cbuc.cat/record=b5170112~S23*sp
- Rodríguez, M. (2018). DESARROLLO SOFTWARE DE UNA APLICACIÓN PARA LA REALIZACIÓN DE PROYECTOS DE INVESTIGACIÓN. *e-reding*, 1-265.
- Rodríguez, M. J. (2010). Los Sistemas De Información Geográfica : Una Herramienta De Análisis En Los Estudios De Impacto Ambiental (Eia). *Sociología ambiental*, 34. Obtenido de <https://rua.ua.es/dspace/bitstream/10045/2725/10/cap10.pdf>
<http://rua.ua.es/dspace/bitstream/10045/2725/10/cap10.pdf>
- Servicio Geológico Mexicano. (2017). *gob.mx*. Recuperado el 14 de 05 de 2019, de *gob.mx*: <https://www.sgm.gob.mx/Web/MuseoVirtual/SIG/Introduccion-SIG.html>
- Steinberg, S. S. (2012). Introduction to Geographic Information Systems. En T. McGraw-Hill, *Geographic Information Systems for the Social Sciences: Investigating Space and Place* (pág. 16). doi:10.4135/9781452239811.n2
- Tuesta, R., Rengifo, J., & Bravo, N. (2011). ArcGIS Basico. 1-148.
- Universidad Autónoma de Juarez. (2015). 2 . ¿ Qué son los Sistemas de Información Geográfica ? *Hoja Técnica de divulgación No.8*, 3. Obtenido de http://www.uacj.mx/ICB/UEB/Documents/Hojas_tecnicas/SIG.pdf
- Universidad de Alicante. (2015). *Servicio de Informática ASP.NET MVC 3 Framework*. Obtenido de Servicio de Informática ASP.NET MVC 3 Framework: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Universidad Técnica del Norte. (2018). *INFRAESTRUCTURA DE DATOS ESPACIALES*. Obtenido de UNIVERSIDAD TECNICA DEL NORTE: http://geoportal.utn.edu.ec/ide_utn/index.php?a=Index#
- Universitat Politècnica de Valencia. (2018). *UPV Innovacion*. Obtenido de <http://i2t.webs.upv.es/2-serv-upv/serv-upv-7/links/anexotec-ej.pdf>

Veintimilla Reyes, J., & Avila Larrea, F. (2015). Análisis e implementación de una infraestructura de datos espaciales (IDE). Caso de estudio: Gobierno Autónomo Descentralizado Municipal del cantón Guachapala. *Revista Tecnológica ESPOL-RTE*, 79-99.

Zea Ordóñez, M. P., Molina Ríos, J. R., & Redrován Castillo, F. F. (2017). Administración De Bases De Datos Con Postgresql. *Administración De Bases De Datos Con Postgresql*, 82. Obtenido de <https://www.3ciencias.com/wp-content/uploads/2017/04/Administraci%C3%B3n-bases-de-datos.pdf>