

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Módulo de análisis espacial estadístico descriptivo para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas (SIG UCI).”

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores:

Elibetsy Herrera Chong
Lisaidi Lucía Coca Conesa

Tutor:

MSc. Daniel Echevarría González
Ing. Alain León Companioni

La Habana, junio, 2015
“Año 57 de la Revolución”

Declaración de autoría.

Declaramos que somos los únicos autores del trabajo “Módulo de análisis espacial estadístico descriptivo para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas” y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, con carácter exclusivo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Autora: Elibetsy Herrera Chong

Autora: Lisaidi Lucía Coca Conesa

Tutor: M. Sc. Daniel Echevarría González

Tutor: Ing. Alain León Companioni

Datos del contacto.

Autoras:

Lisaidi Lucía Coca Conesa

Dirección: Paseo Agramonte, carretera Casilda km#1, #13, Trinidad, Sancti Spíritus, Cuba.

Teléfono (celular): +53 53111169

Correo Electrónico: llcoca@estudiantes.uci.cu

Elibetsy Herrera Chong

Dirección: Borrego #4, Apt 3altos, Entre Saravia y Consejero Arango, Cerro, La Habana, Cuba.

Teléfono (celular): +53 53406092

Correo Electrónico: echong@estudiantes.uci.cu

Tutores:

M. Sc. Daniel Echevarría González

Especialidad: Geografía

Correo Electrónico: danielec@uci.cu

Categoría docente: Asistente

Ing. Alain León Companioni

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI),
graduado en el año 2010.

Correo Electrónico: acompanioni@uci.cu

Categoría docente: Especialista "A" en Ciencias Informáticas.

Dedicatoria:

A nuestros padres...

Agradecimientos:

A mi mamita y a mi papito. Gracias por traerme al mundo, por guiarme, por malcriarme, por respetar mis decisiones. Gracias por su apoyo incondicional, por tener la capacidad de saber ser padres y amigo. Por exigirme y a la vez relajarme, por darme la oportunidad de escoger, por enseñarme lo bueno, lo malo, las piedras y lo importante de los tropezones. Gracias por ser tan buen ejemplo ante todo, porque son lo máximo y lo saben. Porque si Dios me diera la oportunidad de volver a nacer, solo la tomaría si en el paquete vinieran ustedes de nuevo. Los amo, los adoro y me quedo corta.

A mis hermanitos Lisbetsy y Carlos Alexander, que aunque la sangre no sea de un 100%, el corazón lo es de 1 000 000.

A papi Ale y a mami Livi, que me tratan como una hija y yo los quiero como padres.

A mi tío Héctor, que estuvo días aprendiendo Openlayer, jajajaja. Que siempre me ha tratado como una hija, preocupándose por mí, siendo ejemplo a seguir (y muy bueno).

A mi tía Elsitá, que de tantos años en la capital la he visitado poco, pero ella siempre se desespera por mí. Gracias por siempre estar pendiente. Te quiero mucho.

A mi tía Katia, que sigue siendo la misma. Gracias por ayudarme y quererme tanto. Pero te agradezco más que quieras tanto a mi papá y se lo demuestres. Un beso.

A mí tío Ore gracias por guiarme. No se me olvidan nuestras conversaciones.

A toda mi familia, que es bien grande, y la hoja no me alcanza para ponerlos a todos. No obstante todos saben que los quiero y que les agradezco su preocupación y que me hayan malcriado siempre.

A Davi porque estuvo desde el principio conmigo y me guió, me comprendió, me mortificó jajaja, pero siempre me apoyó. Gracias por vivir estas etapas conmigo y por ser, ante todo, mi amigo.

A mi tutor Alain, que aunque daba trabajo dar con él, siempre estuvo ahí. No importaba la hora ni lo que estuviera haciendo.

A mi tutor Daniel por su preocupación y por siempre estar disponible.

A mis amigas Lisaidi y Arlet, que siempre me han apoyado, escuchado, peleado, alimentado y malcriado durante los cinco años.

A nuestro antiguo tutor Adrián, que aunque nos abandonó, nos dejó en buenas manos.

A Rauli que desde el primer día hasta el último siempre se preocupó por mí y me ayudó mucho.

A Yosbel, por su confianza en mí, por hacerme ver de lo que era capaz.

Agradecimientos

A Ángel, por sus críticas constructivas, por su paciencia y por su tiempo. A Yeni por prestármelo jajaja.

A Haniel, por las horas de dedicación.

A los profesores del proyecto, que me atendieron siempre. En especial a Miosotis y Alejandro.

A mis programadores favoritos Jairo, Ronald, Serguey y Guille, que me sacaron de apuros en muchas ocasiones.

A mis compañeros de tertulias cafeteras Yulina, Armando, Jorgito, Yerandy y Lázaro. Por todos los buenos momentos que me hicieron pasar.

A mis compañeros de mesa Vidal y Pedro Manuel, que me aguantaron y me ayudaron mucho.

A Yordan Anglada porque sin conocerme me brindó su apoyo.

A Maikel, por su sacrificio, le estaré agradecida siempre.

A mis vecinos del barrio, que siempre se han preocupado por mí.

A mis amistades, que son muchas, gracias por los momentos compartidos, por el apoyo durante toda la carrera. Hoy estoy aquí, en gran parte, gracias a ustedes.

Elibetsy Herrera Chong

Agradecimientos:

A mi mami y a mi papi lindo por todo lo que hacen por mí, por ser los mejores padres del mundo, por quererme, por toda la preocupación y atención, por permitirme todas las malcriadeces y apoyarme en cualquier decisión que yo tomase. Por dejarme aprender de los errores, por siempre estar, muchas gracias mis amores.

A mis hermanos Alexey y Yosiel, por llenarme de tantas alegrías, por quererme y entenderme.

A mis abuelos, por la infinita dedicación y orgullo.

A mis tías, en especial a Nancy, Elisa, Esmirna, a mis tíos, a todos mis primos, en especial a mi prima Yadamirki, a quien le deseo todo lo bueno de este mundo y que siga hacia delante, que solo Dios sabe porque hace las cosas.

Y en general a mi maravillosa familia que me ha dado todo su amor y ha confiado siempre en mí.

A mis mejores amigas: "Las chicas superpoderosas", Arlet y mi compañera de tesis Eli, gracias por estar siempre, por compartir su amistad, por no dejarme sola, por soportar todos mis pleitos, que todos eran por quererlas tanto. Gracias por los consejos y el cariño.

A mi novio Alejandro por regalarme esos momentos maravillosos que he vivido a su lado a pesar del poco tiempo junto y brindarme siempre su comprensión y cariño en los momentos difíciles, y por siempre hacer de mí, una mejor mujer. Te amo.

A mis suegros, en especial a Merci por su preocupación, por acogerme como una hija más y a toda esa maravillosa familia.

A mi mejor amigo Yaikel, gracias por estar siempre, por hacerme reír, por apoyarme y por los buenos consejos. Te quiero.

A mis compañeros de aula durante los 5 años de la carrera, en especial a Jorge Luis (el pollo), Alejandro, Daryl, en fin a todos muchas gracias por estar.

A mi primer tutor Adrián Gracia, que nos apoyó mucho en la primera etapa de la investigación, cuando no sabíamos por dónde empezar.

A mis dos tutores Alain y Daniel que continuaron guiándonos por el camino correcto para llegar a donde estamos, muchas gracias por la dedicación y el empeño en que las cosas se hiciesen bien.

A todos los integrantes del proyecto Aplicativos SIG que contribuyeron al desarrollo del sistema, en especial a Miosotis, Listley, Alejandro.

A Haniel por estar cuando lo necesitaba y por su dedicación.

En fin a todos los que de una forma u otra contribuyeron y ayudaron en la realización de este trabajo. Muchas gracias.

Lisaidi Lucía Coca Conesa

Resumen

El presente trabajo se centra en el desarrollo de un módulo para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas. Permite la representación temática de capas en el mapa a partir de datos estadísticos del área residencial estudiantil de dicha universidad.

Con el despliegue de esta solución se suple la necesidad de un módulo para el sistema existente, que represente la información asociada a indicadores estadísticos. Se corrigen los errores en la información manejada por los directivos del área residencial estudiantil, y se facilita el proceso de toma de decisiones de estos.

Previo a la implementación se realiza un estudio de los distintos tipos de análisis espacial, teniendo en cuenta varios autores, seleccionando el tipo estadístico descriptivo. Además se llevó a cabo un estudio de algunas de las herramientas informáticas que realizan en la actualidad este tipo de análisis.

Como resultado del trabajo se obtuvo un módulo de análisis espacial estadístico descriptivo, que permite la representación de capas en colores y gráficas.

Palabras claves: Análisis espacial, descriptivo, estadístico, módulo, Sistema de Información Geográfica (SIG).

Abstract

This work focuses on the development of a module for the Geographic Information System of the University of Informatics Sciences. Allows the thematic representation of map layers from statistical data of students residential area in the university.

With the deployment of this solution, it is supplied the need of a module for the existing system, that is representing the information associated with statistical indicators. The errors in the information used by the residential area managers were corrected.

Prior to implementation, a study of different types of spatial analysis is performed taking into account several authors, selecting descriptive statistical type. Also, it was conducted a study of some in informatics tools that currently carry out this type of analysis.

As a result of this work, a module of descriptive statistical spatial analysis was obtained, which allows the representation of layers in colors and graphics.

Keywords: Spatial analysis, descriptive, statistics, module, Geographic Information System (GIS).

Contenido

Resumen	VII
Abstract	VII
Introducción.	1
Capítulo 1: Fundamentación Teórica.	5
1.1. Introducción.	5
1.2. Datos espaciales.	5
1.3. Información Geográfica.....	6
1.4. Sistema de Información Geográfica (SIG).....	6
1.5. Análisis Espacial.....	7
1.6. Estadística Descriptiva.....	10
1.7. Soluciones existentes.	13
Servicio de Mapas Temáticos (WMT) de la UCI.....	13
1.8. Metodología y herramientas a utilizar.	18
1.7.1. Metodología de desarrollo.....	18
1.7.2. Lenguaje Unificado de Modelado (UML) v2.0.....	20
1.7.3. Herramienta de modelado (Visual Paradigm) v8.0.....	21
1.7.4. Servidor Web (Apache) v2.2.	21
1.7.5. Marco de trabajo (Symfony) v2.3.5.	22
1.7.6. Biblioteca (Jquery) v2.0.....	24
1.7.7. Librería (OpenLayers) v2.12.	24
1.7.8. Servidor de Mapa (MapServer) v6.0.1.....	25
1.7.9 Sistema Gestor de Base de Datos (PostgreSQL) v9.1.	25
1.7.10 Extensión (PostGIS) v1.5.3.	26
1.7.11 Entorno de Desarrollo Integrado (NetBeans) v8.0.....	26
1.9. Conclusiones Parciales.....	27
Capítulo 2: Presentación de la Solución Propuesta.....	28
2.1. Introducción.	28

2.2.	Modelo de dominio.	28
2.2.1.	<i>Diagrama de clases del Modelo de Dominio.</i>	29
2.2.2.	<i>Definición de las clases del Modelo de Dominio.</i>	29
2.3.	Técnica de captura de requisitos.	30
2.4.	Especificación de Requisitos.	30
2.4.1.	<i>Requisitos funcionales.</i>	31
2.4.2.	<i>Requisitos No Funcionales (RNF).</i>	32
2.5.	Modelado del sistema.	34
2.5.1.	<i>Definición de los actores que interactúan con el sistema.</i>	34
2.5.2.	<i>Definición de CU del sistema.</i>	34
2.5.3.	<i>Diagrama de CU del sistema.</i>	35
2.5.4.	<i>Descripción textual del CUS: Realizar tematización según tipo de estilo.</i>	35
2.6.	Conclusiones parciales	41
Capítulo 3:	Construcción de la solución propuesta.....	42
3.1.	Introducción.	42
3.2.	Descripción de la arquitectura.....	42
3.2.1.	<i>Patrón arquitectónico Cliente-Servidor.</i>	42
3.2.2.	<i>Patrón arquitectónico Modelo-Vista-Controlador (MVC).</i>	43
3.2.3.	<i>Patrones de diseño.</i>	45
3.3.	Modelo de diseño.	46
3.3.2.	<i>Descripción de las clases.</i>	47
3.4.1.	<i>Diagrama de Clases Persistente.</i>	48
3.4.2.	<i>Modelo Físico de la Base Dato.</i>	49
3.5.	Modelo de Despliegue.	49
3.6.	Modelo de Componentes.....	50
3.7.	Estándares de codificación.	51

3.7.1. Notación Camello.....	51
3.8. Proceso de pruebas del Módulo de Análisis Espacial Estadístico Descriptivo para el SIG UCI...	52
3.8.1. Pruebas Funcionales o de caja negra.....	52
3.8.2. Casos de Pruebas (CP).....	52
3.8.3. Pruebas de Rendimiento.....	57
3.8.4. Prueba de Carga y Estrés.....	57
Conclusiones.....	59
Conclusiones Generales.....	60
Recomendaciones.....	61
Bibliografía y Referencias Bibliográficas.....	62
Anexos.....	67

Índice de Tablas

Tabla # 2.1: Definición del actor del sistema.	34
Tabla # 2.2: Definición de CU: Realizar tematización según tipo de estilo.	34
Tabla # 2.3. Diagrama de CU Módulo de análisis espacial estadístico descriptivo para el SIG UCI.	35
Tabla # 2.4: Descripción textual del CU: Realizar tematización según tipo de estilo.	35
Tabla # 3.1: Prueba del CU Realizar tematización según tipo de estilo, secciones a probar.	53
Tabla # 3.2: Descripción de las variables.	56
Tabla # 3.3: Resultados de la pruebas de carga.	58

Índice de Figuras

Fig. # 1.1: Esquema que muestra los flujos de trabajo y las fases de AUP.	19
Fig. # 2.1: Diagrama de clases del dominio.....	29
Fig. # 3.1: Esquema simplificado de la arquitectura interna de Symfony2.....	44
Fig. # 3.2: Diagrama de clases del CU: Tematizar según tipo de estilo.	47
Fig. # 3.3: Diagrama de Clases Persistente de la solución.....	48
Fig. # 3.4: Modelo físico de la base datos.	49
Fig. # 3.5: Diagrama de Despliegue de la Solución Propuesta.....	50
Fig. # 3.6: Diagrama de Componentes del CU Tematizar según tipo de estilo.	51
Fig. # 3.7: Ejemplo de Notación Camello.	51
Fig. # 3.8: Resultados de las pruebas funcionales.	57

Introducción.

Los Sistemas de Información Geográfica (SIG) han desempeñado un papel fundamental en la representación y análisis de la información geográfica, debido a que proveen los medios necesarios para capturar, organizar, manipular y hacer uso de la información (Olaya, 2011). Así, estas aplicaciones permiten a personas y organizaciones analizar hechos y oportunidades; además de resolver problemas y conflictos utilizando información proveniente de un rango amplio de disciplinas, convirtiéndose en una herramienta fundamental de análisis y toma de decisiones.

Los Sistemas de Información Geográfica hacen uso del análisis espacial para entender aquella información que ha sido recopilada, cartografiada e integrada en el sistema, con el objetivo de estudiarla (Pérez, 2014). El análisis espacial es uno de los aspectos más interesantes y destacables de los SIG; pues permite derivar nueva información de los datos existentes para mejorar la toma de decisiones. En la actualidad este análisis en los SIG es aplicado a disímiles usos, ejemplo, en empresas de marketing, para anticipar y administrar cambios en el mercado; en el gobierno, para ayudar al cumplimiento de la ley sobre la delincuencia; durante inundaciones o huracanes, mediante la gestión de desastres, entre otros.

En Cuba el desarrollo de los SIG también ha sido aprovechado, se han implementado soluciones con el objetivo de mejorar el proceso de toma de decisiones y el ahorro de recursos a diversas empresas. Entre algunos ejemplos de estas soluciones, se encuentra el Sistema de Información Geográfica dedicado a realizar estudios de impacto ambiental en la Empresa Eléctrica Provincial de Santiago de Cuba. Este sistema permite tener una base de datos y realizar proyectos con programas de reducción de CO₂¹ y otros contaminantes para cada zona donde se encuentren ubicados los grupos electrógenos y poder tener producciones de energía más limpia. Otro ejemplo es el sistema para el manejo agroecológico de variedades de Tabaco (nicotina Tabacum.L) en la Empresa Tabaco del municipio de San Luis de la Provincia de Pinar del Río, sistema encargado de determinar las áreas con mayores aptitudes físico-geográficas, además de caracterizar desde el punto de vista fitosanitario los suelos dedicados al cultivo del tabaco.

En el Centro Geoinformática y Señales Digitales (GEYSED) de la Universidad de las Ciencias Informáticas (UCI), la Línea de Productos de Software (LPS) Aplicativos SIG, es el grupo encargado de la construcción y personalización de SIG sobre plataformas Web y diferentes tipos de negocios.

Entre los principales productos desarrollados por la LPS, se encuentra el Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas (SIG UCI), diseñado bajo un esquema diferente al

¹ CO₂: Dióxido de carbono, compuesto natural que es esencial para el ciclo biológico de las plantas y cosechas.

desarrollo de otros sistemas similares que se implementan en la LPS, definido por la utilización de herramientas y tecnologías no aprovechadas anteriormente por el equipo de desarrollo.

Este permite la visualización, localización y consulta de objetivos socioeconómicos de la UCI, edificios docentes (de residencia), plazas, manzanas, consultorios, cajeros, áreas deportivas, cafeterías, entre otros. Sin embargo no cuenta con un mecanismo de análisis que represente la información asociada a indicadores estadísticos; potencialidad de los SIG que no está implementada actualmente en el SIG UCI.

Por otro lado, los directivos del área residencial de la UCI presentan dificultades en el proceso de toma de decisiones, debido a problemas en la recopilación de los datos que estos manejan. Esta selección de información es costosa en tiempo y esfuerzo ya que se realiza mediante correo o teléfono, induciendo muchos errores, además de la duplicación de la información; principalmente en los datos que se actualizan constantemente como las capacidades en los edificios, teniendo como consecuencia que se envíen estudiantes para apartamentos donde no hay capacidades, entre otras incidencias por errores en la información.

Por lo anteriormente expuesto se plantea como **problema a resolver** ¿Cómo facilitar el proceso de toma de decisiones de los directivos que laboran en el área residencial estudiantil de la Universidad de las Ciencias Informáticas?

Para ello se hace necesario el **estudio** del análisis espacial en los SIG, **centrando la atención** en el análisis espacial estadístico descriptivo en el SIG UCI.

Se define como **objetivo general** de la investigación: desarrollar un módulo de análisis espacial estadístico descriptivo para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas que apoye el proceso de toma de decisiones de los directivos de la residencia estudiantil. Para dar cumplimiento al objetivo general se trazaron los siguientes objetivos específicos:

1. Caracterizar los Sistemas de Información Geográfica que hacen uso del análisis espacial estadístico descriptivo.
2. Seleccionar la metodología para el desarrollo del módulo.
3. Presentar la propuesta de solución.
4. Desarrollar la propuesta de solución.
5. Comprobar el funcionamiento del módulo desarrollado.

Para guiar la investigación se definen las siguientes preguntas de investigación:

1. ¿Cuáles son las características de los Sistemas de Información Geográfica que hacen uso del análisis espacial?
2. ¿Cuáles son las condiciones en que se desarrollan los Sistemas de Información Geográfica en la UCI?
3. ¿Cómo seleccionar la metodología para el desarrollo del módulo de análisis espacial?
4. ¿Cómo organizar el proceso de desarrollo del módulo de análisis espacial estadístico descriptivo?
5. ¿El funcionamiento del módulo desarrollado permite facilitar el proceso de toma de decisiones de los directivos que laboran en el área residencial estudiantil de la UCI?

Resultados esperados.

Con la presente investigación se pretende obtener:

1. Un módulo de análisis espacial estadístico descriptivo para facilitar la toma de decisiones mediante la representación de la información a través de tematizaciones en colores y gráficas.
2. La documentación técnica asociada al proceso de desarrollo del módulo.

Para el desarrollo de la investigación se utilizaron varios métodos científicos que permiten obtener una idea más detallada de lo que se quiere lograr.

Los métodos teóricos:

- **Analítico-Sintético:** Se emplea para realizar un análisis de la bibliografía correspondiente a la investigación y extraer los elementos significativos para el diseño e implementación del módulo.
- **Análisis Histórico-Lógico:** Se investiga y se extraen aspectos de interés sobre los Sistemas de Información Geográfica existentes en la Universidad, a nivel nacional e internacional.
- **Modelación:** Se modelan las actividades que se desarrollan en el transcurso de la elaboración de la solución a través de diagramas, lográndose de este modo la comprensión del módulo a desarrollar.

Los métodos empíricos:

- **Entrevistas:** Se realizan entrevistas a varios directivos del área residencial estudiantil de la UCI, para recoger toda la información necesaria sobre las nuevas funcionalidades a desarrollar.

El Trabajo de Diploma se encuentra estructurado en 4 secciones: resumen, introducción, desarrollo y conclusiones. El desarrollo a su vez está compuesto por 3 capítulos donde se detalla el proceso investigativo realizado:

Capítulo 1: Fundamentación Teórica.

En este capítulo se definen los conceptos fundamentales asociados al dominio del problema, para una mayor comprensión del mismo. Se analizan soluciones existentes que pueden aportar elementos para la solución del presente trabajo y se describen las herramientas a utilizar en la construcción del módulo.

Capítulo 2: Presentación de la Solución Propuesta.

En este capítulo se exponen los requisitos, tanto funcionales como no funcionales, que debe cumplir el sistema. Además se describe la propuesta de solución a través de los actores, casos de uso.

Capítulo 3: Construcción de la Solución Propuesta.

En este capítulo se aborda la arquitectura sobre la que se realizará el diseño e implementación de la solución, definiendo los patrones de arquitectura y de diseño que se utilizan, permitiendo definir la estructura general del software, así como el diseño del mismo. Se presentan los Diagramas de Clases del Diseño, además del Diagrama de Despliegue y de Componentes. Concluyendo con las pruebas al software con el objetivo de comprobar el correcto funcionamiento de los requisitos funcionales propuestos para el módulo.

Capítulo 1: Fundamentación Teórica.

1.1. Introducción.

En este capítulo se definen los conceptos fundamentales asociados al dominio del problema, para una mayor comprensión del mismo. Se brinda una introducción al desarrollo de los datos espaciales, la información geográfica, los SIG de manera general hasta caer en el análisis espacial y sus derivados. Se analizan soluciones existentes que pueden aportar elementos para la solución del trabajo y se describen las herramientas a utilizar en la construcción del módulo.

1.2. Datos espaciales.

Los datos espaciales son los datos geográficos o de ubicación que permiten caracterizar un territorio, entre otros, por sus límites políticos administrativos, ríos, vías de comunicaciones, datos altimétricos e instituciones de salud.

Se entiende por dato espacial todo aquel que tiene asociada una referencia geográfica, de tal modo que se pueda localizar exactamente donde sucede un evento dentro de un mapa. Dentro de esta definición se incluyen datos de campos (superficies) o datos asociados a objetos como puntos, líneas o polígonos. Es decir, todo cuanto puede recogerse según los distintos modelos de representación (Olaya, 2011).

Los datos espaciales refieren a entidades o eventos que cumplen los siguientes principios básicos:

- ✚ Tienen posición absoluta: sobre un sistema de coordenadas (x, y, z).
- ✚ Tienen una posición relativa: frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado).
- ✚ Tienen una figura geométrica que las representan (punto, línea, polígono).
- ✚ Tienen atributos que lo describen (características del elemento o fenómeno) (Olaya, 2011).

En la presente investigación se definen los datos espaciales como objetos que se encuentran representados geográficamente en el mapa, es decir, datos que tomando diversas estructuras representan cada uno de los elementos que se ubican en el mapa, ejemplo ríos, que son representados mediante polígonos por lo cual tiene como vértices x, y, y h; las carreteras, que tienen un punto de origen y otro de destino, ancho, material de cobertura, capacidad soportante, ancho de la cuneta, intersección con otras carreteras, entre otros.

1.3. Información Geográfica.

Según (Olaya, 2011) (...) “La información es el resultado de un dato y una interpretación.” El propio autor establece que la información geográfica tiene dos componentes: una temática y otra geográfica, estas van unidas y conforman una unidad única de información geográfica, aunque pueden separarse y analizarse por separado. La componente espacial responde a la pregunta ¿dónde? y la componente temática responde a la pregunta ¿qué?

Además de las componentes: espacial y temática (Sinton, 1978) añade la componente temporal y propone un esquema sistemático que permite clasificar en grupos las distintas clases de información geográfica. Según este esquema, cada una de estas componentes puede estar en uno de los siguientes tres estados posibles: fija, controlada y medida.

Otra definición de este concepto lo emitió Esteban Gianfelici, el cual denomina información geográfica a los datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales (Gianfelici, 2011).

Por lo tanto, se define en la investigación como la interpretación realizada a los datos espaciales. Esta información está dividida en dos partes que funcionan como unidad, por un lado se encuentra la parte espacial, que responde a su localización ¿dónde? y la parte restante es la temática que responde a la caracterización física ¿qué?

1.4. Sistema de Información Geográfica (SIG).

El profesor David Rhind ha definido un Sistema de Información Geográfica como un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados, para la solución de los problemas complejos del manejo y planeamiento territorial (Rhind, 1989).

También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información (Yagüez, y otros, 2002). Los SIG fueron diseñados para trabajar con datos referenciados mediante coordenadas espaciales o geográficas. En otras palabras, un SIG es tanto un sistema de base de datos con capacidades específicas para datos georreferenciados, como un conjunto de operaciones para trabajar con esos datos (Olaya, 2011).

Básicamente, un SIG ha de permitir la realización de las siguientes operaciones:

- ✚ Lectura, edición, almacenamiento y, en términos generales, gestión de datos espaciales.

- ✚ Análisis de dichos datos. Esto puede incluir desde consultas sencillas a la elaboración de complejos modelos, y puede llevarse a cabo tanto sobre la componente espacial de los datos (la localización de cada valor o elemento) o la componente temática (el valor o el elemento en sí).
- ✚ Generación de resultados tales como mapas, informes, gráficos (Olaya, 2011).

Un SIG está integrado por un conjunto de subsistemas donde convergen aspectos muy tecnológicos con otros más tradicionales y estrechamente relacionados entre sí, en los que cada uno de ellos posee una identidad propia, pero el conjunto de todos genera una herramienta de potencial extraordinario. Es por eso que se divide en 5 componentes según (Goodchild, y otros, 2005):

- ✚ **Hardware:** Equipo de cómputo en el que opera el SIG, que actualmente puede ir desde computadoras centralizadas hasta individuales o de red.
- ✚ **Software:** El software resulta un componente SIG de gran importancia, debido a que es el encargado de proveer las funciones y las herramientas que se necesitan para almacenar, analizar y mostrar la información geográfica, ejemplo de esto es ArcGIS, sistema integrado por un grupo de software encargados de la creación de SIG, reconocidos a nivel mundial.
- ✚ **Información o datos:** Los datos son el principal activo de cualquier sistema de información. Por ello el éxito y la eficacia de un SIG, se miden por el tipo, la calidad y la vigencia de los datos con los que opera. Es muy importante para el desarrollo de estos sistemas que se cuente con buenos datos de base.
- ✚ **Personal:** La tecnología de los SIG está limitada si no se cuenta con el personal que opera, desarrolla y administra el sistema; y que establece planes para aplicarlo en problemas del mundo real.
- ✚ **Métodos:** Los métodos son los planes y las reglas de la actividad que, bien diseñados, contribuyen al desarrollo exitoso de los SIG.

En la investigación se definen los SIG como un sistema de apoyo a la toma de decisiones integrado por un conjunto de datos georreferenciados, así como también por componentes para la gestión y representación de dichos datos, con el objetivo de resolver problemas complejos de planificación y gestión.

1.5. Análisis Espacial.

(Sendra, 2004) define el análisis espacial como “el conjunto de procedimientos de estudio de los datos geográficos, en los que se considera de alguna manera, sus características espaciales”.

Según (R.J, 1972) el análisis espacial es el conjunto de técnicas y modelos que hacen un uso explícito; de la referencia espacial de cada dato particular.

Los autores (González del Bosque, y otros) conceptualizan el análisis espacial como una colección de procesos cuyo fin es explotar los datos geográficos. Establecen tres tipos de funcionalidades de análisis:

1. las que se centran en la componente temática.
2. las que se centran en la componente espacial.
3. las que analizan las dos componentes por igual.

Este análisis se puede efectuar de diferentes formas, las cuales se encuentran expuestas a continuación:

Análisis descriptivo de la componente temática:

Se realiza mediante la cuantificación de los datos espaciales a través de los equivalentes de la estadística clásica en tipo especial de datos; incluyendo aquí descriptores de centralidad (como el centro medio, el centro de gravedad) y dispersión (como la distancia típica o el radio dinámico), de dependencia espacial o el estudio de patrones espaciales, entre otros.

Análisis de mapas de puntos:

Consiste en una serie de puntos en el espacio (ejemplo: ubicación de individuos, presencia o ausencia de una determinada variable en el territorio) que contienen una información relevante acerca del objeto de la investigación. Esta distribución puede ser de tipo aleatoria, regular o agregada. En este tipo de análisis es muy importante el tamaño del área de estudio para detectar patrones a escalas diferentes. Aquí se realizan análisis estadísticos como el análisis de cuadrantes aplicando pruebas como la de Kolmogorov Smirnov (dividiendo la zona de estudio en unidades regulares y estudiando el número de puntos que aparecen en cada una), la función K de Ripley (que trata de incorporar la escala como elemento de análisis) y el método del vecino más próximo (deduce el tipo de estructura en una distribución aleatoria de acuerdo a la distancia de cada punto a su vecino más cercano).

Superposición de mapas vectoriales:

Implica resolver tanto la parte geométrica como la temática en la capa de salida. Se realiza superponiendo capas como líneas, puntos, y las más complejas: polígonos.

Cálculo de distancias y análisis de proximidad:

Está compuesto por el análisis de vecindad o proximidad, conocido también como generación de áreas de influencia. Es un tipo de consulta espacial muy complejo porque ha de dibujar polígonos, generar topologías. Este análisis responde a preguntas del tipo ¿qué porcentaje de yacimientos se halla a menos de 2 Km de

distancia de la línea de la costa? ¿Qué instalaciones recaen en el municipio X y además están a menos de 50 m de una carretera?

✚ **Análisis de redes:**

En este tipo de análisis se manifiestan los cálculos de caminos mínimos en una red. También se encuentran las redes de flujo, utilizadas para representar el movimiento de bienes entre dos lugares, teniendo en cuenta la dirección y el volumen. Es muy empleado además en el caso de los estudios localización asignación, que persiguen asignar la localización óptima de una instalación (hospital, farmacia, banco), conociendo tanto su distribución espacial como los elementos que influyen en su utilización.

Víctor Olaya en su libro (Olaya, 2011) confecciona una definición menos formal de este proceso, conceptualiza el análisis espacial sencillamente como “el conjunto de operaciones que se desarrolla en base a los datos espaciales en el trabajo habitual con estos”, y establece algunos tipos de análisis espacial, entre ellos:

- ✚ **Consulta Espacial:** Es el tipo de análisis básico y proporciona información inmediata a partir de una simple observación de los datos.
- ✚ **Análisis Topológico:** Las consultas hechas a las capas de datos espaciales pueden tener relación, no solo con su posición, sino con otros elementos de la misma capa.
- ✚ **Transformaciones:** Una transformación común es la creación de áreas de influencia. Se tiene así una transformación geométrica, pues la forma del objeto se transforma en una nueva estructura que indica la zona que se ve afectada por dicho objeto. Otro tipo de transformación es la simplificación de líneas, que trata de definir los mismos trazados de un conjunto de líneas reduciendo el número de puntos empleados.
- ✚ **Análisis de superficies:** Este análisis es uno de los más potentes. Aunque este análisis de superficies se entiende como el de la superficie terrestre (es decir, el relieve), gran parte de estas operaciones pueden aplicarse a cualquier otra superficie.
- ✚ **Estadística Descriptiva:** Se incluyen aquí descriptores de centralidad y dispersión, de dependencia espacial o el estudio de patrones espaciales, entre otros muchos. Estos pueden a su vez usarse para el contraste de hipótesis que contengan cierta componente espacial.
- ✚ **Inferencia:** Permite inferir comportamientos de las distintas variables y estudiar la forma en que estas van a evolucionar a lo largo del tiempo.
- ✚ **Toma de decisiones y optimización:** La estructura de la información geográfica en capas dentro de un SIG es muy favorable para estudiar de forma combinada los efectos de distintos factores. El

estudio de estos factores puede ser una herramienta clave para tomar decisiones relativas a la actividad sobre la que ejercen su influencia.

Por lo tanto, se define el análisis espacial como el conjunto de técnicas que toman en cuenta la componente espacial del dato geográfico. Estas técnicas son muy variadas y para el trabajo de diploma han sido seleccionadas las clasificaciones establecidas por Víctor Olaya. Centrando la atención en la Estadística Descriptiva debido a que esta responde con las necesidades del módulo de análisis espacial a desarrollar, por el trabajo que realiza este tipo de análisis con la componente temática de la información geográfica.

1.6. Estadística Descriptiva.

La estadística descriptiva proporciona herramientas para organizar, simplificar y resumir información básica a partir de un conjunto de datos que de otra forma sería poco manejable. Incluye la tabulación, representación y descripción de un conjunto de datos, tanto de variables numéricas como de variables categóricas (Lugo, y otros, 2000).

La estadística descriptiva se encuentra constituida por un conjunto de técnicas que tienen por objeto organizar y presentar de manera conveniente para su análisis, la información contenida en una muestra. Dicha muestra se encuentra representada por variables visuales que permiten representar proporciones o cantidades de forma visual. Estas técnicas no son excluyentes, sino complementarias, sin embargo dependiendo del tipo de datos que se desee manejar, no siempre es posible utilizarlas todas (Zylberberg, 2005).

Por lo tanto, se define la estadística descriptiva como el resumen de los datos o en terminología más actual, el análisis estadístico de los datos. Donde se tiene una representación de variables que permiten representar proporciones o cantidades.

1.6.1. Técnicas asociados a la estadística descriptiva.

Varios son los términos que se manejan cuando se hace referencia a la Estadística descriptiva entre ellos se pueden mencionar **la población** que consiste en la totalidad de los elementos bajo consideración, **la muestra** que es parte de la población que se selecciona para su análisis y luego de esto **las técnicas** que se pueden aplicar.

Entre las técnicas que se pueden aplicar dentro de la estadística descriptiva, existen tres clasificaciones según (Zylberberg, 2005): distribución de frecuencias, parámetros numéricos y descripción gráfica.

- ✚ **Distribución de frecuencias:** Es una clasificación de los datos en clases o categorías de acuerdo a sus valores. De aquí salen las tablas de frecuencia, pero antes de su elaboración es indispensable contar, primeramente, con el criterio de clasificación a utilizar, que se define a través de los límites de clases o bien mediante fronteras de clase.
- ✚ **Parámetros numéricos:** Constituyen aquellos valores que sean representativos del comportamiento de los datos y que dependan únicamente de dichos datos, lo que significa que se utilizan para describir el comportamiento de la muestra con un poco más de precisión.
- ✚ **Descripción gráfica:** Utilizada para representar claramente un conjunto de datos, ya que permiten la visualización rápida de los mismos. Según investigaciones para comprender el comportamiento de un conjunto de datos es mucho más sencillo a través de gráficas, que sí se muestra únicamente mediante valores numéricos. Ejemplo de estas gráficas: de barra, de pastel, histogramas de frecuencia, polígonos de frecuencia, entre otros.

1.6.2. *Parámetros numéricos fundamentales de la estadística descriptiva.*

Los parámetros numéricos en cuanto al tipo de información que dan según (Zylberberg, 2005), se clasifican en: medidas de tendencia central, medidas de dispersión y medidas de forma.

Las medidas de tendencia central son los valores que se encuentren dentro del rango de la muestra y que se consideren representativos de la misma. Entre las más usuales se encuentran la media aritmética, la mediana y la moda.

Media aritmética: Es el valor medio ponderado de la serie de datos. Se calcula multiplicando cada valor por el número de veces que se repite, para luego, la suma de estos productos dividirlo por el total de datos de la muestra.

Se define, dada la muestra $\mathbf{M} = \{x_1, x_2, \dots, x_n\}$, la medida típica más comúnmente utilizada es definida por:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

Donde \bar{x} : Media aritmética; n : Número de datos en la muestra; x_i : Valores de la muestra.

Mediana: Es el valor de la serie de datos que se sitúa justamente en el centro de la muestra una vez ordenada.

Si la cantidad de elementos en la serie de datos es impar, se define por:

$$M = y_{\frac{n+1}{2}}$$

Si la cantidad de elementos en la serie de datos es par, se define por:

$$M = \left[\frac{y_{\frac{n}{2}} + y_{\frac{n+2}{2}}}{2} \right]$$

Donde **M**: Mediana; **n**: Cantidad de elementos de la serie de datos; **y**: Posición del elemento en la serie de datos.

Moda: Es el valor de la variable que se presenta con mayor frecuencia en la muestra, es decir, el más se repite.

En la presente investigación, la estadística descriptiva persigue la síntesis de la información en los parámetros fundamentales, debido a que se persiguen valores que determinen el comportamiento global, así como sintetizar los datos en un valor representativo. Es por ello que se hace referencia a las técnicas o métodos del tipo: descripción gráfica y parámetros numéricos, apoyándose en la utilización de las medidas de tendencia central.

Por lo que, si se realiza una evaluación general, sería una investigación muy extensa, gracias a estas herramientas estadísticas se pueden hacer análisis sencillos, logrando rapidez, eficacia y una evaluación exacta de lo que se persigue (Muñoz, 2004).

La media es reconocida como la más apropiada y útil, debido a que:

- ✚ Considera todos los valores de la distribución.
- ✚ Los puntajes contribuyen de manera proporcional al hacer el cómputo de la media.
- ✚ Las medias de dos o más distribuciones pueden ser fácilmente promediadas.
- ✚ Se utiliza en procesos y técnicas estadísticas de gran complejidad (Zylberberg, 2005).

Sin embargo, cuando en una distribución se presentan casos cuyos puntajes son muy bajos o muy altos respecto al resto del grupo, es recomendable utilizar la mediana o la moda, porque dada las características de la media, esta es afectada por los valores extremos.

Las representaciones gráficas constituyen unos de los métodos más sencillos de exponer la información. Debido a la capacidad de impactar al lector, con muy poco esfuerzo de su parte, dando una información rápida y global de los datos, siendo útiles incluso al investigador, pues le permite tener una idea general de los resultados. En la estadística descriptiva, tanto para valores cualitativos como cuantitativos, las gráficas

que predominan son las de barra y las de pastel. Se utilizan estos parámetros tradicionales y las gráficas, con propósitos descriptivos, para organizar y resumir los datos numéricos en la solución de manera rápida y precisa (JUÁREZ, 2003).

1.7. Soluciones existentes.

Para lograr una mejor comprensión del desarrollo y el buen funcionamiento del sistema, se hace necesario realizar un análisis de algunas de las soluciones que poseen características semejantes a las planteadas en la investigación.

Sextante.

Sextante es una librería de código abierto, que contiene algoritmos de análisis espacial, desarrollada por Víctor Olaya. Se encuentra implementada en lenguaje Java y puede ser utilizada por los programadores para incorporar funcionalidades de análisis en su software.

La ejecución de un algoritmo SEXTANTE puede efectuarse desde el cuadro de herramientas o interfaz de línea de comandos. La interfaz gráfica de SEXTANTE y el propio proceso de trabajo es de forma independiente, por lo tanto, los usuarios no necesitan saber nada sobre el mismo algoritmo, pero tienen que entender los parámetros que desea definir para los datos y los resultados que se persiguen (Olaya, 2009).

La ejecución de los procesos se lleva a cabo mediante los siguientes pasos:

1. Seleccionar una capa de entrada (o más) como raster, vector, o una tabla.
2. Definir el algoritmo deseado a través de línea de comandos o cuadro de herramientas.
3. Especificar una capa de salida (o varias) en el formato deseado.

Dependiendo del tipo de geometría cargada, los distintos bloques funcionales del cuadro de herramientas se activan. El procesamiento de los formatos en los que dependes de los sistemas SIG SEXTANTE está siendo utilizada con: gvSIG: shp, dxf, tif, o asc OpenJUMP: shp, tif, uDig: shp, tif. Existen herramientas en general, y dependiendo del tipo de geometría, herramientas vectoriales, como la conversión de un tipo de geometría a otra (polígono -> líneas) o cálculo de las coordenadas del punto (Olaya, 2009).

Servicio de Mapas Temáticos (WMT) de la UCI.

Este servicio permite la creación de mapas temáticos a partir de datos estadísticos proporcionados por los usuarios y de la base geográfica escogida por ellos. Es de código abierto y además se encuentra disponible estando en línea o desconectado de la red (Torres, 2010).

Según (Torres, 2010) el servicio permite además la obtención de los datos estadísticos a partir de un libro de cálculo o de una base de datos generando ocho tipos de mapas temáticos:

- ✚ Mapa de datos puntuales: Identifican el fenómeno y lo sitúan según sus coordenadas. Los fenómenos implicados tienen una situación espacial única (x, y) y un atributo que es representado en el mapa mediante símbolos que no deben implicar ningún tipo de jerarquía.
- ✚ Mapa de datos lineales: Este tipo de mapas cualitativos se utiliza para representar fenómenos que tienen una forma lineal definida, como carreteras, ríos, fronteras, rutas de viaje, etcétera.
- ✚ Mapa de datos superficiales: Los mapas de datos superficiales informan sobre la distribución de fenómenos que ocupan extensiones superficiales. Ejemplos clásicos son los mapas de suelos, geológicos, forestales, etcétera; que ofrecen información cualitativa o descriptiva acerca de estos temas o variables.
- ✚ Mapa de puntos: Se basa en la representación, mediante la repetición de un símbolo puntual uniforme del fenómeno en cuestión atendiendo a la cantidad. Estos mapas son fácilmente comprensibles y muestran de un modo efectivo la distribución de un evento, informando no sólo que existe sino también dónde se concentra. Son complicados de elaborar dada la información estadística precisa y localizada que requieren.
- ✚ Mapa de coropletas: Aportan información cuantitativa y refleja la distribución espacial de un fenómeno mediante tramas o gamas de color, se utilizan para representar fenómenos discretos asociados a unidades de enumeración, generalmente superficies administrativas (países, provincias, municipios, etcétera). Un ejemplo clásico son los mapas de densidad de población. Es el método adecuado para reflejar, por ejemplo, el número de extranjeros que existe por cada mil habitantes, pero no para representar las medias anuales de temperatura de un país.
- ✚ Mapa de símbolos proporcionales: Esta técnica sirve para representar la distribución de datos cuantitativos de clase demográfica y económica mediante símbolos o figuras de diferentes tamaños y es la más utilizada en cartografía temática cuantitativa. Se selecciona un símbolo fijo (círculo, cuadrado, triángulo) y se varía su tamaño en proporción a la cantidad que se representa. Cada símbolo proporcional tiene dos funciones: localizar el dato en un lugar y dar la información de cantidad relativa a ese lugar mediante su tamaño. Son fáciles de interpretar porque asocian las cantidades de un dato (o variable) a los tamaños de forma muy intuitiva.
- ✚ Mapa de flujos: Este tipo de mapas sirve para mostrar el dinamismo de ciertos fenómenos y pueden ser cuantitativos o cualitativos. El movimiento se simboliza mediante una línea o flechas de ancho variable proporcionales a su importancia y esquematizadas de acuerdo con el trazado, que une los

lugares -de origen y de destino- del movimiento. Mediante estas líneas -o flechas cuando se quiere señalar el origen y destino y señalar la dirección y el sentido del flujo- se representa qué tipo de movimiento es el que se da y qué cantidad de movimiento se está dando en el caso de los mapas cuantitativos. Algunos ejemplos son mapas de corrientes marinas, red de transporte, migraciones, rutas migratorias etcétera.

- ✚ Cartograma: El cometido principal del cartograma es ilustrar una distribución temática de forma impactante. Por eso los datos deben adecuarse a este tipo de representación o no merecerá la pena realizarlo. Antes de proceder a la confección del mapa, hay que comparar los datos a cartografiar con la base geográfica. Si a unidades grandes les corresponden datos pequeños y viceversa, entonces resultará adecuado un cartograma como representación de dicha distribución.

Este servicio es multiplataforma, se puede tener instalado en el servidor cualquier sistema operativo. Se encuentra basado en los lenguajes de programación Python y JavaScript, utilizando como gestor de base de datos PostgreSQL con la extensión PostGIS para el manejo de datos geográficos (Torres, 2010).

El servicio cuenta con tres operaciones:

1. GetCapabilities, que retorna los metadatos del servicio.
2. GetThematicMap, que genera un mapa temático y la leyenda asociada a partir de los datos y parámetros definidos por el cliente.
3. LoadMap, que permite cargar un mapa temático generado con anterioridad y su leyenda.

Para acceder a las operaciones del servicio –esté local o remoto- se pueden invocar, mediante una URL que contenga los parámetros requeridos por cada operación. Después de recibir una petición el servicio de mapas temáticos debe satisfacer la petición retornando el documento HTML que contenga el mapa temático y la leyenda correspondiente o retornar una excepción (Torres, 2010).

GeneSIG.

La Plataforma GeneSIG es una plataforma soberana desarrollada en la UCI para la creación de SIG, la misma tiene como objetivos fundamentales según (Zaldívar, 2010):

- ✚ Permitir la representación geoespacial de la información asociada a cualquier negocio que lo requiera.
- ✚ Proporcionar servicios de acceso a la información geográfica, para su consulta, análisis y visualización, mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnología SIG.

- ✚ Integrar la información socioeconómica existente (recursos humanos, activos fijos, entidades de servicios, lugares de interés) con la información geográfica asociada.

Es un producto encaminado a realizar la representación y análisis geoespacial de información geográfica y su estructura arquitectónica permite personalizar sus funcionalidades a cualquier negocio que lo requiera a través de la reutilización de sus componentes. Puede ser considerado como un Sistema de Información Geográfica Web único y extensible, basado en estándares OpenGIS² que incluye funcionalidades operativas de las aplicaciones de esta tecnología (Zaldívar, 2010).

Otro aspecto a destacar es que está implementada sobre herramientas y tecnologías libres, lo que facilita su uso y manejo de acuerdo a las necesidades específicas de sus usuarios. Además de contar con una interfaz bastante sencilla y de fácil manejo, permitiendo la personalización y uso de todos los componentes que la integran.

Posibilita modificar el mapa creando tematizaciones en cuanto a:

- ✚ Estilo Símbolo Proporcional: Tematiza según la selección de una capa, un criterio de análisis, un color y un símbolo.
- ✚ Estilo Graduado: Basado en la definición de la capa, el criterio de análisis y la cantidad de clases (rangos), estableciendo un color para ellos.
- ✚ Estilo Categorizado: Posibilita constituir los estilos de las capas estableciendo colores diferentes a los distintos valores escogidos por el usuario.
- ✚ Gráficas Dinámicas: Permite visualizar en el mapa datos establecidos por el usuario a través de gráficas de pie y de barras.

Luego del estudio realizado a los sistemas homólogos, se le realiza un análisis al SIG UCI, en cuanto a sus características, con el objetivo de determinar el posible uso de estos sistemas en este SIG:

Características del SIG UCI.

El SIG UCI fue desarrollado en su primera versión con la versión 1.0 de la plataforma GeneSIG; diseñado con el propósito de representar información socioeconómica de la Universidad. Luego de desplegada la solución hubo quejas por parte de la comunidad universitaria, pues el sistema era muy lento. Esta afectación en el rendimiento era dada por el uso de una plataforma tan potente, diseñada para sistemas más complejos, basada en cargar una sola imagen del mapa cada vez que se realizaba una interacción sobre este.

² OpenGIS: Define estándares abiertos interoperables entre World Wide Web.

Sin embargo existían otras herramientas que podían ser empleadas para representar esta información y que aportaban mayor agilidad en la interacción con el mapa. Por ello la LPS decide desarrollar una nueva versión del SIG UCI basada en estas tecnologías.

De las tecnologías escogidas fue seleccionado el framework Symfony para el desarrollo del SIG; que basa su funcionamiento interno en la arquitectura Modelo Vista Controlador. Estructurando sus componentes en bundles, tales como el bundle para el Control de Capas y el bundle de Navegación. Está basado en los lenguajes php, del lado del servidor, y JavaScript, del lado del cliente.

Emplea Doctrine2, como ORM, el cual convierte las tablas de la base de datos en entidades, tratándolas como objetos e independizando la aplicación de la base de datos. Como gestor de base de datos utiliza PostgreSQL, con la extensión PostGIS para el soporte de datos espaciales.

Como Servidor de Mapas, es empleado MapServer6, utilizando como archivo de configuración del mapa el mapfile³. Para la interacción con el mapa, en el cliente, es usada la librería JavaScript Open Layers. Como librerías para la interacción con el HTML y para dotar al SIG de un estilo amigable se utilizan JQuery y Bootstrap.

✚ A continuación del previo análisis se concluye que:

Sextante brinda posibilidades de integración con otros sistemas, pero solamente de escritorio, ejemplo de ello son GvSIG y uDig. Estos SIG posibilitan mezclar las funcionalidades de la biblioteca mediante la utilización de java, no obstante, no se puede utilizar debido a que actualmente Sextante no soporta la interoperabilidad con sistemas web (OSGeo, 2010).

El Servicio de Mapas Temáticos es una herramienta interesante, permite que el usuario cree un mapa temático estableciendo él mismo su fuente de datos y el mapa base. A pesar de que esta herramienta brinda interoperabilidad no basa su funcionamiento en el uso del mapfile, con lo cual las capas temáticas no serían guardadas en este archivo y el posterior acceso a ellas no sería posible. Este servicio retorna un documento html con el mapa temático, el cual no puede ser accedido mediante una librería como OpenLayers debido a que no cumple con los estándares OGC para los servicios de mapas (WMS).

³ Mapfile: Archivo donde se encuentra la configuración del mapa del servidor de mapas MapServer.

Los tipos de mapas que este servicio genera son muy variados, no obstante no tiene objetivo representar algunos de ellos, ejemplo de esto son los mapas lineales y de datos superficiales pues las capas que se manejan no representan ni carreteras, ni suelos.

Concluyendo, que para el desarrollo de la solución, las autoras se basan en las funcionalidades que brinda la plataforma GeneSIG, no haciendo uso de ella debido a que sus arquitecturas son diferentes, pero guiando la solución en la representación temática de los atributos cuantificados que esta genera, tales como las tematizaciones para la representación de gráficas de barra, gráficas de pastel, estilo graduado, estilo proporcional y estilo categorizado.

1.8. Metodología y herramientas a utilizar.

1.7.1. Metodología de desarrollo.

El proceso de creación de un software no es una tarea fácil, por lo que existen diferentes metodologías para el desarrollo de los mismos. Para el logro de un producto final de calidad, con el cual estén satisfechos tanto los clientes como el equipo de desarrollo, es necesario la utilización de una metodología, tomando esta como el conjunto de procedimientos, técnicas y el soporte documental que ayuda a los desarrolladores de software a obtener un nuevo producto. Las metodologías detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. Son las encargadas de definir Quién hace Qué, Cuándo y Cómo y han sido desarrolladas diferenciándose por su fortaleza y debilidad.

Según la gran variedad de metodologías que existen, se clasifican en dos grandes grupos: las metodologías ágiles y las pesadas o tradicionales. Entre las principales metodologías ágiles se tiene a Programación Extrema (XP, por sus siglas en inglés), Scrum⁴, Iconix⁵, Crystal⁶, Proceso Unificado Ágil (AUP, por sus siglas en inglés). Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Entre las metodologías tradicionales se tienen al Proceso Unificado Racional (RUP, por sus siglas en inglés), Marco de Soluciones de Microsoft (MSF, por sus siglas en inglés). Estas metodologías se focalizan en documentación, planificación y procesos. Están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, presentan cierta resistencia a los

⁴ Scrum: Marco de trabajo para la gestión y desarrollo de software basado en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

⁵ Iconix: Metodología pesada-ligera de desarrollo del Software que se halla a medio camino entre un RUP (Rational Unified Process) y un XP (eXtreme Programming).

⁶ Crystal: Compilación de un conjunto de metodologías que facilitan el desarrollo de software.

cambios, el cliente interactúa con el equipo de desarrollo mediante reuniones, más artefactos, más roles, por lo que es un proceso más controlado, con numerosas normas o políticas (G. Figueroa, y otros, 2011). Luego de un estudio realizado a las distintas metodologías se propone como metodología de desarrollo a utilizar Proceso Unificado de Desarrollo Ágil.

Versión simplificada de Proceso Unificado de Desarrollo (RUP), fácil de entender y aplicable al desarrollo de software comercial, utilizando las tendencias y conceptos que aún permanecen fieles a RUP.

A diferencia de este, que cuenta con 9 **disciplinas**, AUP presenta sólo 7, de las cuales algunas son la unión de otras en RUP (**Ver Figura 1.2**).

- ✚ Modelado, Implementación, Prueba, Despliegue, Administración de la configuración, Administración o gerencia del proyecto, Entorno.

Divide su ciclo de vida en 4 **fases** (**Ver Figura 1.2**):

- ✚ Iniciación, Elaboración, Construcción, Transición.

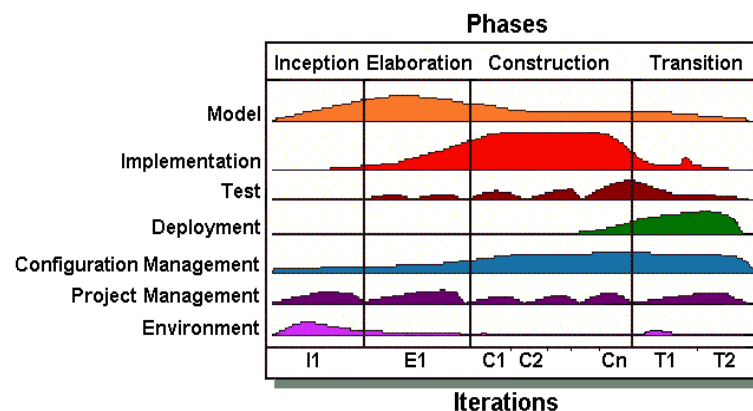


Fig. # 1.1: Esquema que muestra los flujos de trabajo y las fases de AUP.

AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas, Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad. Surge por una necesidad de acelerar el proceso de desarrollo de software para proyectos que sean pequeños. RUP Ágil es flexible, está orientada a equipos pequeños y presenta una significativa simplificación pero a pesar de ello no renuncia a las buenas prácticas ingenieriles para asegurar la calidad del producto. Propone los mismos roles, artefactos pero en una versión simplificada; es decir en RUP Ágil sólo se utilizan los artefactos que son imprescindibles y realmente necesarios para la realización del producto (G. Figueroa, y otros, 2011).

Esta versión ágil de la metodología RUP se basa en los siguientes principios:

- ✚ Simplicidad: Todo se describe concisamente utilizando poca documentación.

- ✚ Agilidad: El ajuste a los valores y principios de la Alianza Ágil.
- ✚ Centrarse en actividades de alto valor: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
- ✚ Adaptación de este producto para satisfacer sus propias necesidades: Es de fácil acomodo a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptarla a AUP.
- ✚ Herramienta de la independencia: Usted puede usar cualquier conjunto de herramientas que desea con AUP. Se sugiere utilizar las herramientas más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto (G. Figueroa, y otros, 2011).

En la presente investigación se opta por la elección de esta metodología ya que es la utilizada por los especialistas en el proyecto. La misma se encuentra diseñada para el trabajo de grupos pequeños de desarrollo y es flexible. Concuera con la solución de esta investigación porque está diseñada para proyectos donde la cantidad de casos de uso es pequeña. Aplicando esta metodología se generan roles y artefactos pero más simplificados.

1.7.2. Lenguaje Unificado de Modelado (UML) v2.0.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas (Giraldo , y otros, 2005).

Está pensado para usarse con todos los procesos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar (Fowler, 1999).

Las principales características de UML son:

- ✚ Lenguaje unificado para la modelación de sistemas.
- ✚ Tecnología orientada a objetos.
- ✚ Corrección de errores viables en todas las etapas.
- ✚ Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Se seleccionó UML como Lenguaje de Modelado, ya que soporta la metodología de desarrollo AUP, está consolidado como lenguaje estándar de análisis y diseño y es fácil de utilizar. Dentro de sus ventajas figura

que la trazabilidad y la documentación del proyecto se realiza de una forma ordenada y guiada por casos de uso.

1.7.3. Herramienta de modelado (Visual Paradigm) v8.0.

Visual Paradigm for UML es una poderosa herramienta CASE que utiliza UML para el modelado; es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (Visual Paradigm, 2006).

Visual Paradigm ofrece:

- ✚ Un entorno de creación de diagramas para UML 2.0.
- ✚ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✚ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✚ Capacidades de ingeniería directa e inversa.
- ✚ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✚ Disponibilidad de múltiples versiones, para cada necesidad.
- ✚ Disponibilidad de integrarse en los principales IDEs.
- ✚ Disponibilidad en múltiples plataformas.

Por todas las ventajas anteriormente expuestas se seleccionó Visual Paradigm como herramienta de modelado teniendo en cuenta además, que el equipo de desarrollo cuenta con experiencia suficiente en el trabajo con la misma.

1.7.4. Servidor Web (Apache) v2.2.

Apache es un servidor HTTP⁷ libre, de código abierto y muy usado a nivel mundial. Es el encargado de contestar las peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo en correspondencia con los comandos solicitados (textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música) (Cibernetia, 2010).

Cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

⁷ HTTP: Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la Web.

- ✚ Espera peticiones en el puerto TCP⁸ indicado (el estándar por defecto para HTTP es el 80).
- ✚ Recibe una petición.
- ✚ Busca el recurso.
- ✚ Envía el recurso utilizando la misma conexión por la que recibió petición.
- ✚ Vuelve al segundo punto.

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan (Cibernetia, 2010):

- ✚ Multiplataforma.
- ✚ Es un servidor de web conforme al protocolo HTTP/1.1.
- ✚ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API⁹ de programación de módulos, para el desarrollo de módulos específicos.

1.7.5. Marco de trabajo (Symfony) v2.3.5.

Symfony2 es un framework completo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Symfony2 ha sido ideado para exprimir al límite todas las nuevas características de PHP¹⁰ 5.3 y por eso es uno de los frameworks PHP con mejor rendimiento. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la

⁸ TCP: Protocolo de Control de Transmisión, es uno de los protocolos fundamentales en Internet.

⁹ API: Interfaz de programación de aplicaciones, es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas comúnmente "librerías").

¹⁰ PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

mayoría de gestores de bases de datos, como MySQL¹¹, PostgreSQL, Oracle¹² y Microsoft SQL Server¹³ (Fabien, 2007).

Características de Symfony2:

- ✚ Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows, Unix, Linux, entre otros.)
- ✚ Independiente del sistema gestor de bases de datos.
- ✚ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✚ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony2 está basado en un patrón clásico del diseño web conocido como arquitectura Modelo- Vista- Controlador (MVC) (Fabien, 2007).

El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La vista transforma el modelo en una página web que permite al usuario interactuar con ella.

El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Symfony2 tiene muchas ventajas, dentro de ellas cabe destacar:

- ✚ Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.
- ✚ Proporciona varias herramientas y clases, las cuales están encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.
- ✚ Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

En el desarrollo del subsistema se trabajará con la versión 2.3.5 de Symfony2, que cuenta con las características planteadas anteriormente además de la rapidez y flexibilidad. También cuenta con la nueva estructura de directorios de los proyectos llamada bundles que presta una mejor organización del código y

¹¹ MySQL: Sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

¹² Oracle: es un sistema de gestión de base de datos objeto-relacional, se considera uno de los sistemas de bases de datos más completos.

¹³ Microsoft SQL Server: Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

los archivos de configuración que se pueden escribir en PHP, XML¹⁴ o YAML¹⁵ (Ain't Markup Language, por sus siglas en inglés), a gusto del programador.

1.7.6. Biblioteca (Jquery) v2.0.

Biblioteca de JavaScript de código abierto, que permite simplificar la manera de interactuar con los documentos HTML¹⁶, manipular el árbol DOM¹⁷, manejar eventos, desarrollar animaciones y agregar interacciones con la técnica AJAX¹⁸ a páginas web. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Alvarez, 2009).

Algunas de sus características son:

- ✚ Manipulación de la hoja de estilos CSS.
- ✚ Efectos y animaciones.
- ✚ Animaciones personalizadas.
- ✚ JavaScript asíncrono y XML (AJAX).
- ✚ Soporta extensiones.
- ✚ Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- ✚ JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.

1.7.7. Librería (OpenLayers) v2.12.

Es una librería de código abierto que permite incorporar mapas dinámicos en las páginas web. Los mapas se pueden dotar de diversos controles con capacidades de zoom, panning, medida de distancias y muchas otras herramientas, es decir se utiliza para acceder, manipular y mostrar mapas en páginas web. Fue

¹⁴ XML: Es un metalenguaje extensible de etiquetas desarrollado por el sistema de distribución de información (world Wide Web).

¹⁵ YAML: Es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl.

¹⁶ HTML: Sigla de la expresión inglesa Hypertext Markup Language, “lenguaje marcado de hipertexto”, que sirve para escribir las páginas en Internet, dar formato a los textos y crear hiperenlaces.

¹⁷ DOM: DOM o Document Object Model es un conjunto de utilidades específicamente diseñadas para manipular documentos XML.

¹⁸ AJAX: Es un acrónimo de Asynchronous JavaScript + XML, “JavaScript asíncrono + XML”, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

desarrollado por MetaCarta como respuesta a GoogleMaps y no requiere descargas extras del lado del servidor. Proporciona herramientas para acceder a todo tipo de información geográfica proveniente de muy variadas fuentes, por ejemplo Web Map Services, Web Feature Services, Mapas comerciales, información genérica vectorial, entre otros (Higuera, 2010).

Con OpenLayers se puede:

- ✚ Crear mapas interactivos.
- ✚ Visualizar información espacial/geográfica detallada.
- ✚ Editar información espacial/geográfica.
- ✚ Interactuar con servicios SIG externos como GoogleMaps, MapServer, OpenStreamMap, entre otros.

1.7.8. Servidor de Mapa (MapServer) v6.0.1.

Entorno de desarrollo en código abierto para la creación de aplicaciones SIG en Internet/Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer (Mapserver, 2011).

Sus características principales son:

- ✚ Se ejecuta bajo plataformas Linux/Apache y Windows (MS4W).
- ✚ Formatos vectoriales soportados: ESRI shapefiles, PostGIS, ESRI ArcSDE, GML y otros muchos vía OGR.
- ✚ Formatos raster soportados: JPG, PNG, GIF, TIFF/Geo TIFF, EPPL7 y otros vía GDAL¹⁹.
- ✚ Fuentes TrueType.
- ✚ MapScript proporciona una API para poder acceder a las funcionalidades de MapServer mediante lenguajes de programación como PHP, Java, Perl, Python, Ruby o C#.
- ✚ Soporte de estándares interoperables y conformes con Open Geospatial Consortium, como WMS, SLD, WFS, WCS y SOS.

1.7.9 Sistema Gestor de Base de Datos (PostgreSQL) v9.1.

El PostgreSQL es un Sistema Gestor de Base de Datos (SGBD) que posee características significativas, entre las que se pueden incluir subconsultas, se puede usar, modificar y distribuir de forma gratuita debido

¹⁹ GDAL: Es una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales, publicada bajo la Licencia MIT por la fundación geoespacial de código abierto.

a la liberación de su licencia. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (QUIÑONES, 2007).

Ventajas:

- ✚ PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- ✚ Puede estar en funcionamiento varios años bajo una alta actividad y no presentar fallos en el servicio.

1.7.10 Extensión (PostGIS) v1.5.3.

PostGIS es una extensión, de software libre, que añade soporte de objetos geográficos a la base de datos objeto-relacional de PostgreSQL. Esto la convierte en una base de datos espacial para su utilización en Sistemas de Información Geográfica (opengeo.org, 2014).

Entre sus principales características se destacan:

- ✚ Alto rendimiento.
- ✚ Simples características para SQL.
- ✚ Ofrece representaciones espaciales de los tipos de geometría (puntos, líneas, polígonos).
- ✚ Soporte para operaciones espaciales comunes y avanzadas como la creación de la geometría y la conversión, reproyección, buffer, convexa, la generalización, la unión, entre otras.
- ✚ Geodésica de apoyo para las mediciones en todo el mundo.
- ✚ Línea de comandos y herramientas gráficas para la gestión flexible.
- ✚ PostGIS está ampliamente reconocida como una base de datos espacial de back-end de software cliente y servidor, incluyendo:

Cliente:

- Privativo: ArcGIS, Manifold, Caja fuerte FME, Cadcorp SIS, MapInfo Professional.
- Libre: GRASS, QGIS, uDig, gvSIG.

Servidor:

- Privativo: ArcServer, Enterprise iónicos, MapDotNet servidor.
- Libre: GeoServer, Mapserver, Mapnik, Deegree, SharpMap.

1.7.11 Entorno de Desarrollo Integrado (NetBeans) v8.0.

NetBeans es un entorno de desarrollo integrado, creado principalmente para el lenguaje de programación Java, aunque puede servir para otros lenguajes de programación. Esta herramienta ayuda a los

programadores escribir, compilar, depurar y ejecutar programas. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, facilitándole al usuario comenzar a trabajar inmediatamente. Además ofrece servicios comunes permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. El mismo es un producto libre y gratuito sin restricciones de uso.

Su aprendizaje se ha convertido en fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma. Está disponible para diversos sistemas operativos como Solaris, Windows, MacOS y GNU Linux. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (NetBeans.org, 2010).

Para el desarrollo con el framework Symfony, este IDE trae por defecto un módulo, el cual posibilita crear bundles desde el mismo IDE. Permite además ejecutar tareas desde una interfaz gráfica, ofreciendo una serie de comandos, con su sintaxis, una ayuda, y una opción para establecer los parámetros este. Otra de las ventajas es el autocompletamiento de código de las plantillas twig, además de la diferenciación en colores de las sentencias (Puertas, 2013).

Dadas las características planteadas anteriormente y que permite el trabajo con el framework de desarrollo seleccionado (Symfony2) y los lenguajes de programación (PHP, JavaScript, SQL) es ideal para el desarrollo del módulo.

1.9. Conclusiones Parciales.

En el actual capítulo se realizó un análisis de los principales conceptos asociados al problema a resolver lo que permitió ver la importancia del análisis espacial en los SIG, así como sus principales potencialidades. Mediante la identificación de las soluciones existentes y la problemática, se evidenció la necesidad de desarrollar un módulo de análisis espacial estadístico descriptivo para el SIG UCI, aunque solo se pudieron utilizar de las soluciones existentes algunas de sus funcionalidades, ya que por sus características no permitieron su utilización e integración. Además se realizó la selección de las herramientas y la metodología a utilizar, siendo estas las que más se adecuan a las características del módulo que se desea desarrollar, garantizando un producto en el tiempo requerido y ajustándose a las necesidades del cliente.

Capítulo 2: Presentación de la Solución Propuesta

2.1. Introducción.

Para el desarrollo de un software es necesario tener conocimiento del negocio y los requerimientos que el mismo debe cumplir. En este capítulo se describe la propuesta de solución para el problema existente utilizando la metodología AUP para la planificación, investigación y diseño de la herramienta a desarrollar. Para ello se describen los procesos del negocio que tienen que ver con el objeto de estudio, presentando el modelo de dominio y se realiza la especificación de requisitos funcionales y no funcionales que debe cumplir el módulo que se propone. Así como también una breve descripción de los actores del sistema, así como el modelo de casos de uso.

2.2. Modelo de dominio.

Es el artefacto más importante que se crea durante el análisis orientado a objetos, permite la representación visual de las clases conceptuales u objetos significativos de un dominio de interés. En la realización de este modelo se deben capturar las abstracciones e informaciones necesarias para entender el dominio en el contexto de los requisitos actuales, permitiendo a las personas comprender el dominio, sus conceptos, terminología y relaciones (Larman, 2003).

Para la realización de esta investigación se decidió realizar el modelo del dominio debido a que se cuenta con pocos expertos en el tema y gran parte de la información utilizada ha sido obtenida mediante el estudio de sistemas similares. Este modelo es un diagrama formado por las clases conceptuales, las asociaciones entre las mismas y los atributos.

2.2.1. Diagrama de clases del Modelo de Dominio.

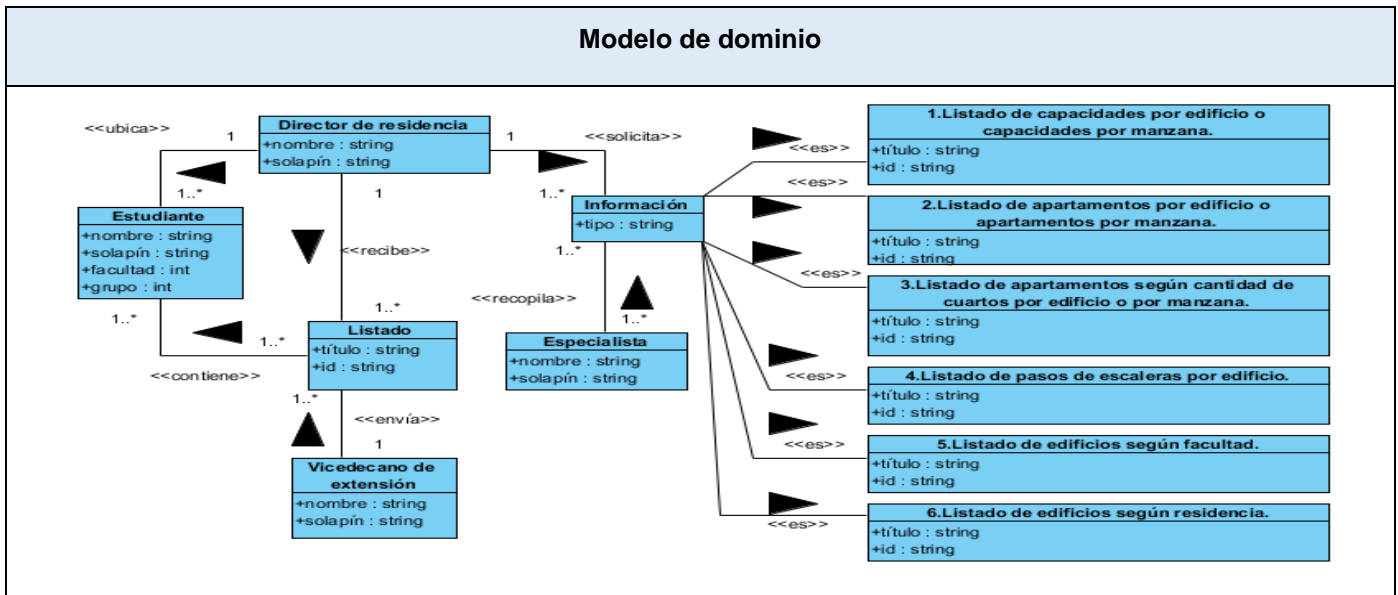


Fig. # 2.1: Diagrama de clases del dominio.

2.2.2. Definición de las clases del Modelo de Dominio.

- ✚ **Director de Residencia:** Persona que recibe del vicedecano de extensión, un listado de estudiantes a ubicar en la residencia. Para este proceso el director solicita información al especialista.
- ✚ **Estudiante:** Sujeto a ubicar en la residencia.
- ✚ **Vicedecano de extensión:** Persona de la facultad que se encuentra frente a los estudiantes en la residencia y entrega listados de los mismos al Director de la residencia para su posterior ubicación.
- ✚ **Especialista:** Encargado de recopilar información acerca de las capacidades en la residencia.
- ✚ **Información:** Conjunto de datos recopilados por el especialista en la residencia teniendo en cuenta capacidades de estudiantes por edificio o por manzana, los apartamentos disponibles y ocupados por edificio o por manzana, los apartamentos según cantidad de cuartos, los pasos de escalera por edificio, listado de apartamentos según cantidad de cuartos, listado de edificios según facultad y listado de edificios según residencia.
- ✚ **Listado:** Listado de los estudiantes a ubicar enviado por el Vicedecano de extensión al Director de la residencia.
- ✚ **Listado de capacidades por edificio o capacidades por manzana:** Listado que contiene la capacidad total de estudiantes y capacidad libre por edificios.

- ✚ **Listado de apartamentos por edificio o apartamentos por manzana:** Listado que contiene los apartamentos disponibles y ocupados por edificio.
- ✚ **Listado de apartamentos según cantidad de cuartos por edificio o por manzana:** Listado que contiene los apartamentos que tengan uno, dos, tres, y cuatro cuartos por edificio.
- ✚ **Listado de pasos de escalera por edificio:** Listado que refleja los pasos de escalera por edificio.
- ✚ **Listado de edificios según facultad:** Listado que contiene los edificios, según la facultad.
- ✚ **Listado de edificios según residencia:** Listado que contiene los edificios, según la residencia.

2.3. Técnica de captura de requisitos.

Existen varias técnicas que posibilitan capturar los requisitos de software de forma eficiente y segura, como la entrevista, introspección, cuestionarios, listas de verificación, tormenta de ideas y análisis de la documentación. Para satisfacer las necesidades del cliente se realizó la extracción de los requisitos del software mediante la utilización de las técnicas: entrevista y tormenta de ideas.

La entrevista resulta una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural (Baños, 2012).

La tormenta de ideas permite “la realización de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de captura, cuando los requisitos son todavía muy difusos” (Baños, 2012).

2.4. Especificación de Requisitos.

Según la (IEEE, 2004) los requisitos son una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Los requisitos son independientes del diseño, deben enseñar ¿qué? debe hacer el sistema más que el ¿cómo? debe hacerse. Se expresan en un lenguaje natural, pero pueden tener una representación abstracta de alto nivel o una descripción detallada y formal de la función del sistema, por lo que no hay una única manera de expresarlos. Según Wiegiers existen dos categorías de requisitos: Funcionales y no funcionales (Wiegiers, 2003).

2.4.1. Requisitos funcionales.

Los requisitos funcionales describen con claridad lo que debe hacer el sistema para satisfacer las necesidades del cliente, definiendo las entradas, salidas y excepciones del mismo. Seguidamente se enumeran los que se han capturado para el desarrollo de esta investigación (Wieggers, 2003).

- **RF1. Realizar tematización según estilo proporcional.**

Esta funcionalidad permite que el usuario pueda crear un mapa temático mediante “Estilo Proporcional” y mostrar la capa tematizada en el mapa, en función de la capa a tematizar, del criterio de análisis definido y opcionalmente la opacidad y la transparencia con la que se visualizará dicha capa en el mapa.

- **RF2. Realizar tematización según estilo categorizado.**

Esta funcionalidad permite que el usuario pueda crear un mapa temático mediante “Estilo Categorizado” y mostrar la capa tematizada en el mapa, en función de la capa a tematizar y del criterio de análisis definido. Puede además especificar la información que se visualizará, a través de un filtro, en el cual se especifica el operador de comparación y el valor a comparar en cuanto a cantidad según criterio seleccionado. También opcionalmente se puede especificar la opacidad y la transparencia con la que se visualizará dicha capa en el mapa.

- **RF3. Realizar tematización según estilo graduado.**

Esta funcionalidad permite que el usuario pueda crear un mapa temático mediante “Estilo graduado” y mostrar la capa tematizada en el mapa, en función de la capa a tematizar, la cantidad de clases y el criterio de análisis definido. Además opcionalmente puede especificar la opacidad y la transparencia con la que se visualizará dicha capa en el mapa.

- **RF4. Realizar tematización según gráfica de barra.**

Esta funcionalidad permite que el usuario pueda crear un mapa temático mediante “Gráfica de barra” y mostrar estas tematizadas en el mapa, en función de la capa a tematizar de los criterios de análisis definidos. Además opcionalmente puede especificar la opacidad y la transparencia con la que se visualizará dicha capa en el mapa.

- **RF5. Realizar tematización según gráfica de pastel.**

Esta funcionalidad permite que el usuario pueda crear un mapa temático mediante “Gráfica de pastel” y mostrar estas tematizadas en el mapa, en función de la capa a tematizar de los criterios de análisis definidos. Además opcionalmente puede especificar la opacidad y la transparencia con la que se visualizará dicha capa en el mapa.

Capítulo 2: Presentación de la solución propuesta

El sistema deberá permitir que el usuario pueda crear un mapa temático por colores teniendo en cuenta los diferentes estilos, por gráficas de barras y de pastel, mostrando estas tematizaciones en el mapa en función del criterio de análisis que se haya seleccionado.

Estos criterios son los siguientes:

1. Capacidades por edificio o capacidades por manzana.
 - Capacidad libre.
 - Capacidad ocupada.
2. Apartamentos por edificio o apartamentos por manzana.
 - Apartamentos disponibles.
 - Apartamentos ocupados.
3. Apartamentos según cantidad de cuartos por edificio o por manzana.
 - Apartamentos con un cuarto.
 - Apartamentos con dos cuartos.
 - Apartamentos con tres cuartos.
 - Apartamentos con cuatro cuartos.
4. Pasos de escaleras por edificio.
5. Edificios según facultad.
6. Edificios según residencia.
7. Representar la media aritmética, la mediana o la moda de:
 - Capacidades por edificio o capacidades por manzana.
 - Apartamentos por edificio o apartamentos por manzana.
 - Edificios según facultad.
 - Edificios según residencia.

2.4.2. Requisitos No Funcionales (RNF).

Los requisitos no funcionales detallan las propiedades o cualidades que el sistema debe cumplir, los cuales se encuentran separados por categorías que se mencionan a continuación:

✓ RnF 1 Tipo de usuario final.

Debido a las acciones que se realizarán en la aplicación deberá ser manejada por usuarios con conocimientos básicos en informática.

✓ RnF 2 Finalidad.

La aplicación que se propone tendrá como objetivo fundamental la representación geoespacial estadística

Capítulo 2: Presentación de la solución propuesta

descriptiva de la información manejada por los directivos de la residencia estudiantil UCI.

✓ RnF 3 Ambiente.

Requisitos de Hardware y Software para Servidor Base de Datos y Servidor de Mapas

Hardware:

- Procesador: 3Ghz como mínimo.
- Memoria RAM: 2Gb como mínimo.
- Disco Duro: 40Gb.

Software:

- Sistema Operativo: GNU/Linux Ubuntu Server 12.04.
- Servidor Web Apache 2.2, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL 9.1 como Sistema Gestor de Base de Datos.
- PostGIS 1.5.3 como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 6.0.1, con extensión PHP MapScript.

Requisitos de Hardware y Software para estaciones clientes.

Hardware:

- Procesador: 512 MHz como mínimo.
- Memoria RAM: 128Mb como mínimo.
- Tarjeta de red.

Software:

- Sistema Operativo: GNU/Linux, Windows o Mac OS.
- Un navegador como Mozilla Firefox, u otro que cumpla con los estándares W3C (World Wide Web, por sus siglas en inglés), y cuente con soporte para CCS3 y HTML5.

✓ RnF 4 Eficiencia.

El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.

✓ RnF 5 Soporte

El proyecto deberá ofrecer la capacitación en el uso y mantenimiento del sistema al personal designado por la residencia estudiantil.

✓ RnF 6 Restricciones de diseño.

Los aspectos que a continuación se exponen garantizarán el correcto funcionamiento del software y su escalabilidad en cuanto al diseño de implementación.

- Para el modelado del sistema se utilizará como lenguaje de modelado UML.
- El sistema estará implementado en lenguaje php, en su versión 5.3. Se desarrolla a partir de Symfony2 en su versión 2.3.5 como framework de desarrollo.
- Se hace uso de Netbeans 8.0 como entorno de desarrollo integrado.
- Gestor de base de datos PostgreSQL 9.1.
- El sistema se construirá siguiendo la arquitectura que es propuesta por su plataforma base.

✓ RnF 7 Interfaces de usuario

Tener una apariencia profesional y un diseño gráfico sencillo.

2.5. Modelado del sistema.

El Modelo de Casos de Uso del Sistema permite que los desarrolladores de software y clientes lleguen a un acuerdo sobre las condiciones y posibilidades que debe cumplir el sistema. Proporciona la entrada fundamental para el análisis, el diseño y las pruebas (Rumbaugh, y otros, 2000).

2.5.1. Definición de los actores que interactúan con el sistema.

Un actor es un agente externo que interactúa con el sistema en pos de obtener un resultado esperado. Encontrar los actores es uno de los primeros pasos en la definición del uso del sistema. El sistema cuenta con el actor que se especifica a continuación:

Tabla # 2.1: Definición del actor del sistema.

Actor	Justificación
Director de Residencia	Es la persona con necesidad de acceder al sistema para consultar toda información referente a su trabajo en la residencia.

2.5.2. Definición de CU del sistema.

Tabla # 2.2: Definición de CU: Realizar tematización según tipo de estilo.

CU-1	Realizar tematización según tipo de estilo.
Actor	Director de residencia.

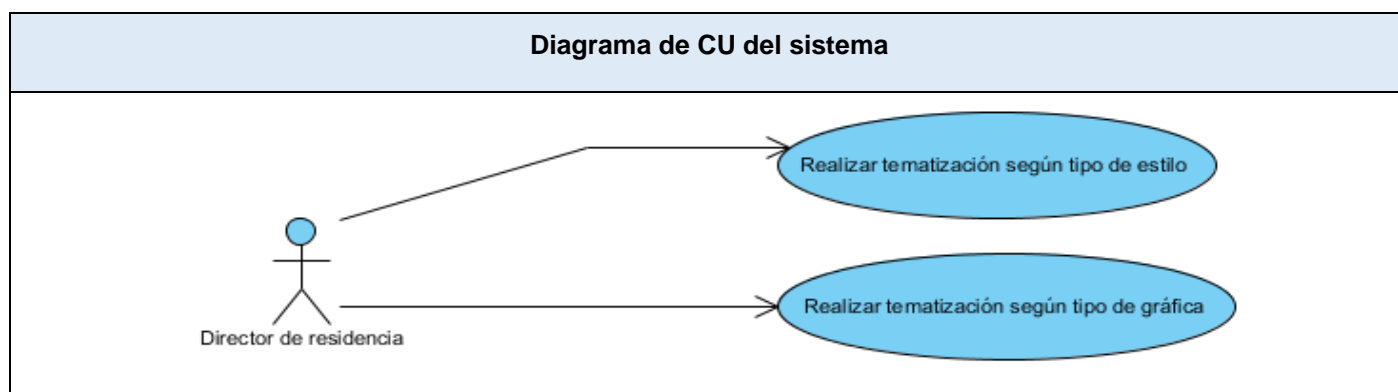
Capítulo 2: Presentación de la solución propuesta

Descripción	Permite al usuario tematizar según el tipo de estilo (proporcional, graduado, categorizado) seleccionado, teniendo en cuenta los campos a llenar en cada tipo de estilo.
Referencias	RF1,RF2,RF3

2.5.3. Diagrama de CU del sistema.

El diagrama de casos de uso del sistema brinda las funcionalidades que el sistema debe ofrecer para aportar un resultado de valor para sus actores.

Tabla # 2.3. Diagrama de CU Módulo de análisis espacial estadístico descriptivo para el SIG UCI.



2.5.4. Descripción textual del CUS: Realizar tematización según tipo de estilo.

Tabla # 2.4: Descripción textual del CU: Realizar tematización según tipo de estilo.

Nombre del Caso de uso del sistema:	Realizar tematización según tipo de estilo.
Actores del sistema:	Director de residencia.
Referencias:	RF1, RF2, RF3
Resumen:	El caso de uso comienza cuando el actor del sistema decide realizar alguna tematización teniendo en cuenta los tipos de estilos (proporcional, categorizado, graduado). Este finaliza cuando es mostrada en una capa del mapa la tematización según opciones seleccionadas.
Precondiciones:	–

Capítulo 2: Presentación de la solución propuesta

Poscondiciones:	El sistema realiza tematización según estilo seleccionado.
Prioridad:	Crítico.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. El caso de uso inicia cuando el usuario selecciona la opción Crear mapa temático. Ver interfaz 1</p> <p>Puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> a) Seleccionar Estilo Proporcional, ver sección 1: Estilo Proporcional. b) Seleccionar Estilo Categorizado, ver sección 2: Estilo Categorizado. c) Seleccionar Estilo Graduado, ver sección 3: Estilo Graduado. <p>Ver interfaz 2.</p>	<p>2. El sistema muestra la ventana Crear mapa temático.</p>
Prototipo de Interfaz	
Sección 1 “Estilo Proporcional”	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

Capítulo 2: Presentación de la solución propuesta

1. Elige la opción Estilo Proporcional.	2. Muestra interfaz con los campos a completar para realizar tematización según estilo proporcional con las opciones: <ul style="list-style-type: none"> • Capa (Manzana, Edificio). • Criterio. • Propiedades de la capa (Opcional): <ul style="list-style-type: none"> - Transparencia (Sí o No). - Opacidad (0 a 1).
3. Selecciona la capa.	4. Habilita la opción criterio.
5. Selecciona criterio de tematización.	6. Habilita el botón Tematizar Mapa.
7. Despliega las propiedades de la capa.	8. Muestra las opciones transparencia y opacidad.
9. Llena los campos de las propiedades de la capa, Transparencia y Opacidad y presiona el botón Tematizar Mapa.	10. Cierra la ventana Crear Tematización y visualiza en el mapa la capa tematizada según estilo proporcional con un color diferente.
Flujos alternos	
Acción del actor	Respuesta del sistema
3. Selecciona la opción Cancelar.	4. Cierra la ventana Crear Tematización y no realizar la tematización.
3. Selecciona la opción Cerrar.	4. Cierra la ventana Crear Tematización y no realizar la tematización.
5. Selecciona la opción ****Seleccione**** del campo criterio y vuelve al paso 5 .	6. Deshabilita el botón Tematizar Mapa y vuelve al paso 6 .
9. Deja campos opcionales Transparencia y Opacidad sin llenar.	10. Realiza tematización según estilo proporcional y regresa al paso 1 .
Sección 2 “Estilo Categorizado”	

Capítulo 2: Presentación de la solución propuesta

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Elige la opción Estilo Categorizado.	2. Muestra interfaz con los campos a completar para realizar tematización según estilo proporcional con las opciones: <ul style="list-style-type: none"> • Capa (Manzana, Edificio). • Criterio. • Parámetro estadístico (Opcional): <ul style="list-style-type: none"> - Moda. - Media aritmética. - Mediana. • Filtro (Opcional): <ul style="list-style-type: none"> - Operador (Operadores de comparación). - Valor. • Propiedades de la capa (Opcional): <ul style="list-style-type: none"> - Transparencia (Sí o No). - Opacidad (0 a 1).
3. Selecciona la capa.	4. Habilita la opción criterio.
5. Selecciona el criterio de tematización.	6. Carga datos en la tabla y habilita el botón Tematizar Mapa.
7. Despliega la opción parámetro estadístico.	8. Muestra las opciones moda, mediana, media y deshabilita la opción filtro.
9. Selecciona el parámetro estadístico.	10. Reinicia la opción criterio.
11. Regresa al paso 5 .	12. Vuelve a realizar el paso 6 .
13. Despliega la opción filtro.	14. Muestra las opciones operador y valor, además deshabilita la opción parámetro estadístico.
15. Llena campos del filtro Operador y Valor.	16. Reinicia la opción criterio.

Capítulo 2: Presentación de la solución propuesta

17. Regresa al paso 5 .	18. Vuelve a realizar el paso 6 .
19. Despliega las propiedades de la capa.	20. Muestra las opciones transparencia y opacidad.
21. Llena los campos de las propiedades de la capa, Transparencia y Opacidad y presiona el botón Tematizar Mapa.	22. Cierra la ventana Crear Tematización y visualiza en el mapa la capa tematizada según estilo categorizado con un color diferente.
Flujos alternos	
Acción del actor	Respuesta del sistema
3. Selecciona la opción Cerrar.	4. Cierra la ventana Crear Tematización y no realiza la tematización.
3. Selecciona la opción Cancelar.	4. Cierra la ventana Crear Tematización y no realiza la tematización.
5. Selecciona la opción ****Seleccione**** del campo criterio.	6. Deshabilita el botón Tematizar Mapa.
9. Selecciona la opción en blanco del campo parámetro estadístico.	10. Deshabilita el botón Tematizar Mapa.
15. Selecciona la opción en blanco del campo Operador.	16. Limpia y deshabilita el campo valor.
19. Deja los campos opcionales Filtro y Propiedades de la capa sin llenar.	20. Realiza tematización según estilo categorizado y regresa al paso 1 .
Sección 3 “Estilo Graduado”	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

Capítulo 2: Presentación de la solución propuesta

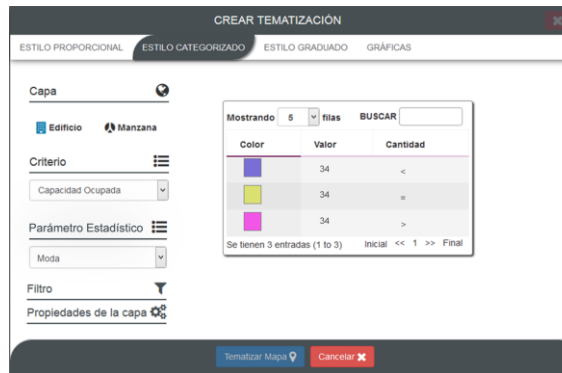
1. Elige la opción Estilo Graduado.	2. Muestra interfaz con los campos a completar para realizar tematización según estilo graduado con las opciones: <ul style="list-style-type: none"> • Capa (Manzana, Edificio). • Clases • Criterio. • Propiedades de la capa (Opcional): <ul style="list-style-type: none"> - Transparencia (Sí o No). - Opacidad (0 a 1).
3. Selecciona la capa.	4. Habilita la opción clases.
5. Introduce la cantidad de clases.	6. Habilita la opción criterio.
7. Selecciona criterio de tematización.	8. Carga datos en la tabla y habilita el botón Tematizar Mapa.
9. Despliega las propiedades de la capa.	10. Muestra las opciones Transparencia y Opacidad.
11. Llena campos de las Propiedades de la capa, Transparencia y Opacidad y presiona el botón Tematizar Mapa.	12. Cierra la ventana Crear Tematización y visualiza en el mapa la capa tematizada según estilo graduado con un color diferente.
Flujos alternos	
Acción del actor	Respuesta del sistema
3. Selecciona la opción Cerrar.	4. Cierra la ventana Crear Tematización y no realiza la tematización.
3. Selecciona la opción Cancelar.	4. Cierra la ventana Crear Tematización y no realiza la tematización.
7. Seleccionar la opción ****Seleccione**** del campo criterio y regresa al paso 7 .	8. Deshabilita el botón Tematizar Mapa y regresa al paso 8 .

Capítulo 2: Presentación de la solución propuesta

11. Deja campos opcionales Transparencia y Opacidad sin llenar.

12. Realiza tematización según estilo graduado y regresa al **paso 1**.

Ejemplo de prototipo de Interfaz de Estilo Categorizado



2.6. Conclusiones parciales

En este capítulo se mostraron los principales elementos que conforman la presentación de la solución propuesta. Evidenciando que la modelación del dominio, la especificación de requisitos funcionales y no funcionales y el modelado del sistema contribuyeron al entendimiento del equipo de desarrollo, lo que representa una ventaja para la correcta ejecución de las restantes etapas en la construcción de la solución propuesta.

Capítulo 3: Construcción de la solución propuesta.

3.1. Introducción.

En el siguiente capítulo se aborda todo lo relacionado con el diseño del módulo. Se hace una descripción de los distintos patrones que se usan en el proceso de implementación, ya sea tanto arquitectónico como de diseño. Se detalla una gran parte del proceso de implementación del módulo, mostrando los diagramas correspondientes al diseño. Se presenta el modelo de despliegue del módulo, el cual muestra cómo estarán distribuidos los recursos que interactúan con el sistema. Además, se especifican las técnicas a utilizar para realizar las pruebas al sistema.

3.2. Descripción de la arquitectura.

Durante el proceso de desarrollo de un software, el arquitecto de software es quien tiene una responsabilidad global sobre el proyecto, es quien diseña los aspectos más significativos del sistema para que funcionen en su conjunto. Sin embargo, la arquitectura de software está afectada no solo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, factibilidad de compresión, las restricciones y compromisos económicos y tecnológicos, y la estética (Rumbaugh, y otros, 2000).

Los patrones arquitectónicos definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, subsistemas) de la arquitectura (Pressman, 2005). Se hace necesario emplear patrones arquitectónicos para describir y diferenciar una arquitectura de tal manera que ayuden a definir una estructura para todos los componentes del sistema.

3.2.1. Patrón arquitectónico Cliente-Servidor.

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones. Desde el punto de vista funcional, la arquitectura permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente lo que en computación distribuida afecta directamente el tráfico de la red, reduciéndolo grandemente (Avendaño, y otros, 2013).

Capítulo 3: Construcción de la solución propuesta

El Cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end, normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

El cliente es el encargado de:

- ✚ Administrar la interfaz de usuario.
- ✚ Interactuar con el usuario.
- ✚ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✚ Generar requerimientos de bases de datos.
- ✚ Recibir resultados del servidor.
- ✚ Formatear resultados.

El servidor es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

El servidor es el encargado de:

- ✚ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ✚ Procesar requerimientos de bases de datos.
- ✚ Formatear datos para transmitirlos a los clientes.
- ✚ Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Este patrón permite la combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, entre otros. Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad, capacidades del disco y dispositivos de entrada y salida.

3.2.2. Patrón arquitectónico Modelo-Vista-Controlador (MVC).

Symfony utiliza el patrón arquitectónico Modelo Vista Controlador (MVC) garantizando un modelo, una vista y una controladora de forma independiente. El mismo accede a realizar modificaciones en cada una de las partes sin afectar las demás, permitiendo que el desarrollo de aplicaciones sea rápido y sencillo. La figura

Capítulo 3: Construcción de la solución propuesta

3.1 muestra el funcionamiento interno de Symfony con la utilización del patrón arquitectónico MVC (Eguiluz, 2013).

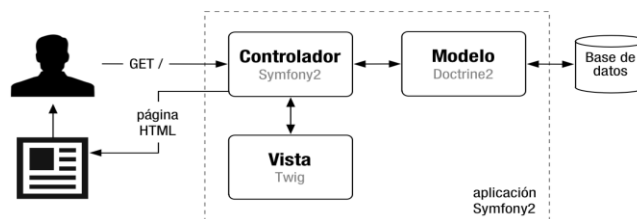


Fig. # 3.1: Esquema simplificado de la arquitectura interna de Symfony2 (Eguiluz, 2013).

El modelo es el responsable de:

- ✚ Encapsular los datos y las funcionalidades.
- ✚ Representar la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✚ Definir las reglas de negocio (la funcionalidad del sistema).
- ✚ Llevar un registro de las vistas y controladores del sistema.
- ✚ Si se está ante un modelo activo, notificará a las vistas los cambios en los datos.

En Symfony, el acceso y la modificación de los datos se realizan mediante objetos. Doctrine2 es el motor que se encarga de esta generación automática para construir sus clases, creando la estructura y generando el código de las mismas. Las clases y archivos relacionados con el modelo se guardan en el directorio `src/mi_Bundle/Entity/` (Eguiluz, 2013).

El controlador es responsable de:

- ✚ Recibir los eventos de entrada.
- ✚ Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

Es el intermediario entre el modelo y la vista. Se encarga de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o la vista. En Symfony2 todas las peticiones se realizan a través del controlador frontal `app.php`, el cual a través del enrutamiento delega las responsabilidades en las clases Controller de cada bundle (Eguiluz, 2013).

Las vistas son responsables:

- ✚ Transforma el modelo en una página web que permite al usuario interactuar con ella.
- ✚ Recibir datos del modelo y mostrarlos al usuario.

- ✚ Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- ✚ Pueden dar el servicio de "Actualización" para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Con el uso del Modelo-Vista-Controlador se logra la independencia de la base de datos. Se reutiliza la lógica de negocio para diferentes clientes o sistemas. Mejora considerablemente la escalabilidad y la flexibilidad.

3.2.3. Patrones de diseño.

Los patrones ayudan a manejar la complejidad del software. Cada patrón describe una manera para mejorar el problema que este resuelve. Cuando se encuentra un patrón que soluciona un problema de diseño en particular, no vale la pena invertir tiempo inventando una nueva solución. Si no, que aplicando correctamente el patrón se consigue obtener la solución al problema (Liebener, y otros, 2003).

3.2.3.1. Patrones para asignar responsabilidades (GRASP).

En el mundo de desarrollo de software se encuentran los Patrones de Software para la Asignación General de Responsabilidades (GRASP, por sus siglas en inglés). Entre estos se hallan Bajo Acoplamiento, Alta Cohesión, Experto, Creador y Controlador.

- ✚ **Experto:** Se evidencia en las entidades del modelo, ya que estas poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan. Ejemplo de ello son las entidades Datos del Edificio, Manzana, Edificio, entre otros.
- ✚ **Creador:** En la clase EntityManager se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase EntityManager es "creador" de dichas entidades. Ejemplos de algunas funciones utilizadas en la clase EntityManager son: findAll (), findByXXX (), findOneBy ().
- ✚ **Alta Cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Action, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las Properties.

- ✚ **Bajo Acoplamiento:** La clase `TematizarColorController` hereda únicamente de `Controller` para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.
- ✚ **Controlador:** Todas las peticiones web son manipuladas por un solo controlador frontal (`app.php`), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases `AppKernel`, `Controller`, `Request`, los “actions” y el `app.php` del ambiente.

3.2.3.2. Patrones GOF.

Los patrones GOF (Gang of Four, en español: Pandilla de los Cuatro) se clasifican en 3 categorías basadas en su propósito: comportamiento, creacionales y estructurales. El framework de desarrollo `Symfony2` emplea varios de estos patrones, los cuales son usados, pero no observados, en el desarrollo del módulo. Ejemplo de ellos son: Inyección de dependencias que según su propósito entra dentro del grupo de los creacionales, el patrón `Estrategia`, dentro de la categoría de comportamiento y `Decorador`, del tipo estructural (Hamon, 2014).

3.3. Modelo de diseño.

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que contiene los artefactos, clase de diseño, interfaz, paquete de diseño, subsistema de diseño, sucesos, señal, ejecución de guiones de uso, operación, realización de una operación, y componente de servicio (IBM, 2006).

3.3.1. Diagrama de clases del diseño del CU: Tematizar según tipo de estilo.

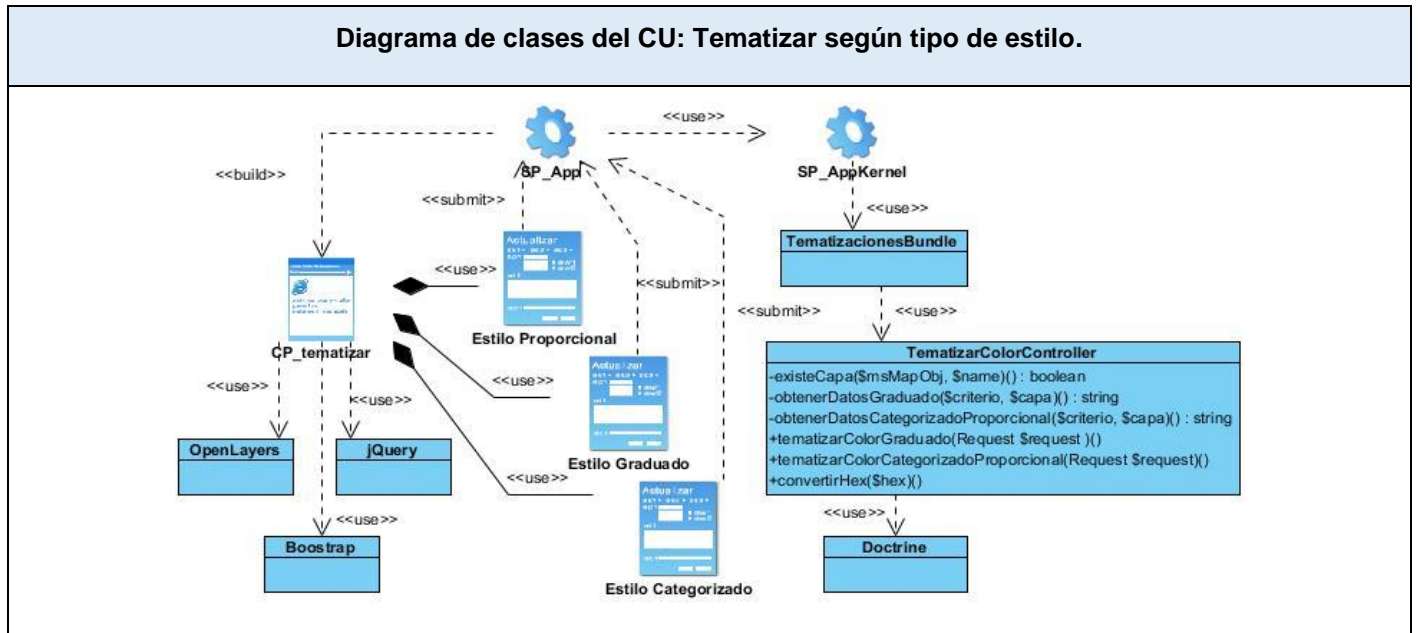


Fig. # 3.2: Diagrama de clases del CU: Tematizar según tipo de estilo.

3.3.2. Descripción de las clases.

- ✚ **OpenLayers:** Librería que se utiliza para cargar el mapa y para poder visualizar las capas tematizadas en él.
- ✚ **CP_tematizar:** Interfaz que brinda los formularios mediante el cual se capturan los datos para crear las distintas tematizaciones.
- ✚ **SP_App:** Controlador Frontal a través del cual se procesan todas las peticiones del usuario.
- ✚ **SP_AppKernel:** Clase que contiene los bundles e inicializa el bundle que se va a manejar la petición.
- ✚ **TematizacionesBundle:** Bundle que construye la tematización.
- ✚ **TematizacionColorController:** Controlador que maneja las acciones de la tematización.
- ✚ **Doctrine:** Librería que proporciona herramientas para simplificar el acceso y manejo de la información de la base de datos.
- ✚ **Estilo Proporcional:** Formulario para la captura de datos con la que se realiza la tematización según el tipo estilo proporcional.
- ✚ **Estilo Categorizado:** Formulario para la captura de datos con la que se realiza la tematización según el tipo estilo categorizado.

- ✚ **Estilo Graduado:** Formulario para la captura de datos con la que se realiza la tematización según el tipo estilo graduado.

3.4. Modelo de datos.

Un modelo de datos es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de reglas de integridad, las cuales definen un conjunto de estados consistentes. El cual puede utilizarse como una herramienta para especificar los tipos de datos y la organización de los mismos. Además para la manipulación de consultas y datos, así mismo es el elemento clave en el diseño de la arquitectura de un manejador de BD (Hernández, 2010).

3.4.1. Diagrama de Clases Persistente.

El objetivo fundamental del diseño de la base de datos es garantizar que los datos persistentes se almacenen de forma coherente y eficaz (IBM, 2006). Para diseñar la base de datos se utilizan el diagrama de clases persistentes y el diagrama entidad-relación. El diseño de la base de datos está compuesto por 5 tablas debidamente normalizadas para evitar la repetición de datos. A continuación se presentan en la **(Figura 3.3)** el Diagrama de Clases Persistentes de la solución propuesta.

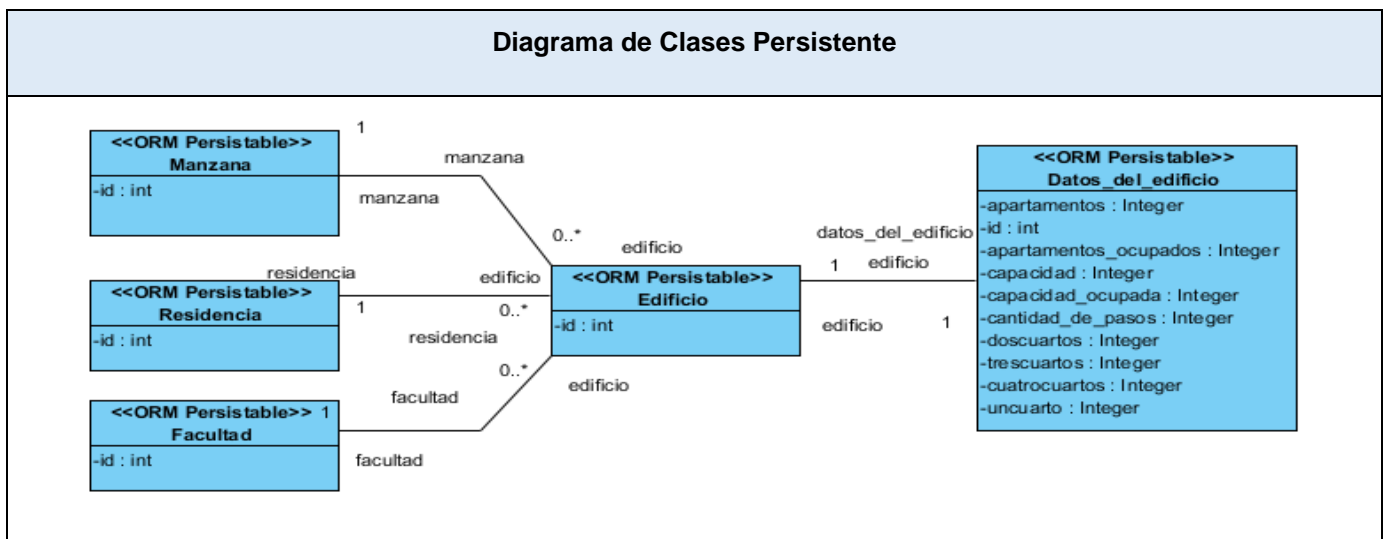


Fig. # 3.3: Diagrama de Clases Persistente de la solución.

3.4.2. Modelo Físico de la Base Dato.

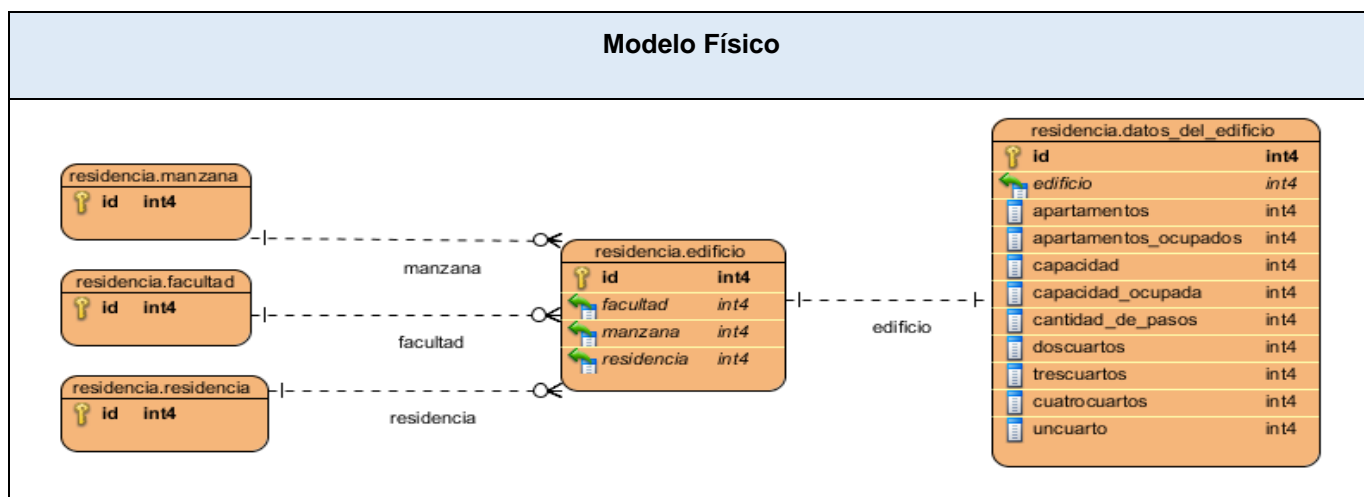


Fig. # 3.4: Modelo físico de la base datos.

3.5. Modelo de Despliegue.

Un modelo de despliegue es un modelo de objeto que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Utiliza como entrada fundamental las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (Rumbaugh, y otros, 2000).

En la **(Figura 3.4)** se representa el diagrama de despliegue del Módulo de análisis espacial estadístico descriptivo para el SIG UCI.

El diagrama de despliegue está compuesto por:

- ✚ **Nodos:** Elementos de procesamiento con al menos un procesador, memoria y otros dispositivos.
- ✚ **Conectores:** Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.
- ✚ **Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

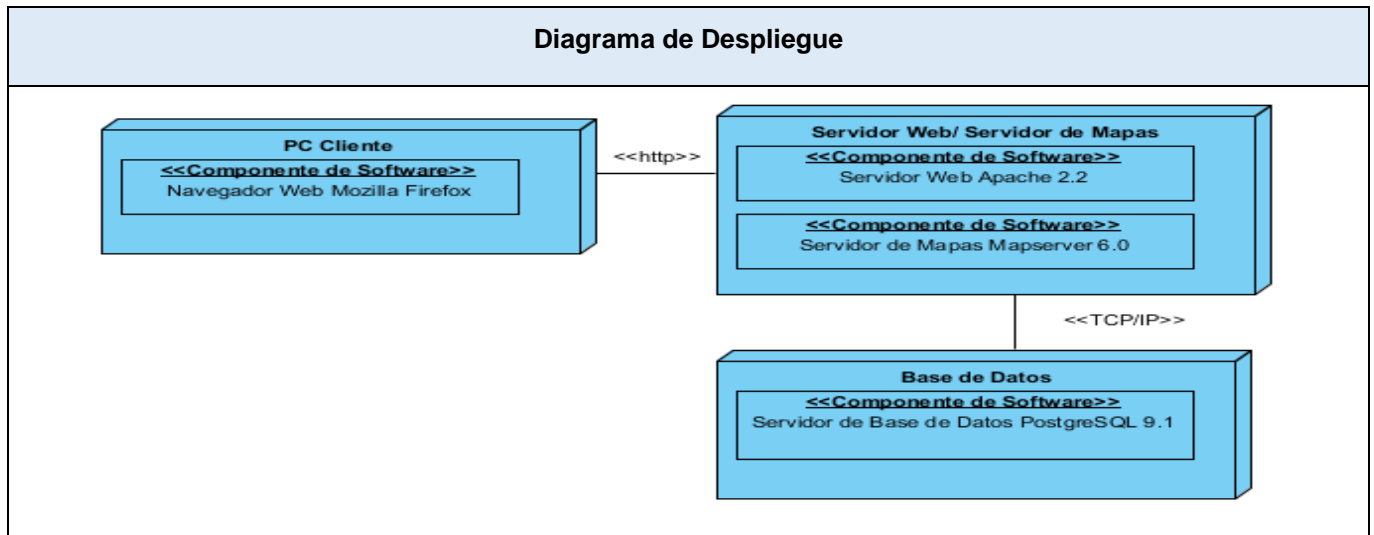


Fig. # 3.5: Diagrama de Despliegue de la Solución Propuesta.

Descripción de los Nodos.

- ✚ **PC Cliente:** Ordenador cliente capaz de conectarse al servidor de aplicaciones mediante el protocolo de comunicaciones HTTP.
- ✚ **Servidor Web/Servidor de Mapas:** Ordenador en que se encuentra el servidor web Apache y el servidor de mapas Mapserver, este será el lugar en que se gestione todo el contenido de la aplicación. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo HTTP y con el servidor de base de datos por medio del protocolo TCP/IP.
- ✚ **HTTP:** Protocolo de transferencia de hipertexto, método más común de intercambio de información.
- ✚ **Servidor de Base de Datos:** Ordenador en que se encuentra el gestor de base de datos PostgreSQL capaz de mantener persistente la información generada y a utilizar por la aplicación. El mismo establece comunicación con el servidor Web usando el protocolo TCP/IP.

3.6. Modelo de Componentes.

Los modelos de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Además de mostrar las dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados (Cockburn, 2005). En la **(Figura 3.5)** se representa el diagrama de componentes del CU Tematizar según tipo de estilo.

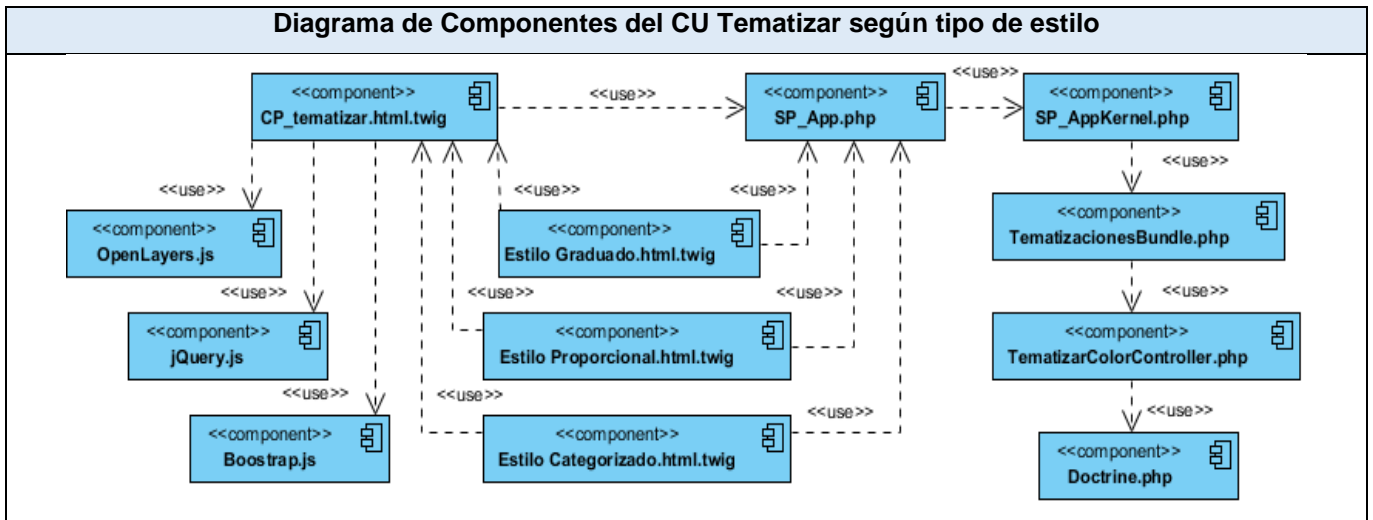


Fig. # 3.6: Diagrama de Componentes del CU Tematizar según tipo de estilo.

3.7. Estándares de codificación.

El estándar en el código de programación es muy importante en cualquier proyecto. Esto asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de entender. El estándar utilizado para el desarrollo del sistema fue el siguiente:

3.7.1. Notación Camello.

Consiste en escribir los identificadores de variables o funciones con la primera letra en minúscula y el resto en mayúscula o viceversa en dependencia de la variante que se escoja. El nombre se debe porque los identificadores recuerdan las jorobas de un camello. Un ejemplo de su utilización se muestra en la (Figura 3.7):

Notación Camello
<pre> private function existeCapa(\$msMapObj, \$name) { \$layers = \$msMapObj->getAllLayerNames(); foreach (\$layers as \$layer) { if (\$layer == \$name) return true; } return false; } </pre>

Fig. # 3.7: Ejemplo de Notación Camello.

3.8. Proceso de pruebas del Módulo de Análisis Espacial Estadístico Descriptivo para el SIG UCI.

Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso, externo al equipo de desarrollo, que determina si el software satisface los requisitos, y verificación como el proceso interno que determina si los productos de una fase satisfacen las condiciones de dicha fase (Pressman, 2005).

3.8.1. Pruebas Funcionales o de caja negra.

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software. Se trata de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado (Pressman, 2005).

Dentro del método de caja negra según (Quezada, 2007) existen algunas técnicas como:

- ✚ Partición de Equivalencia: consiste en dividir el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ✚ Análisis de Valores Límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ✚ Grafos de Causa-Efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Para el módulo desarrollado, dentro del método de caja negra, se aplicó la técnica Partición de Equivalencia.

3.8.2. Casos de Pruebas (CP).

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de usos del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requisitos funcionales del sistema (Quezada, 2007).

A continuación se hará referencia a solo un CP del CU Realizar tematización según tipo de estilo y el CP del CU Realizar tematización según tipo de gráfica será descrito para mayor información en el Expediente del Proyecto.

3.8.2.1. Caso de Prueba del Caso de Uso Realizar tematización según tipo de estilo.

- ✚ **Descripción general:** El caso de uso comienza cuando el actor del sistema decide realizar alguna tematización teniendo en cuenta los tipos de estilos


Capítulo 3: Construcción de la solución propuesta

(proporcional, categorizado, graduado). Este finaliza cuando es mostrada en una capa del mapa la tematización según opciones seleccionadas.


✚ **Condiciones de Ejecución:** No tiene.

3.8.2.2. Secciones y los escenarios a probar del CU:


Tabla # 3.1: Prueba del CU Realizar tematización según tipo de estilo, secciones a probar.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: "Estilo Proporcional".	EC 1.1: Realiza tematización según Estilo Proporcional correctamente.	El usuario selecciona la opción Crear Mapa Temático. El usuario selecciona la opción Estilo Proporcional e introduce los siguientes datos: -Capa. -Transparencia -Opacidad. El usuario luego de llenar los campos da clic en el botón Tematizar Mapa. El sistema realiza la tematización.	Módulo de Análisis Espacial Estadístico Descriptivo para el SIG UCI. 1. El usuario introduce los datos y da clic en el botón Tematizar Mapa. 2. El sistema muestra la tematización según estilo proporcional.
	EC 1.2: Cierra la ventana "Crear Tematización". 	El usuario decide no realizar tematización y cierra la ventana.	El usuario presiona la opción cerrar.
	EC 1.3: Presiona el botón "Cancelar".	El usuario decide no realizar tematización y presiona el botón "Cancelar".	El usuario presiona el botón "Cancelar".
	EC 1.4: Selecciona la opción ****Seleccione**** del campo criterio.	El usuario deja el campo criterio sin seleccionar la opción válida.	El usuario selecciona la opción ****Seleccione**** del campo criterio.
	EC 1.5: Deja campos opcionales sin llenar.	El usuario deja campos opcionales sin llenar y finaliza tematización	El usuario presiona el botón "Tematizar Mapa"

Capítulo 3: Construcción de la solución propuesta

		presionando el botón "Tematizar Mapa".	
SC 2: "Estilo Categorizado".	EC 1.1: Realiza tematización según Estilo Categorizado correctamente.	El usuario selecciona la opción Crear Mapa Temático. El usuario selecciona la opción Estilo Categorizado e introduce los siguientes datos: -Capa. -Criterio. -Transparencia. -Opacidad. -Filtro. El usuario luego de llenar los campos da clic en el botón Tematizar Mapa. El sistema realiza la tematización.	Módulo de Análisis Espacial Estadístico Descriptivo para el SIG UCI. 1. El usuario introduce los datos y da clic en el botón Tematizar Mapa. 2. El sistema muestra la tematización según estilo categorizado.
	EC 1.2: Cierra la ventana "Crear Tematización". 	El usuario decide no realizar tematización y cierra la ventana.	El usuario presiona la opción cerrar.
	EC 1.3: Presiona el botón "Cancelar".	El usuario decide no realizar tematización y presiona el botón "Cancelar".	El usuario presiona el botón "Cancelar".
	EC 1.4: Selecciona la opción ****Seleccione**** del campo criterio.	El usuario deja el campo criterio sin seleccionar la opción válida.	El usuario selecciona la opción ****Seleccione**** del campo criterio.
	EC 1.5: Selecciona la opción en blanco del campo parámetro estadístico.	El usuario deja el campo parámetro estadístico sin seleccionar la opción válida.	El usuario presiona el botón "Tematizar Mapa".

Capítulo 3: Construcción de la solución propuesta

	EC 1.6: Deja campos opcionales sin llenar.	El usuario deja campos opcionales sin llenar y finaliza tematización presionando el botón "Tematizar Mapa".	El usuario presiona el botón "Tematizar Mapa"
SC 3: "Estilo Graduado".	EC 1.1: Realiza tematización según Estilo Graduado correctamente.	El usuario selecciona la opción Crear Mapa Temático. El usuario selecciona la opción Estilo Graduado e introduce los siguientes datos: -Capa. -Criterio. -Transparencia. -Opacidad. El usuario luego de llenar los campos da clic en el botón Tematizar Mapa. El sistema realiza la tematización.	Módulo de Análisis Espacial Estadístico Descriptivo para el SIG UCI. 1. El usuario introduce los datos y da clic en el botón Tematizar Mapa. 2. El sistema muestra la tematización según estilo graduado.
	EC 1.2: Cierra la ventana "Crear Tematización". 	El usuario decide no realizar tematización y cierra la ventana.	El usuario presiona la opción cerrar.
	EC 1.3: Presiona el botón "Cancelar".	El usuario decide no realizar tematización y presiona el botón "Cancelar".	El usuario presiona el botón "Cancelar".
	EC 1.4: Selecciona la opción ****Seleccione**** del campo criterio.	El usuario deja el campo criterio sin seleccionar la opción válida.	El usuario selecciona la opción ****Seleccione**** del campo criterio.
	EC 1.5: Deja campos opcionales sin llenar.	El usuario deja campos opcionales sin llenar y finaliza tematización presionando el botón "Tematizar Mapa".	El usuario presiona el botón "Tematizar Mapa"

3.8.2.3. Descripción de variables.

Capítulo 3: Construcción de la solución propuesta

Tabla # 3.2: Descripción de las variables.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Capa	Campo de selección	No	Campo de selección para seleccionar la capa donde se mostrará la tematización en el mapa.
2	Criterio	Lista Desplegable	No	Lista desplegable de los criterios de análisis.
3	Parámetro estadístico	Lista Desplegable	Sí	Lista desplegable de los parámetros estadísticos.
4	Transparencia	Campo de selección	Sí	Campo de selección para seleccionar la transparencia con la que se mostrará la tematización en el mapa.
5	Opacidad	Campo de texto	Sí	Debe ser solo valores numéricos decimales de 0 a 1.
6	Operador	Lista Desplegable	Sí	Lista desplegable de los operadores de comparación.
7	Valor	Campo de texto	Sí	Debe ser solo valores numéricos.
8	Clases	Campo de texto	No	Debe ser solo valores numéricos de 1 a 10.

3.8.2.4. Resultados de las pruebas de caja negra.

Las pruebas de caja negra realizadas arrojaron, 2 casos de uso y por ello, 2 casos de pruebas. En la primera iteración se detectaron 5 no conformidades a las cuales se les dio solución. La segunda y última iteración arrojó 2 no conformidades, las mismas se resolvieron en su totalidad. En las dos iteraciones efectuadas se detectaron un total de 7 no conformidades, las cuales mayormente respondían a errores de bajo impacto en el correcto funcionamiento del sistema y todas tuvieron solución un tiempo máximo de 2 días, como se muestra en la (Figura 3.8).



Fig. # 3.8: Resultados de las pruebas funcionales.

3.8.3. Pruebas de Rendimiento.

Las pruebas de rendimiento se diseñan con el objetivo de asegurar que el sistema pueda procesar su carga esperada. Esto normalmente implica planificar una serie de pruebas en las que la carga se va incrementando regularmente hasta que el rendimiento del sistema se hace inaceptable. Estas pruebas se ocupan tanto de demostrar que el sistema satisface sus requerimientos, como de descubrir problemas y defectos en el sistema (Sommerville, 2005).

3.8.4. Prueba de Carga y Estrés.

Estas pruebas van realizando pruebas acercándose a la máxima carga del diseño del sistema, hasta que el mismo falla. Este tipo de pruebas, tiene dos funciones según (Sommerville, 2005):

1. Prueba el comportamiento de fallo de ejecución del sistema. Pueden aparecer circunstancias a través de una combinación no esperada de eventos en donde la carga sobre el sistema supere la máxima carga anticipada. Las pruebas de estrés verifican que las sobrecargas en el sistema provocan <<fallos ligeros>> en lugar de colapsarlo bajo su carga.
2. Sobrecargan el sistema y pueden provocar que se manifiesten defectos que normalmente no serían descubiertos.

3.8.4.1. Resultados de las pruebas de rendimiento.

Para la ejecución de las pruebas se utilizó la herramienta JMeter teniendo en cuenta el siguiente entorno de trabajo:

Capítulo 3: Construcción de la solución propuesta

1 PC cliente con 256 MB de memoria RAM, con el sistema operativo Ubuntu 12.04, mientras que la PC servidora contiene 2 GB de memoria RAM y 1 TB de espacio en el disco duro.

Carga:

Para las pruebas de carga, el tiempo promedio de ejecución de una petición es de 4 segundos, realizándose un total de 150 solicitudes al servidor sin errores, lo que significa que todas las peticiones fueron servidas correctamente. Además se puede observar un rendimiento de 6,4 peticiones por segundos y un tiempo máximo de ejecución de petición que no sobrepasan los 10 segundos. Por lo que se puede afirmar que el tiempo de respuesta del módulo es aceptable para 150 solicitudes que se realicen desde 30 máquinas que se conecten.

Se continuó realizando las mismas transacciones de carga de trabajo, pero para una cantidad variada de usuarios, obteniéndose los resultados expuestos en la siguiente tabla.

Tabla # 3.3: Resultados de la pruebas de carga.

Usuarios	Cantidad de Solicitudes	Tiempo de Respuesta	% de errores	Rendimiento (petición/segundo)	Tiempo Máximo de una petición
30	150	4.1	0	6.4	9.2
60	300	14.4	0	3.8	25.1
120	600	36.7	0	3.0	53.3

Estrés:

Esta prueba se realizó con un total de 200 usuarios conectados y 5 bucles²⁰, arrojando un total de 1000 peticiones. Para esta muestra el sistema demostró poca solidez pues el rendimiento demoró más de una hora.

²⁰ bucles: Es la cantidad de ejecuciones que desarrolla una estación de trabajo.

Conclusiones.

En el presente capítulo se identificaron los patrones de diseño GRASP y GOF permitiendo que la programación estuviese bien estructurada. Además se facilitó la asignación de responsabilidades logrando un diseño de software que sirviese de apoyo a la implementación del sistema. Con la realización del diagrama de clases del diseño se obtuvo una visión más exacta del sistema en términos de implementación, siendo de gran ayuda para el equipo de desarrollo. La realización del modelo de despliegue brinda una distribución completa del acople de los distintos componentes por lo que está compuesto el sistema. La confección del modelo de componentes permitió conocer cómo interactúan los diferentes subsistemas de implementación. El uso de la notación Camello, como estándar de codificación, garantizó que la implementación se realizara de manera organizada, permitiendo un mayor entendimiento de la misma. Se realizaron pruebas funcionales o de caja negra y pruebas de rendimiento, demostrando que las funciones del módulo son efectivas, producen un resultado satisfactorio y el sistema cumple con los objetivos trazados.

Conclusiones Generales.

A partir del desarrollo de la presente investigación las autoras arribaron a las siguientes conclusiones:

- ✚ El análisis y estudio de las tecnologías y tendencias necesarias para el desarrollo del sistema permitió definir los lenguajes, herramientas y métodos a seguir para su implementación.
- ✚ El uso del marco de trabajo Symfony2 ayudó a confeccionar una propuesta de solución que se ajustara a las pautas de codificación definidas para el desarrollo de software.
- ✚ Se definieron 5 requisitos funcionales, 7 no funcionales, y 2 casos de uso con sus respectivas descripciones, permitiendo especificar las características del módulo y sirviendo de guía para los flujos y fases posteriores.
- ✚ Para comprobar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron casos de prueba, los que arrojaron resultados satisfactorios, demostrando el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del módulo.
- ✚ El módulo implementado para el SIG UCI, como resultado de la investigación realizada, será de gran utilidad para los directivos del área residencial de la Universidad de las Ciencias Informáticas, permitiéndole una mejor toma de decisiones.

Recomendaciones.

Luego de la presente investigación se recomienda:

- ✚ Extender la investigación hacia otras áreas de la residencia de la universidad. Un ejemplo de esta sería el área residencial de los trabajadores.
- ✚ Que se integre lo más rápido posible este módulo al SIG UCI, para que el mismo ayude a los directivos de la residencia a una mejor toma de decisiones.

Bibliografía y Referencias Bibliográficas.

1. **Aguila, Adrián Gracia. 2009.** Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la facultad 9. La Habana : s.n., 2009.
2. **Alvarez, Miguel Angel. 2009.** Desarrollo Web. Desarrollo Web. [En línea] 25 de marzo de 2009. [Citado el: 14 de febrero de 2015.] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.
3. **Anon. 2008.** “Modelo de dominio « tecnología y synergix. “Modelo de dominio « tecnología y synergix. [En línea] 10 de 07 de 2008. <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
4. —. “Requerimientos funcionales y no funcionales.”. [En línea] [Citado el: 8 de noviembre de 2014.] <http://www.mitecnologico.com/Main>.
5. **Avendaño, Bertha Mariel Márquez y Zulaica Rugarcía, José Manuel . 2013.** Colección de Tesis Digitales de la Universidad de las Américas Puebla. [En línea] 03 de mayo de 2013. [Citado el: 02 de febrero de 2015.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.
6. **Bahit, Eugenia. 2011.** Arquitecta GLAMP & Agile Coach. Agile Coaching. . [En línea] . [En línea] 2011. [Citado el: 25 de 1 de 2015.] <http://www.eugeniahahit.com/mvc/>.
7. **Baños, Oliannis. 2012.** Análisis del módulo de Admisnistración de la Plataforma Atlas. . La Habana,Cuba : s.n., 2012.
8. **Camacho, Sunamy Ruiz. 2010.** Personalización de GeneSIG para la Universidad de las Ciencias Informáticas.Rol Analista de Sistemas. Habana : UCI, 2010.
9. **Cibernetia. 2010.** Ciberaula. Cursos para empresas. [En línea] 2010. [Citado el: 24 de 12 de 2014.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
10. **Cockburn, A. 2005.** Crystal Clear: A Human Powered Methodology for Small Teams. 2005.
11. **Cox, Colleen. 2011.** MapInfo Professional Guía del Usuario. New York : One Global View, Troy, 2011. 12180–8399.
12. **Eguiluz, Javier. 2011.** Desarrollo web ágil con Symfony2. s.l. : easybook versión 3.1, 2011. versión 1.

13. —. **2013**. Desarrollo web ágil con Symfony2. 2013.
14. **ESRI**. ArcGIS | Esri España. [En línea] [Citado el: 05 de 11 de 2014.] <http://www.esri.es/es/productos/arcgis/>.
15. —. **2014**. ArcGIS Resources. ArcGIS Help 10.2, 10.2.1, 10.2.2. [En línea] ESRI, 05 de 03 de 2014. [Citado el: 24 de 10 de 2015.] <http://resources.arcgis.com/en/help/main/10.2/index.html#//00s500000029000000>.
16. **Fabier, Francois y Zaninotto. 2007**. Symfony la guía definitiva. 2007.
17. **Fowler, Martin. 1999**. Uml Gota a Gota. Mexico : Addison Wesley Longman, 1999.
18. **G. Figueroa, Roberth, J. Solís, Camilo y A. Cabrera, Armando. 2011**. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2011.
19. **Gianfelici, Esteban. 2011**. Mapas y mapas.com.ar. . [Online] 07 2008. [En línea] 11 de 10 de 2011. [Citado el: 19 de 10 de 2014.] <http://www.mapasy mapas.com.ar/el%20dato%20geografico.php..>
20. **Giraldo , Luis y Zapata, Yuliana. 2005**. Herramientas de desarrollo de ingeniería de SW para LINUX. s.l. : [Presentación]), 2005.
21. **González del Bosque , Isabel, y otros. 2002**. Los Sistemas de Información Geográfica y la Investigación en Ciencias Humanas y Sociales. s.l. : Confederación Española de Centros de estudios Local.Apuntes de Ciencias Instrumentales y Técnicas de Investigación, 2002. ISBN: 978-84-615-9825-0.
22. **Goodchild, M.F, y otros. 2005**. Geographic Information Systems and Scienc. Chichester: Wiley. 2nd : s.n., 2005.
23. **Hamon, Hugo. 2014**. Identifying Design Patterns in the Symfony Framework. [PPT] Istanbul, Turkey : SensioLabs, 2014.
24. **Hernandez, A González. 2010**. Requisitos a partir del Modelo del Negocio. Ciudad de La Habana : Ingeniería Industrial, 2010.
25. **Higuera, Santiago. 2010**. Manual OpenLayers. [En línea] 07 de junio de 2010. <http://openlayers.bicimap.es/manualOpenLayers.html>.

26. **IBM. 2006.** Rational Unified Process. s.l. : IBM Corp, 2006.
27. **IEEE. 2004.** SWEBOOK. s.l. : Capítulo 2,Pág2, 2004.
28. **JUÁREZ, I. P. A. 2003.** Estadística Descriptiva. 2003.
29. **Lago, Ramiro. 2007.** [En línea] . [En línea] Abril de 2007. [Citado el: 28 de 01 de 2015.] <http://www.proactiva-calidad.com/java/patrones/mvc.html> .
30. **Larman, Craig y Hall, Prentice. 2003.** UML y patrones:una introducción al análisis y diseño orientado a objetos y al proceso unificado. s.l. : Pearson Educación, 2003. ISBN:8420534382.
31. **Liebener, L, M.A, Rossi y Plinker. 2003.** Relaciones entre los patrones de diseño. [En línea] 2003.
32. **Limia, Alberto Navarro. 2007.** Sistema de descarga y procesamiento automatizado de patentes. Diseño del Sistema. Ciudad de la Habana : s.n., 2007.
33. **Lugo, Elsa Karina Lopéz y Juárez García, Francisco. 2000.** Apuntes de estadística descriptiva. 2000.
34. **Mapserver. 2011.** MapServer open source web mapping. MapServer open source web mapping. [En línea] 2011. [Citado el: 03 de febrero de 2015.] <http://mapserver.org/trunk/es/about.html#about..>
35. **Mendoza, R F. 2007.** El lenguaje visual gráfico en geografía. 2007.
36. **Mondaray, Sergio Garcia. 2012.** GodTIC, un mundo binario. Ingeniería del Software, Programación. [En línea] 15 de 11 de 2012. [Citado el: 12 de 11 de 2014.] <http://tinyurl.com/asmbd5k>.
37. **Muñoz, D R. 2004.** Manual de Estadística. Juan Carlos Martínez Coll. s.l. : Casa del Libro, 2004.
38. **NetBeans.org. 2010.** NetBeans. [En línea]. [En línea] 2010. [Citado el: 17 de 12 de 2014.] <http://netbeans.org/features/index.html.>
39. **Olaya, Víctor. 2011.** Libro SIG Versión1.0. 2011.
40. **Olaya, Victor. 2009.** SEXTANTE Programming Guide. SEXTANTE Programming Guide. 2009. Edición 1.0.

41. **opengeo.org. 2014.** opengeo.org [En línea]. [En línea] 2014. [Citado el: 18 de 12 de 2014.] [http://opengeo.org/technology/postgis/.](http://opengeo.org/technology/postgis/)
42. **OSGeo. 2010.** INTEGRATING SEXTANTE AND GRASS. [En línea] 9 de septiembre de 2010. [Citado el: 15 de marzo de 2015.] http://2010.foss4g.org/presentations_show.php%3Fid=3382.html.
43. **Pérez, Luis Miguel Royo. 2014.** IniSIG. Plataforma de Formación y Divulgación SIG. [En línea] 10 de 2 de 2014. [Citado el: 09 de 10 de 2014.] <http://inisig.com/introduccion-al-analisis-espacial-i/>.
44. **Pressman, Roger. 2005.** Ingeniería de Software.Un enfoque práctico. España : Mc Graw Hil, 2005. 9701054733.
45. **Project, QGIS. 2015.** QGIS User Guide. 2015. Publicación 2.6.
46. **Quezada, Juan Antonio. 2007.** Pruebas de software. 2007.
47. **QUIÑONES, ERNESTO. 2007.** Introducción a postgresql. [En línea]. [En línea] 2007. [Citado el: 15 de 12 de 2014.] http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
48. **R.J, Chorley. 1972.** Spatial Analysis in Geomorphology Methuen. 1972.
49. **Rhind, David. 1989.** [aut. libro] Víctor Olaya. Libro SIG. 1989.
50. **Rumbaugh, G. , Jacobson, J. y Booch, I. 2000.** El Proceso Unificado de Desarrollo de Software. Madrid: Addison – Wesley : Edición en español por la Pearson Educación S.A, 2000.
51. **Sendra, J. Bosque. 2004.** Sistemas de Información Geográfica y localización de instalaciones y equipamientos . Ra-Ma : s.n., 2004.
52. **Sinton, D.F. 1978.** The inherent structure of information as a constraint to analysis: Mapped thematic data as a case study. In Harvard Papers on Geographic Information Systems. s.l. : Addison-Wesley, 1978. vol. 6, pages 117. .
53. **Sommerville, Ian. 2005.** Ingeniería del software.Séptima Edición. Madrid : Pearson Educación. S.A, 2005. ISBN 84-7829-074-5.
54. **Torres, Alexander Rodríguez. 2010.** Servicio de Mapas Temáticos. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.

55. **Vázquez, Jesús Palomar. 2009.** Introducción a gvSIG. Introducción a gvSIG. [En línea] noviembre de 2009. [Citado el: 10 de Mayo de 2015.] http://cgat.webs.upv.es/bigfiles/gvsig/gvsig_112.htm?t666.html.
56. **Victor Puertas. 2013.** Las novedades NetBeans 7.3 para Symfony y PHP. Las novedades NetBeans 7.3 para Symfony y PHP. [En línea] 28 de enero de 2013. [Citado el: 28 de febrero de 2015.] <http://yosymfony.com/las-novedades-netbeans-7-3-para-symfony-y-php/#more-73>.
57. **Visual Paradigm. 2006.** [En línea] 2006. [Citado el: 5 de febrero de 2015.] <http://www.visualparadigm.com/product/vpuml/>.
58. **Wieggers, Karl. 2003.** Software Requirements. SecondEdition. Chapter1. Versión en .chm. 2003.
59. **Zaldívar, Yoenis Pantoja. 2010.** MaGIStral : Sistema Integral de gestion de recursos para el proyecto GENESIG. La Habana : s.n., 2010.
60. **Zayas, Dr. Carlos Álvarez de. 2007.** METODOLOGÍA DE LA INVESTIGACIÓN CIENTÍFICA. Santiago de Cuba : s.n., 2007.
61. **Zylberberg, Alejandro D. 2005.** Probabilidad y estadística. Buenos Aires : s.n., 2005. ISBN 987-1104-33-2.

Anexos.

Preguntas realizadas en las entrevistas

1. ¿Cuál es la estructura del área donde labora?
2. ¿Cuál o cuáles son los procesos fundamentales que manejan en el área?
3. ¿Tienen algún sistema que maneje y gestione esta información?
4. ¿Han encontrado errores en la información que se maneja?
5. ¿Qué información se maneja en los reportes generados en estos procesos?

Otras posibles preguntas:

6. ¿Qué tipo de información se maneja en su área?
7. ¿Cómo se realizan los procesos para obtener esa información?
8. ¿Cuál es el tiempo mínimo en recopilarla?
9. ¿Qué reportes se generan a partir esta?
10. ¿Cómo usted cree que el módulo a desarrollar le ayudaría en su trabajo?