

Universidad de las Ciencias Informáticas

Facultad de Ciencias y Tecnologías Computacionales

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



Título: Módulo de análisis temático para el Sistema de Información Geográfica de la Universidad de Ciencias Informáticas v3.0.

Autores

Luis David González Sarduy


Asley Arias Sainz

Tutores

MsC. Lisbeth Olinda López Verdecie

Ing. Yosiel Medero Cuello

La Habana, junio de 2017



*“La inteligencia consiste no solo en el conocimiento,
sino también en la destreza de aplicar los
conocimientos en la práctica”.*

Aristóteles

DECLARACIÓN DE AUTORÍA

Nosotros, Luis David González Sarduy y Asley Arias Sainz, declaramos ser los autores del presente trabajo de diploma titulado “Módulo de análisis temático para el Sistema de Información Geográfica de la Universidad de Ciencias Informáticas v3.0” y cedemos los derechos patrimoniales a la Universidad de las Ciencias Informáticas con carácter exclusivo.

Para que así conste, firmamos a los ____ días del mes de _____ del año _____.

Firma del Autor
Luis David González Sarduy

Firma del Autor
Asley Arias Sainz

Firma del Tutor
MsC. Lisbeth Olinda López Verdecie

Firma del Tutor
Ing. Yosiel Medero Cuello

SÍNTESIS DE LOS TUTORES

MsC. Lisbeth Olinda López Verdecie

Graduada de Ingeniera en Ciencias Informáticas. Máster en Gestión de Información desde el año 2014. Profesora asistente con 6 años de experiencia en la docencia y desarrollo de software. Actualmente se desempeña como subdirectora del centro de Geoinformática y Señales Digitales.

Ing. Yosiel Medero Cuello

Ingeniero en Ciencias Informáticas, alcanzó el Nivel Superior en el año 2015 en la Universidad de las Ciencias Informáticas. Cuenta con 3 años de experiencia como programador en la línea de los Sistemas de Información Geográfica como parte del proyecto GeneSIG del Centro Geoinformática y Señales Digitales (GEYSED) de la Universidad de las Ciencias Informáticas. Se ha desempeñado como programador de la Línea de Productos de Software Aplicativos SIG. Actualmente se desempeña como jefe de proyecto y desarrollador del producto Sistema de Información Geográfica para la UCI v3.0.

DEDICATORIA

A Dios por su bondad y fidelidad, gracias a su amor y misericordia hoy me puedo graduar.

A mi mamá y a mi papá por su ejemplo y amor constante.

A mi hermana y mi sobrinito por su preocupación y atención. En especial a mis abuelos José Luis y Santiago Regino que aunque físicamente ya no están conmigo siempre están en mi pensamiento y hoy puedo cumplir uno de sus mayores sueños en convertirme en un profesional y un hombre de bien para la sociedad.

Luis David

Dedico mi tesis y todos estos años de estudios a mi abuelo Juan Osvaldo Sainz Tamarit, a mi hermano Javier Sainz, a mis exsuegros Trujillo, Gladis y a Yusleidy, por estar en cada paso de mi vida universitaria.

Asley

AGRADECIMIENTOS

En primer lugar, le doy gracias a Dios por haber pensado en mí cuando nadie lo había hecho, por darme aliento de vida, por escogerme los padres ideales y gracias a Él estoy aquí hoy. En segundo lugar, agradezco a esa familia que Dios me dio, a mi mamá y a mi papá por creer en mí, por apoyarme siempre y nunca darme la espalda en los momentos más difíciles y que hicieron todo lo que estaba a su alcance y más para que nunca me faltara de nada en estos 5 años, por eso y por mucho más a ustedes doy gracias y Felicidades ya que este también es el fruto de su esfuerzo. Agradezco a mi hermana que también ha sido mi verdadera amiga, gracias por demostrarme que nada es imposible cuando luchas con todo para lograrlo y que rendirse nunca es una opción, a mi sobrinito Ulises que por él me esfuerzo cada día y trato siempre de ser un buen ejemplo para él. Agradezco también a mi abuela Ángela, a mis tías Mary y Raquel que siempre han estado pendientes de mí, a mi tío Joaquín y Juan gracias por toda la ayuda y siempre estar atentos, a mis primos Dayron, Joffre y Diley. Agradezco también a mis tutores Lisbeth y Yosiel que gracias a la ayuda de ellos en todo momento y que siempre estuvieron muy atentos hoy me puedo graduar, decirles que muchas gracias. Agradezco también a esos amigos que siempre han estado conmigo sobre todo a Ranniel, Gisell, Lily, Haileen y José Rubén, gracias porque siempre estaban ahí cuando más lo necesitaba. Agradezco a mis hermanos en la fé que durante este último año estuvieron pendientes de esta investigación en especial a Eliany, Raimel, Alexei y Robín. En fin, son muchas las personas que de un modo u otro me han apoyado durante estos 5 años, solo me queda decirles que muchas gracias.

Luis David

A Ernesto Alfredo Molina, Regino Leyva, Adaris, Raimel, Leandro, Noel, Osmanis Gallardo a los amigos que siempre estuvieron a mi lado desde los inicios de la universidad y los que conocí durante todos estos años, A los profesores que aguantaron malcriadeces mías y para los que no,

a

mis tías de edificio, a mi profesora la Negra que con un corazón grande siempre muy atenta a sus estudiantes,

a mis tutores Yosiel Medero, Lisbeth Olinda, Ronny, Yadiel, Aris, Ramón, Rubén, Pedro, Ameth, Javier, Merino, Juliet mi hermanita de corazón, Boris, Ever, Eime, Margarita, a mi amiga la del café del framboyán Yarelis, Juan, Osvaldo, Carlos a todos los frikis de la universidad por hacer mis días menos aburridos y por ser buenos amigos, Albuernes que, aunque se le olvide muchas veces las letras siempre rompiendo el aburrimiento con su guitarra y Rubén con su voz, Rene a esas personas que fueron y son especiales en mi vida. A los locos de mi apartamento, y a todos aquellos que hoy no están en la escuela, a todos ellos muchas Gracias.

Asley

RESUMEN

En el presente trabajo de investigación se propone realizar la implementación de un módulo de análisis temático para el Sistema de Información Geográfica de la Universidad de Ciencias Informáticas. Para la implementación se definió como metodología a seguir para regir el curso de vida del producto AUP_UCI, utilizando el escenario número cuatro de historias de usuarios. Se confeccionaron los diagramas correspondientes al diseño, modelo de datos, y a los componentes que interactúan en el sistema, además se siguió una línea de estandarización de código para lograr una comunicación en el equipo de desarrollo. Una vez terminada la implementación se definieron un conjunto de pruebas para comprobar si el sistema cumple con las funcionalidades propuestas por el cliente y comprobar si en realidad satisface sus necesidades. Como principal resultado se obtuvo un módulo de análisis temático y la información ingenieril asociada al mismo, este módulo permite la representación temática del área de la residencia de la universidad con el objetivo de facilitar el proceso de toma de decisiones de los directivos que allí laboran.

Palabras Clave: Sistemas de Información Geográfica (SIG), análisis temático, módulo.

ABSTRACT

In this research work is proposed the implementation of a module of thematic analysis for the Geographic Information System of the University of Computer Science. For the implementation was defined as methodology to follow to govern the lifetime of the product AUP_UCI, using scenario number four of user stories. The diagrams corresponding to the design, data model, and the components that interacted in the system were also made, and a line of code standardization was followed to achieve communication in the development team. Once the implementation was completed, a set of tests were defined to verify if the system complies with the functionalities proposed by the client and to verify if it actually satisfies their needs. As main result was obtained a thematic analysis module and associated engineering information, this module allows the thematic representation of the area of residence of the university with the objective of facilitating the decision-making process of the managers who work there.

Keywords: Geographic Information Systems (GIS), thematic analysis, module.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN 1

CAPÍTULO 1: Fundamentación Teórica 5

 1.1 Definición de mapas 5

 1.1.1 Componentes del mapa temático 6

 1.2 Técnicas de representación de datos en mapas temáticos 7

 1.2.1 Datos cualitativos 7

 1.2.2 Datos cuantitativos 7

 1.3 Análisis espacial 9

 1.4 Soluciones existentes 10

 1.4.1 Quantum GIS (QGis) 10

 1.4.2 GeneSIG 11

 1.4.3 Servicio de mapas temáticos 12

 1.4.4 Módulo de análisis estadístico descriptivo para el SIG_UCI v2.0 12

 1.4.5 UDig 12

 1.4.6 Características del SIG_UCI v3.0 13

 1.5 Metodología y Tecnologías de desarrollo 14

 1.5.1 Metodología de desarrollo de software 14

 1.5.2 Lenguaje de Modelado Unificado (UML) v2.1 16

 1.5.3 Visual Paradigm v8.0 17

 1.5.4 Lenguaje de programación JavaScript v1.8 17

 1.5.5 Entorno de Desarrollo Integrado WebStorm v2016.1 18

1.5.6	Servidor web Nodejs v4.4.3	18
1.5.7	ExpressJS v4.0.0	18
1.5.8	AngularJS v1.5.6	18
1.5.9	Librería OpenLayers v3.14	19
1.5.10	MapServer v6.4	19
1.5.11	Sistema Gestor de Base de Datos PostgreSQL v9.3	19
1.5.12	Extensión PostGIS v2.1	20
1.5.13	Sequelize v3.0.0	20
1.6	Conclusiones parciales	20
CAPÍTULO 2: Propuesta de Solución		22
2.1	Especificación de los requerimientos del sistema	22
2.1.1	Requisitos funcionales	22
2.1.2	Requisitos no funcionales (RnF)	24
2.2	Descripción de la solución propuesta	26
2.3	Análisis y diseño	28
2.3.1	Arquitectura de software	28
2.3.2	Patrón arquitectónico Modelo Vista Controlador	29
2.3.3	Patrones de diseño	30
2.3.4	Diagrama de clases del diseño	32
2.3.5	Diseño de la base de datos	34
2.4	Conclusiones parciales	35
CAPÍTULO 3: Implementación y Prueba		37

3.1	Diagrama de componentes	37
3.2	Diagrama de despliegue	37
3.3	Estándares de codificación	38
3.4	Pruebas de la propuesta de solución	40
3.4.1	Definición y descripción de las pruebas	41
3.5	Conclusiones parciales	47
CONCLUSIONES GENERALES.....		48
RECOMENDACIONES		49
Bibliografía.....		50
GLOSARIO DE TÉRMINOS.....		52
ANEXOS.....		54
Anexo 1.....		54
Anexo 2.....		54
Anexo 3.....		62

ÍNDICE DE FIGURAS

Figura 1: Fases de la metodología AUP 15

Figura 2: Patrón Modelo Vista Controlador..... 29

Figura 3: Patrón Controlador..... 31

Figura 4: Patrón Experto..... 31

Figura 5: Patrón Creador..... 31

Figura 6: Patrón inyección de dependencias..... 32

Figura 7: Diagrama de clases del diseño correspondiente al paquete Tematizaciones. 33

Figura 8: Modelo Entidad Relación. 34

Figura 9: Diagrama de componentes de la solución..... 37

Figura 10: Diagrama de despliegue de la solución..... 38

Figura 11: Estándares de codificación para márgenes, indentación, líneas y espacios..... 39

Figura 12: Estándares de codificación para variables, identificadores y parámetros. 40

Figura 13: Gráfica de los resultados de las pruebas de caja negra. 45

ÍNDICE DE TABLAS

Tabla 1: Historia de usuario del requisito "Realizar tematización según estilo gráfico de barras".26

Tabla 2: Descripción de los atributos de la tabla *tematization*.....35

Tabla 3: Caso de prueba "Realizar tematización según estilo gráfico de barras".....43

Tabla 4: Descripción de variables del caso de prueba "Realizar tematización según estilo gráfico de barras".
.....44

Tabla 5: Resultados de las pruebas de caja negra.44

Tabla 6: Resultado de las pruebas de carga.46

Tabla 7: Historia de usuario del requisito "Realizar tematización según estilo proporcional."54

Tabla 8: Historia de usuario del requisito "Listar tematizaciones."56

Tabla 9: Historia de usuario del requisito "Realizar tematización según estilo gráfico de pastel."58

Tabla 10: Historia de usuario del requisito "Realizar tematización según estilo categorizado.".....60

Tabla 11: Caso de prueba del requisito "Realizar tematización según estilo gráfico de pastel"62

Tabla 12: Caso de prueba del requisito "Realizar tematización según estilo categorizado"63

Tabla 13: Caso de prueba del requisito "Listar tematizaciones"64

Tabla 14: Caso de prueba del requisito "Visualizar tematizaciones"65

Tabla 15: Caso de prueba del requisito "Eliminar tematizaciones"65

Tabla 16: Caso de prueba del requisito "Realizar tematización según estilo proporcional".....66

INTRODUCCIÓN

Actualmente las Tecnologías de la Información y las Comunicaciones (TICs) constituyen un factor decisivo en el desarrollo económico y productivo de los países. El desarrollo del potencial de las nuevas generaciones depende del aprovechamiento que puedan hacer del uso de las TICs. El marcado avance de las TICs ha llevado a la creación de nuevas herramientas en diversos campos, las cuales permiten mejorar el proceso de toma de decisiones en toda la sociedad. Una de las más importantes en este sentido son los Sistemas de Información Geográfica (SIG).

Los SIG son herramientas que permiten la lectura, edición, almacenamiento y en términos generales la gestión de los datos espaciales. Mediante los SIG la información espacial puede ser aprovechada en mayor medida y convertirse en un elemento sumamente enriquecedor y clave para muchos análisis (OLAYA 2014).

Entre las potencialidades brindadas por los SIG, se encuentran los mapas temáticos. Estos están hechos para reflejar un aspecto particular de la zona geográfica sobre la que se definen. Pueden centrarse en variables físicas, sociales, políticas, culturales y económicas y cualquier otra relacionada con un territorio concreto (CARTOGRÁFICA DE CANARIAS 2015). Los mapas temáticos permiten entender la información asociada a cualquier evento socioeconómico en aras de hacer comparaciones y dar un criterio propio acerca del evento cartografiado.

Existen en la actualidad diversos tipos de mapas temáticos, entre los más comunes están los mapas de gráficos y los de coropletas, ambos son de gran importancia para representar información cuantitativa y cualitativa. En Cuba existen diferentes instituciones que hacen uso de estas potencialidades, ejemplo de ello es en las empresas que transportan mercancías a todo lo ancho del país, representando mediante tematizaciones el recorrido de sus vehículos y el peso de la carga que transportan sobre ellos. También, son utilizados para representar situaciones del pasado ya sean de carácter político o económico, ejemplo los mapas de carácter histórico.

En el centro de Geoinformática y Señales Digitales (GEYSED) de la Universidad de las Ciencias Informáticas (UCI), la Línea de Productos de Software Aplicativos SIG, es el equipo encargado de la construcción y personalización de Sistemas de Información Geográfica sobre plataformas web, para diferentes tipos de negocios. Actualmente desarrolla un Sistema de Información Geográfica para la universidad, que permite la visualización, localización y consulta de objetivos socioeconómicos de la institución como edificios docentes, de residencia, plazas, manzanas, consultorios, cajeros, áreas deportivas, cafeterías, entre otros. Sin embargo, este software no cuenta con un mecanismo de análisis que represente la información asociada a indicadores estadísticos, potencialidad de los SIG que contribuye

considerablemente a la toma de decisiones.

Los directivos en la universidad actualmente obtienen la información para la toma de decisiones de diferentes fuentes, esto implica la utilización de varias herramientas para su procesamiento y análisis, entre las que se incluyen generalmente las del paquete Office, tal es el caso cuando se refiere a los datos de los inmuebles pertenecientes a la universidad o el área de la residencia. Esta forma de comprensión de los datos es costosa en tiempo y esfuerzo, y aumenta la posibilidad de introducir errores e impide evaluar cómo está distribuido determinado indicador en un área específica.

Por lo antes expuesto se identifica el **problema de la investigación**: ¿Cómo favorecer el proceso de toma de decisiones de los directivos que laboran en el área de la residencia de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** son los mapas temáticos en Sistemas de Información Geográfica, enmarcado en el **campo de acción**: los mapas temáticos en el Sistema de Información Geográfica para la Universidad de Ciencias Informáticas v3.0.

Para dar solución al problema se plantea como **objetivo general**: Desarrollar un módulo para crear mapas temáticos en el SIG_UCI v3.0, lo que facilitará el proceso de toma de decisiones de los directivos que laboran en el área de residencia de la universidad.

Para guiar la investigación se definen las siguientes preguntas científicas:

1. ¿Qué referentes teórico-metodológicos sustentan el proceso de creación de mapas temáticos en SIG?
2. ¿Qué características tiene la creación de mapas temáticos en el Sistema de Información Geográfica de la Universidad de Ciencias Informáticas en su versión 3.0?
3. ¿Cómo organizar el proceso de desarrollo del módulo para el SIG_UCI v3.0 que permita crear mapas temáticos?
4. ¿El sistema desarrollado favorece el proceso de toma de decisiones de los directivos que laboran en el área de residencia de la Universidad de las Ciencias Informáticas?

Para llevar a cabo el objetivo de la investigación se trazaron las siguientes tareas a realizar:

1. Determinación de los referentes teórico-metodológicos que sustentan la creación de mapas temáticos en SIG.
2. Valoración de las soluciones informáticas que permiten crear mapas temáticos, sus limitaciones y fortalezas.

3. Caracterización del proceso de creación de mapas temáticos en el SIG_UCI v3.0.
4. Desarrollar el módulo para el SIG_UCI v3.0 que permita crear mapas temáticos.
5. Diseñar las pruebas correspondientes para validar el módulo de análisis temático.

En la investigación se emplearon los siguientes métodos científicos:

Métodos Teóricos:

Histórico-Lógico: Este método se utilizó para obtener información acerca de los Sistemas de Información Geográfica que implementan mapas temáticos y cómo éstos se han desarrollado a lo largo de la historia tanto a nivel mundial como en la universidad.

Analítico-Sintético: Se empleó este método para descomponer el objeto de estudio y estudiar sus partes con facilidad y así descubrir las características fundamentales de los mismos y extraer los elementos significativos para la construcción del módulo.

Método Empírico:

Análisis documental: Este método es utilizado para analizar todos los documentos y conceptos que sustentan la investigación y así seleccionar los más adecuados para la misma.

Resultados esperados:

- Un módulo de análisis temático que facilite el proceso de toma de decisiones de los directivos que laboran en el área de residencia de la universidad a través de gráficos, símbolos y colores.
- Documentación con la información ingenieril asociada al proceso de implementación del módulo.

El documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación Teórica: En este capítulo se presentan los conceptos asociados al dominio del problema, como los mapas temáticos y sus características, además de los conceptos asociados al análisis espacial. Se realiza un análisis de soluciones existentes que pueden aportar elementos de interés para el desarrollo del módulo, y se describen las tecnologías a utilizar en el desarrollo de la solución, así como la metodología a seguir para la construcción del producto.

Capítulo 2: Propuesta de Solución: En este capítulo se definen los requisitos no funcionales y funcionales del sistema. Se describen las historias de usuarios y se confeccionan los diagramas de clases del diseño y del modelo de datos, así como la arquitectura que guía la implementación de la solución y los patrones de diseño empleados para la construcción del módulo.

Capítulo 3: Implementación y Prueba de la Solución: En este capítulo se confeccionan los diagramas

correspondientes al despliegue y los componentes principales del módulo, se especifican los estándares de codificación seguidos por el equipo de desarrollo. Se definen las pruebas que se realizan con el objetivo de que la solución cumpla con el correcto funcionamiento de los requisitos funcionales propuestos por el cliente.

CAPÍTULO 1: Fundamentación Teórica

En el presente capítulo se exponen los principales elementos teóricos que sustentan la investigación, relacionados con los mapas temáticos y el análisis espacial. Además, se incluye el análisis de soluciones existentes que permiten crear mapas temáticos en SIG. Por último, se hace un análisis de la metodología a seguir para el proceso de desarrollo del módulo y se describen las tecnologías empleadas para la construcción de la solución.

1.1 Definición de mapas

Según (CIENCIA GEOGRÁFICA 2014) *“Los mapas geográficos son representaciones planas de la superficie terrestre, son bidimensionales ya que tienen dos dimensiones largo y ancho, se representan mediante coordenadas geográficas, dígame longitud y latitud, que sirven para marcar los grados sobre la superficie de la tierra”.*

En definición de los autores los mapas son aquellas imágenes que se representan gráficamente, brindan información de un área geográfica determinada, así como sus características. En ellos se pueden representar los distintos terrenos que puede tener una superficie. Tienen gran utilidad en todos los ámbitos ya que recogen información de factores útiles para la sociedad.

Los mapas pueden ser clasificados por diversos criterios, dependiendo del interés particular que se tenga sobre ellos. Según (RODRÍGUEZ 2010) los mapas se clasifican atendiendo al formato en que son almacenados, su escala de trabajo y de acuerdo al tipo de mapa que es. A continuación, se describen estas clasificaciones:

Formato en que son almacenados: Según su formato existen dos tipos, el formato ráster y el formato vectorial. Cada uno de ellos influye en el comportamiento de los datos brindando una información de utilidad en cada uso.

- **Formato Ráster:** Los espacios son reducidos a pequeños cuadrados y rectángulos, donde el tamaño es fundamental y a la vez determinan una solución. La zona de estudio se divide de forma sistemática en una serie de unidades mínimas denominadas celdas, y para cada una de estas se recoge la información que la describe.
- **Formato Vectorial:** Se basa en tres primitivas básicas.
 - Nodo o punto, que es la unidad básica para representar entidades con posición, pero sin dimensión.
 - La línea o el arco que representan las entidades de una dimensión, y están restringido a línea recta

en algunas implementaciones.

- Polígono o área que se utiliza para representar las entidades bidimensionales.

Escala de trabajo: La escala en cartografía es definida como la relación matemática que existe entre las dimensiones reales y las del dibujo que representan en la realidad en un mapa plano, esquema o croquis (IBÁÑEZ, MORENO y GISBERT 2012).

- **Mapas de Escala Pequeña:** Son aquellos que presentan una escala numérica de 1:50.000, ejemplo los que representan países, continentes y hemisferios.
- **Mapas de Escala Mediana:** Son aquellos que presentan una escala numérica en un rango de 1:50.000 y 1:250.000
- **Mapas de Escala Grande:** Son aquellos que se representan en una escala numérica de 1:250.000 y menores.

En cuanto al **tipo de mapa**, se clasifican en mapas topográficos y mapas temáticos.

- **Mapas Topográficos:** El mapa topográfico es una representación de la superficie terrestre mediante curvas de nivel que tiene como finalidad mostrar las variaciones del relieve de la Tierra. El manejo de este tipo de mapa permite cuantificar distancias, superficies y volúmenes y se pueden elaborar una infinidad de cartografías temáticas, como un mapa de suelo, de vegetación, de cultivos u otros (CUADRADO 2016).
- **Mapas Temáticos:** *“Los mapas temáticos están hechos para reflejar un aspecto particular de la zona geográfica sobre la que se definen. Pueden centrarse en variables físicas, sociales, políticas, culturales y económicas y cualquier otra relacionada con un territorio concreto. Están diseñados con un propósito específico o para ilustrar un tema determinado, tales como la orografía del terreno, las construcciones, las vías de comunicación entre otros”* (CARTOGRÁFICA DE CANARIAS 2015).

“Los mapas temáticos utilizan los mapas topográficos como mapa base para la representación gráfica de datos de diversa índole lo que se conoce como cartografía temática”(RODRÍGUEZ 2010).

Un mapa temático se puede definir como un mapa que de acuerdo a un evento permite conocer mediante estadísticas la situación actual de un problema específico. Está basado en mapas topográficos con el objetivo de brindar información relevante para la toma de decisiones.

1.1.1 Componentes del mapa temático

Todo mapa temático está compuesto por dos elementos fundamentales, el mapa base y el componente

temático, una vez que estos se integran crean un mapa temático.

El **mapa base** proporciona información espacial sobre la cual se puede referenciar el contenido propio correspondiente a un cierto tema específico. Debe estar correctamente diseñado e incluir únicamente la cantidad de información necesaria (GUTIÉRREZ 2012).

La **componente temática** consiste en la representación gráfica de datos que representan un fenómeno particular (GUTIÉRREZ 2012).

1.2 Técnicas de representación de datos en mapas temáticos

Para brindar información los mapas temáticos trabajan con datos cualitativos o cuantitativos. Con estos datos que representan sucesos o estados de la superficie terrestre, se puede clasificar un evento.

1.2.1 Datos cualitativos

Los mapas cualitativos son aquellos que representan cualidades sobre un evento o suceso determinado. Estos datos están clasificados en puntuales, lineales y superficiales (RODRÍGUEZ 2010). A continuación, se describen éstos.

- **Mapas de representación de datos Puntuales:** Identifican un fenómeno y los sitúa según sus coordenadas. Los símbolos que lo representan pueden ser geométricos, dígame cuadrados, puntos, rectángulos, círculos, entre otros. Es fundamental que en la leyenda se aprecie los diferentes símbolos y el color que se utiliza en cada variable.
- **Mapas de representación de datos Lineales:** Es utilizado para representar fenómenos de forma lineal dígame carreteras, ríos, fronteras, rutas de viajes, entre otras. En estos casos los símbolos son representados por el color y la forma, algunos de ellos describen el mismo fenómeno, pero aun así se deben representar de forma diferente. Ejemplo de ello las carreteras de automóviles y las líneas de trenes que representan el mismo evento (carreteras), pero no se representan iguales.
- **Mapas de representación de datos Superficiales:** Estos mapas brindan información sobre la distribución de fenómenos que ocupan extensiones superficiales. Ejemplo son los mapas de suelos, geológicos, forestales, entre otros. Como simbolización utiliza varios colores que permiten distinguir las categorías. Es fundamental que se aprecie en la leyenda los símbolos y color de cada variable.

1.2.2 Datos cuantitativos

Los datos cuantitativos son los datos que brindan informaciones de cantidad, conocer cantidad de personas, cantidad de agua, entre otros criterios. Según (RODRÍGUEZ 2010), hay seis maneras de crear mapas

temáticos cuantitativos, entre ellos se encuentra:

- **Mapas de representación de datos por puntos:** Se caracterizan por la representación uniforme de puntos sobre una superficie que indican una cantidad. Estos mapas son fácilmente comprensibles y no solo muestran que existen los datos, sino también donde se concentran. En este tipo de mapas, los puntos son siempre del mismo tamaño, están representados por el mismo color y muestran datos cuantitativos que deberán estar en valor absoluto y referirse siempre a unidades territoriales. Están simbolizados generalmente por símbolos de igual tamaño.
- **Mapas de representación de datos por símbolos proporcionales:** Esta técnica sirve para representar la distribución de datos cuantitativos y 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 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.

A diferencia de los mapas de representación puntual, los símbolos proporcionales sí varían en tamaño. Esto se debe a las características del territorio que representa. Al tratarse de datos cuantitativos, pueden representar la cantidad de personas en distintos estados y el tamaño de los símbolos varía según la cifra de personas en cada localidad.

- **Mapas de representación de datos por isolíneas:** La creación y definición de las isolíneas supone un proceso costoso y laborioso, mayormente se utiliza en mapas para medir la temperatura. Una isolínea es una línea imaginaria que une puntos que tienen el mismo valor de una variable geográfica determinada. Nunca se cruzan, ni se bifurcan¹ y siempre deben tratarse cerradas.
- **Mapas de representación de datos mediante flujos:** Los mapas de flujos o mapas dinámicos muestran movimientos lineales, se emplean para representar cargas de transportes; importación y exportación de mercancías, pueden representar fenómenos cuantitativos y cualitativos. El movimiento se simboliza mediante líneas o flechas de acuerdo al contexto y unen lugares de origen y destino del movimiento. Otros ejemplos de movimiento son de corrientes marinas, migraciones, rutas migratorias, entre otras.
- **Mapas de representación de datos por coropletas:** Este tipo de mapa brindan información cuantitativa de los eventos a representar mediante tramas o gamas de colores. Con los mapas de representación de datos por coropletas se pueden representar fenómenos como la densidad de la

¹ Bifurcan: División en dos partes iguales.

población, los consumos de energía, lugares en donde los suelos estén muy afectados comparados con otros lugares, nivel de analfabetismo de la población, entre otros.

- **Mapas de representación de datos por gráficos:** Los mapas que representan información mediante gráficos, son mapas que se utilizan para representar valores cuantitativos, existen dos formas de representar los datos mediante esta técnica, estas son, las gráficas de barras y las gráficas de pastel. Mayormente se utiliza para representar comparaciones con datos obtenidos anteriormente.
- **Mapas de representación de datos mediante cartogramas:** Los cartogramas son diagramas que muestran datos cuantitativos asociados a áreas, en los que las unidades de enumeración son proporcionales al dato representado. Este método aporta información distorsionando las superficies reales, utilizando cada superficie de enumeración a modo de un símbolo proporcional que aumenta o disminuye siempre en función de los correspondientes valores. Estos mapas carecen de mapa base, ya que es la propia base geográfica la que se convierte con su distorsión en contenido temático (RODRÍGUEZ 2010).

1.3 Análisis espacial

Según (HERRERA y COCA 2015), definen el análisis espacial como uno de los aspectos más interesantes de los SIG, ya que mediante ellos se puede derivar nueva información de los datos existentes para mejorar la toma de decisiones.

Tipos de análisis espacial

Víctor Olaya en su libro “Sistemas de Información Geográfica” establece algunos tipos de análisis espacial, los cuales se mencionan a continuación (OLAYA 2014).

- **Consulta Espacial:** Es aquel que proporciona una información inmediata a partir de una simple observación de los datos.
- **Análisis Topológico:** Las consultas hechas a las capas de datos espaciales no solo pueden tener relación con la ubicación, sino también con otros elementos de la misma capa.
- **Transformaciones:** Una de las transformaciones es la creación del área de influencia. Estos convierten los distintos elementos geográficos en áreas que reflejan la influencia de estos elementos en base de parámetros como distancias o coste.
- **Análisis de Superficie:** Es una de las grandes potencialidades del SIG, aunque este análisis de superficie se entiende como el de la superficie terrestre (relieve), parte de estas operaciones pueden

aplicarse a cualquier otra superficie.

- **Estadística Descriptiva:** Aquí se denotan descriptores de centralidad y dispersión, dependencia espacial o estudio de patrones espaciales, entre otros. Pueden usarse para el contraste de hipótesis que contengan una cierta componente espacial.
- **Inferencia:** Permite inferir el comportamiento de variables y estudiar las formas en que éstas se van a desarrollar a lo largo del tiempo.
- **Toma de Decisiones y Optimización:** La estructura en que están desarrollados los Sistemas de Información Geográfica son factores claves que ayudan a combinar aspectos de diferentes sectores. Esto hace que sea un factor primordial en la toma de decisiones relativa a la actividad donde se ejerce influencia.

De esta forma se define como análisis espacial al conjunto de técnicas que ayudan a inferir y comprender los datos espaciales con el fin de brindar informaciones que sean de relevancia a un hecho específico.

1.4 Soluciones existentes

Para una mejor comprensión, estudio y desarrollo del sistema, se hace necesario realizar un análisis a soluciones que se han desarrollado y utilizan entre sus funcionalidades la creación de mapas temáticos, con el objetivo de conocer las ventajas y desventajas que presentan y de sentar las bases para tener un mayor dominio acerca del sistema a implementar.

1.4.1 Quantum GIS (QGIS)

Quantum GIS puede crear plugins personalizados y aplicaciones habilitadas para Sistemas de Información Geográfica usando Python o C++. Es una de las más conocidas y cuenta con un pequeño módulo de análisis temático que permite generar mapas de datos puntuales, símbolos proporcionales, mapas de coropletas y mapas de datos superficiales.

A continuación, se representan cuatro modos que esta herramienta utiliza para crear mapas temáticos:

Graduated Symbol: Utilizado para crear mapas temáticos de símbolos proporcionales, teniendo en cuenta los valores del campo seleccionado. El color de los objetos es mostrado según el campo seleccionado.

Single Symbol: Utilizado para asignar un mismo estilo a todos los elementos de una capa. Mayormente es utilizado para crear mapas de datos puntuales ya que localiza fenómenos de los que no se cuenta con un nivel de medida cuantitativo.

Continuos Color: Es utilizado para crear mapas temáticos de coropletas. Los objetos de la capa

seleccionada son mostrados mediante la degradación de colores.

Unique Value: Se utiliza para crear mapas de datos superficiales, aquí solo se puede mostrar fenómenos que no tengan asociados datos cuantitativos. Los objetos de la capa son clasificados por valores únicos y mostrados, teniendo cada uno un color diferente.

1.4.2 GeneSIG

Es una plataforma desarrollada por la XETID, Geocuba y la UCI para el desarrollo de Sistemas de Información Geográfica, según (PANTOJA 2010) tiene como objetivos fundamentales:

- Permitir la información geoespacial a cualquier área que lo requiera.
- Proporcionar servicios de acceso a la información geográfica para la consulta y análisis de datos, mediante una interfaz sencilla y de fácil manejo para personas no especializadas en Sistemas de Información Geográfica.
- Integrar la información socioeconómica existente (recursos humanos, activos fijos, entidades de servicios, lugares de interés, entre otras) con la información geográfica asociada.

Otro aspecto fundamental es que está desarrollada sobre herramientas y tecnologías libres tales como:

- Sistema Gestor de Base de Datos PostgreSQL
- Extensión PostGIS para el tratado de mapas en base de datos.
- MapServer como servidor de mapas.
- Apache como servidor de aplicaciones.
- Lenguaje de programación JavaScript por parte del cliente y PHP (*Hypertext Preprocessor*) por parte del servidor.
- Como framework de desarrollo Ext.js.

Esta herramienta modifica los mapas creando tematizaciones de tipo:

- Estilo Graduado: Establece un color para cada clase o rango basado en la definición de la capa.
- Estilo Categorizado: Posibilita construir los estilos de las capas estableciendo colores diferentes a los distintos valores escogidos por el usuario.
- Gráficas Dinámicas: Posibilita al usuario visualizar datos cuantitativos a través de gráficas de pastel y de barras.

- Estilo de Símbolo Proporcional: Según la selección de una capa, tematiza atendiendo a un criterio de análisis, un color y un símbolo.

1.4.3 Servicio de mapas temáticos

Esta herramienta permite generar mapas temáticos para el desarrollo de actividades. El mismo genera ocho tipos de mapas temáticos, entre ellos los mapas de datos puntuales, mapas de datos lineales, mapas de datos superficiales, mapas de puntos, mapas de coropletas, mapas de símbolos proporcionales, mapas de flujos y mapas de cartogramas. Utiliza tecnologías tales como Python, lenguaje de programación del lado del servidor y JavaScript del lado del cliente.

Según (RODRÍGUEZ 2010) el servicio de mapas temáticos produce un mapa conformado por un mapa base y una capa de contenido temático obtenida a partir de informaciones estadísticas. Como resultado se obtiene una imagen apropiada para mostrarse en un ordenador y la leyenda asociada al mapa, estos elementos se transfieren utilizando el protocolo de transferencia de hipertexto (HTTP por sus siglas en inglés).

1.4.4 Módulo de análisis estadístico descriptivo para el SIG_UCI v2.0

Este módulo permite visualizar mediante tematizaciones el área de residencia de la universidad. Cuenta con funcionalidades que permiten representar como está distribuido cierto indicador mediante parámetros asociados a la estadística descriptiva (HERRERA y COCA 2015). Se encuentra desarrollado bajo herramientas y tecnologías tales como:

- Symfony como framework de desarrollo.
- Gestor de bases de datos PostgreSQL.
- Extensión PostGIS para el tratado de mapas en base de datos.
- Servidor de aplicaciones Apache.
- Librería OpenLayers.
- Entorno de desarrollo NetBeans.

El módulo permite generar mapas temáticos de estilo proporcional, estilo graduado, estilo categorizado, y estilo gráfico.

1.4.5 UDig

Este software está basado en los parámetros del OGC (*Open Geospatial Consortium*) que regula los estándares en Sistemas de Información Geográfica. Está programado en lenguaje de programación Java,

mediante un entorno de desarrollo llamado Eclipse. Mediante UDig se puede tener acceso a través de Internet a objetos geográficos no sólo para efectos de visualización en un mapa, sino además para su consulta (UDig 2010). Permite la creación de un solo tipo de mapa temático, los mapas de coropletas, y al igual que QGis no es posible accederlo desde la web. Los datos que requiere para generar mapas son la capa y el campo seleccionado para así evaluar el fenómeno. Como principales desventajas del UDig se tiene:

- No está disponible para trabajar desde la web y esto lo limita de forma considerada.
- Comparado con otras soluciones, esta herramienta solo puede obtener los mapas de coropletas.

1.4.6 Características del SIG_UCI v3.0

El Sistema de información Geográfica para la Universidad de las Ciencias Informáticas (SIG_UCI v3.0) permite la consulta, análisis, gestión, y visualización de la información relacionada con la organización en general. Esto significa una mejora considerable en el proceso de toma de decisiones de las diferentes áreas que forman parte de la estructura de la universidad. El sistema está compuesto por los siguientes módulos:

- **Módulo de Seguridad:** Permite administrar la seguridad y el acceso al sistema a través de usuarios, contraseñas y privilegios por roles.
- **Módulo de Navegación:** Permite acercar, alejar y mover el mapa. Además, permite navegar por el mapa de referencia.
- **Módulo de Identificación:** Permite realizar identificación puntual de la información que se muestra.
- **Módulo de Localización:** Permite localizar los objetos geográficos y las entidades asociadas a la información que maneja el SIG.
- **Módulo de Análisis:** Permite medir distancia, calcular área y perímetro.
- **Módulo de Administración:** Permite la gestión de grupos de capas, y la geo-referenciación de nuevos lugares de la institución.
- **Módulo de Visualización:** Permite habilitar y deshabilitar capas.

Las principales tecnologías en que está desarrollado este sistema son:

- Framework de desarrollo Angularjs por parte del cliente y Express en la parte del servidor, utilizando Nodejs como servidor web.
- Gestor de base de datos PostgreSQL con la extensión PostGIS.

- Librería OpenLayers para el tratado de mapas en la web y crear diferentes capas.
- Lenguaje de programación JavaScript desde el cliente hasta el servidor.

A partir del análisis realizado sobre las soluciones existentes, los autores concluyen lo siguiente:

- QGis y UDig son soluciones de gran provecho, pero ambas son herramientas de escritorio, esto hace que no sean accesible desde la web, por tanto, no pueden ser integradas al módulo.
- El Servicio de mapas temáticos permite generar varios tipos de mapas estableciendo la fuente de datos y el mapa base, sin embargo, muchas de estas tematizaciones no son necesario utilizarlas, ejemplo los mapas de datos superficiales y lineales ya que la capa que se va a manejar no representa ni suelos ni carreteras, por otra parte, las tecnologías que utiliza no pueden ser integradas a este módulo.
- El módulo de análisis estadístico descriptivo y GeneSIG son herramientas que poseen tecnologías diferentes al SIG_UCI v3.0, por tanto, no pueden ser integradas. Sin embargo, se toman en cuenta las funcionalidades que brindan estas dos soluciones asociadas a los tipos de mapas temáticos que implementan, tales como los mapas de estilo gráfico, estilo categorizado y estilo proporcional.

1.5 Metodología y Tecnologías de desarrollo

En el desarrollo de un software, constituyen elementos importantes, la metodología y tecnologías a utilizar. La metodología rige el curso de vida del producto y las tecnologías se utilizan para la construcción del software. A continuación, se describe el entorno para la construcción del módulo de análisis temático. Es válido aclarar que la solución que se propone será integrada al SIG_UCI v3.0, y por tanto sigue la arquitectura que este propone.

1.5.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software son marcos de trabajo utilizados para estructurar, planificar y controlar el proceso de desarrollo del mismo. Éstas brindan una serie de técnicas, procedimientos, herramientas y soporte documental para desarrollar un producto. En la actualidad existen varias metodologías de software, éstas se clasifican en metodologías ágiles y en metodologías tradicionales.

Actualmente el proceso de desarrollo de software dentro de la actividad productiva de la UCI, se caracteriza por el uso de varias metodologías de desarrollo, desde las metodologías ágiles hasta las tradicionales o robustas como también se les conoce. Entre las metodologías con que trabaja la universidad se encuentran XP, Open Up, RUP, KINBALL, Scrum, Nova Open Up (SÁNCHEZ 2015).

Proceso Unificado Ágil (AUP_UCI).

Esta metodología de desarrollo es una versión simplificada del Proceso de Unificado Racional (RUP por sus siglas en inglés). AUP describe la forma de desarrollar aplicaciones en un software usando métodos y conceptos que se mantienen en RUP. AUP establece cuatro técnicas ágiles, entre ellas se encuentran Desarrollo Dirigido por Pruebas (*Test Driven Development* – TDD en inglés), Modelado Ágil, Gestión de Cambios Ágil, y la Refactorización de bases de datos para mejorar la productividad.

Según (SÁNCHEZ 2015), AUP establece las siguientes cuatro fases para el desarrollo del software, las mismas transcurren de manera consecutiva como muestra la **Figura 1**.

- **Inicio:** En esta fase se llega a una comprensión común cliente - equipo de desarrollo y se definen una o varias arquitecturas para el sistema.
- **Elaboración:** Tiene como objetivo que el equipo de desarrollo precise acerca de los requisitos del sistema y validar la arquitectura.
- **Construcción:** En esta fase, el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** El sistema es sometido a pruebas de validación y aceptación, y finalmente se despliega.

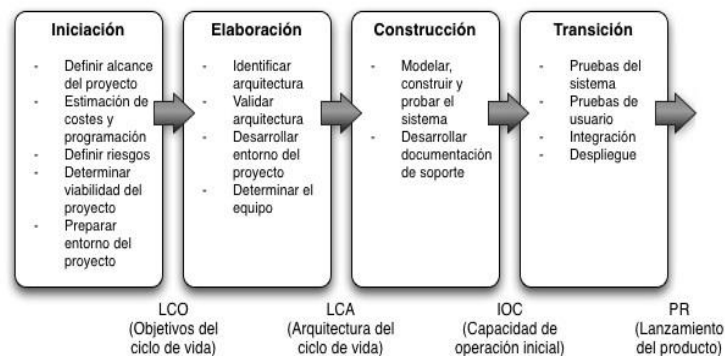


Figura 1: Fases de la metodología AUP (TORRECILLA 2012).

En la versión AUP_UCI se establecen tres fases, se mantiene la fase de inicio donde se llevan a cabo las actividades relacionadas con la planeación del proyecto, se realizan las estimaciones de esfuerzo, tiempo y costo y se decide si se ejecuta o no el proyecto. En cuanto a las fases de elaboración, construcción, y transición se unifican en una llamada fase de ejecución, donde se ejecutan las actividades requeridas para el desarrollo del software, se definen los requisitos del sistema, se elabora la arquitectura y diseño del producto, se implementa y se libera el software. Por último, en la fase de cierre se analizan los resultados

del proyecto y su ejecución (SÁNCHEZ 2015).

AUP_UCI a la vez cuenta con 7 disciplinas entre ellas se encuentran Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación (SÁNCHEZ 2015), estas se describen a continuación:

- **Modelado de negocio:** Es la disciplina destinada a comprender los procesos de negocio de una organización.
- **Requisitos:** Comprende la administración y gestión de los requisitos funcionales y no funcionales del producto, en ella hay tres formas de encapsular los requisitos, Casos de uso del sistema (CUS), Historias de usuarios(HU) y Descripción de requisitos por proceso (DRP).
- **Análisis y diseño:** Se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluidos los requisitos no funcionales.
- **Implementación:** A partir de los resultados del análisis y diseño se comienza a construir el módulo.
- **Pruebas internas:** Se verifica el resultado de la implementación probando cada construcción.
- **Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa.
- **Pruebas de aceptación:** Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software esté listo y pueda ser utilizado por los usuarios finales.

Para la construcción del módulo de análisis temáticos se hizo uso de las disciplinas Requisitos, Análisis y diseño, Implementación y Pruebas internas, utilizando el escenario número cuatro de historias de usuario.

Escenario No 4: Historias de usuarios.

Este escenario se aplica a los proyectos que hayan evaluado el objetivo a informatizar y como resultado se tiene un negocio bien definido. El cliente debe estar siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se utiliza en proyectos no muy extensos ya que una historia de usuario no debe poseer demasiada información (SÁNCHEZ 2015).

1.5.2 Lenguaje de Modelado Unificado (UML) v2.1

El lenguaje de Modelado Unificado (UML por sus siglas en inglés) es un estándar diseñado para visualizar, especificar, construir, y documentar software orientado a objetos. Este lenguaje es esencial en la construcción de software para comunicar la estructura de un sistema complejo, especificar el

comportamiento deseado del sistema, comprender mejor lo que se está construyendo, encontrar oportunidades de simplificación y reutilización (BOOCH y RUMBAUGH 2005).

Como ventaja de este lenguaje se tiene que:

- Se puede utilizar para diferentes tipos de sistemas.
- Tecnología orientada a objetos.
- Mejor soporte a la planeación y al control de proyectos.
- Fácil actualización o modificación del software a programar.

Con este lenguaje de modelado se construyeron los artefactos necesarios para llevar a cabo la construcción del módulo.

1.5.3 Visual Paradigm v8.0

Visual Paradigm para UML es una plataforma de modelado diseñada para apoyar a los arquitectos de sistemas, desarrolladores y diseñadores UML, para acelerar el proceso de análisis y diseño de aplicaciones empresariales complejas (SOFTLAND 2017).

Todos los productos de Visual Paradigm están desarrollados y diseñados para eliminar la complejidad, mejorar la productividad y comprimir el tiempo de desarrollo de software en los plazos dados con los clientes (VISUAL PARADIGM 2010).

Entre sus principales funcionalidades se tiene:

- Capacidades de ingenierías directas e inversas.
- El uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Con esta herramienta de modelado se diseñaron los diagramas de clases, modelo de datos y los prototipos de interfaz de usuario para una mejor comprensión de la solución. Además, es útil para mostrarles ideas a clientes o para poner en orden las ideas a la hora de comenzar una aplicación.

1.5.4 Lenguaje de programación JavaScript v1.8

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario (EGUÍLUZ 2013).

Entre las principales ventajas que presenta este lenguaje se tiene:

- Es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente. Si un usuario omite escribir su nombre o algún otro campo en un formulario, una función de validación en JavaScript puede desplegar en pantalla un mensaje para hacerle saber al usuario acerca de la omisión.
- Crea efectos dinámicos tales como imágenes dinámicas y presentaciones de diapositivas.

1.5.5 Entorno de Desarrollo Integrado WebStorm v2016.1

Es un IDE (*Integrate Develop Enviroment*) para JavaScript que facilita el desarrollo en dicho lenguaje. Muestra para cada objeto sus posibles métodos, así como sus propiedades. Tiene un editor HTML (*Hypertext Markup Language*) que muestra el CSS (*Cascade Style Sheet*) asociado a la etiqueta de la página que se esté editando (CASADO 2013). También es un IDE que permite el desarrollo en JavaScript por la parte del cliente y también por la parte del servidor utilizando Nodejs.

1.5.6 Servidor web Nodejs v4.4.3

Nodejs v4.4.3 es un entorno de programación en la capa del servidor con entrada y salida de datos en una arquitectura orientada a eventos. Surgió con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo servidores web. Además de las posibilidades de desarrollo modular y escalable que ofrece, hay que destacar el número de librerías (módulos) que actualmente existen y se encuentran bajo supervisión de la comunidad Open Source. El uso de estas librerías, en la mayoría de los casos, es libre y se encuentran disponibles a través del gestor de paquete NPM (*Node Package Manager* por sus siglas en inglés) que incluye Nodejs (SANTAMARÍA 2015).

1.5.7 ExpressJS v4.0.0

Es un framework en la capa del servidor basado en el lenguaje de programación JavaScript. Express.js se utiliza para el desarrollo de aplicaciones web minimalista y flexible para Node.js. Está inspirado en Sinatra, además es robusto, rápido, flexible y muy simple. Entre otras características, ofrece router de URL (*Uniform Resource Locator*) (Get, Post, Put) (BALDOQUÍN y VALLÍN 2015).

1.5.8 AngularJS v1.5.6

Es un framework estructural que permite generar las bases fundamentales de la aplicación, es creado por google, lo que garantiza que es un software que está en constante evolución, está basado en JavaScript y tiene como objetivo crear aplicaciones web dinámicas. La base fundamental de este framework es el patrón modelo - vista - controlador. Permite crear inyecciones de dependencias, esto hace que se pueda utilizar

una aplicación de manera modular y permitir el acceso a nuevos servicios. Entre sus principales características se tiene:

- No genera componentes gráficos o CSS ya que se va a tener total libertad para crear la aplicación en los términos que se desee y no se va a tener que estar aferrado a una línea gráfica específica.
- Es liviano y eficiente ya que el código que genera es optimizado y se puede utilizar para trabajar en dispositivos móviles.
- Puede coexistir con otros frameworks.

1.5.9 Librería OpenLayers v3.14

Es una biblioteca de mapeo para clientes web y móviles, utiliza lenguaje de programación JavaScript del lado del cliente y permite crear un mapa en cualquier página web, así como añadir capas en formato ráster y vectorial, dar estilos a los elementos de un mapa, mostrar mosaicos de mapas, entre otras cosas. Es desarrollado para fomentar el uso de información geográfica de todo tipo. OpenLayers 3.14 es un rediseño de la biblioteca de mapas web de OpenLayers.

Esta librería permite observar el mapa desde diferentes dimensiones, es utilizada para crear capas sobre el mapa de la universidad, así como para realizar las tematizaciones mediante las figuras que éste genera.

1.5.10 MapServer v6.4

MapServer 6.4 es un entorno de desarrollo o plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas en Internet/Intranet. Permite visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer. Ofrece la posibilidad de ser utilizado como servidor de mapas de terceros programas y admite múltiples formatos de datos vectoriales, características que hacen de MapServer una herramienta potente (SPHINX 2017).

1.5.11 Sistema Gestor de Base de Datos PostgreSQL v9.3

PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) que posee varias características significativas entre las que se pueden incluir las subconsultas, además se puede usar, modificar y distribuir de forma gratuita debido a la liberación de su licencia. Según (QUIÑONES 2017) como ventajas se tiene:

- Puede estar funcionando varios años y no presentar fallos en su servicio.
- Es diseñado y creado para tener un mantenimiento mucho menor que otros productos.

1.5.12 Extensión PostGIS v2.1

Es una extensión del sistema de bases de datos objeto - relacional PostgreSQL, añade soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL. PostGIS en el año 2006 fue certificado por el OGC lo que permite la interoperabilidad con otros sistemas. Tiene varias características entre ellas se destaca:

- El alto rendimiento.
- Presenta una línea de comandos y herramientas gráficas para la gestión flexible.
- Ofrece representaciones espaciales de los tipos de geometrías (líneas, puntos, polígonos).

En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL (RAMSEY 2017).

1.5.13 Sequelize v3.0.0

Es un ORM (*Object Relational Mapping*) para Nodejs, soporta los dialectos para PostgreSQL, MySQL, SQLite y MSSQL, es fácil y flexible de utilizar al mapear las tablas en objetos relacionales. Este ORM es de gran utilidad para el manejo de información que se necesite manipular del lado del servidor y del lado del cliente. Para realizar consultas, inserciones, eliminaciones, no es necesario realizar grandes *queries* con sentencias anidadas complicadas, con Sequelize existen funciones que facilitan el trabajo para ingresar parámetros que indiquen a una función determinada, el tipo de consulta a realizar, así como otras operaciones (KEVANORT 2015).

Este ORM se utilizó para crear el modelo correspondiente a la base de datos del módulo, donde se van a guardar los datos de las tematizaciones generadas.

1.6 Conclusiones parciales

Por las características expuestas en este capítulo se arribó a las siguientes conclusiones:

- El análisis del marco teórico de la investigación permitió obtener un conocimiento acerca de los distintos mapas temáticos existentes y su representación en SIG.
- Mediante la problemática planteada surgió la necesidad de crear un módulo de análisis temático que satisfaga las necesidades de los directivos que laboran en el área de residencia de la universidad.
- A partir del análisis realizado a las soluciones existentes, se identificaron funcionalidades útiles para el desarrollo de la solución.

- Debido a que este módulo será integrado al SIG_UCI v3.0, se decidió optar por las mismas tecnologías y metodología en que este sistema está desarrollado.

CAPÍTULO 2: Propuesta de Solución

Para la construcción de un software es necesario que se tenga un dominio acerca de lo que se quiere desarrollar. En el presente capítulo se describe la solución propuesta siguiendo la metodología AUP_UCI, tomando como escenario las historias de usuarios. Se describen los requisitos funcionales y no funcionales del sistema. Además, se presenta la arquitectura a utilizar y los patrones de diseño empleados en el módulo, se confeccionan los diagramas correspondientes al diseño de la solución y al modelo de la base de datos, así como la descripción de sus atributos.

2.1 Especificación de los requerimientos del sistema

Para la construcción de un buen producto de software hay que tener en cuenta uno de los pasos más importantes para el desarrollo del proceso, como es la obtención de los requisitos del sistema o las exigencias por parte del cliente como también se le conoce. Conocer los requisitos ayuda a los ingenieros a comprender el problema y así construir una solución que satisfaga las necesidades del interesado.

Para obtener los requisitos del sistema existen varias técnicas que posibilitan su obtención segura y eficiente, entre ellas se encuentran la encuesta, listas de verificación, tormenta de ideas, entrevista, cuestionarios y el análisis de la documentación. Para el levantamiento de los requisitos se utilizó la entrevista, las cuales se realizaron a dos especialistas de la Dirección de Informatización con el objetivo de obtener información sobre la problemática existente y definir los requisitos funcionales para el desarrollo del módulo, para ver las preguntas realizadas remitirse al **Anexo 1**.

Según (BAÑOS 2012) la entrevista es una técnica muy aceptada dentro de la ingeniería de requisitos y tiene un uso ampliamente extendido. Permite al analista tomar conocimiento del problema, así como comprender los objetivos de la solución buscada.

2.1.1 Requisitos funcionales

Los requisitos funcionales refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Los requisitos funcionales se escriben generalmente de forma que se entiendan por los usuarios del sistema. Estos varían desde requisitos generales que cubren lo que el sistema debe hacer, hasta requisitos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes en una organización (SOMMERVILLE 2011).

A continuación, se describen los requisitos funcionales del sistema.

RF1: Realizar tematización según estilo categorizado.

Esta funcionalidad permite al cliente obtener un mapa temático según el estilo categorizado y muestra la capa tematizada en el mapa.

RF2: Realizar tematización según el gráfico de barras.

Esta funcionalidad permite al cliente realizar una tematización mediante el gráfico de barras, para ello es necesario tener en cuenta los colores de las gráficas y la capa sobre la cual se va a tematizar.

RF3: Realizar tematización según el gráfico de pastel.

Esta funcionalidad permite al cliente realizar una tematización mediante el gráfico de pastel, para ello es necesario tener en cuenta los colores a representar y la capa sobre la cual se va a tematizar.

RF4: Realizar tematización según el estilo proporcional.

Esta funcionalidad le permite al cliente realizar una tematización según el estilo proporcional, para ello se tiene en cuenta el tamaño de los símbolos (pequeños círculos que representan un estado del objetivo a tematizar) y la capa sobre la cual se va a tematizar.

RF5: Listar tematizaciones.

Esta funcionalidad le permite al cliente listar las tematizaciones que haya realizado.

RF6: Visualizar tematización.

Esta funcionalidad le permite al cliente visualizar una tematización una vez que esta haya sido añadida a la lista.

RF7: Eliminar tematización.

Esta funcionalidad le permite al cliente eliminar una tematización una vez que esta haya sido añadida a la lista.

RF8: Modificar tematización.

Esta funcionalidad permite modificar una tematización una vez que haya sido creada.

El sistema debe permitir que se puedan crear mapas temáticos por colores, mediante símbolos proporcionales y gráficas, atendiendo a los diferentes estilos y mediante los criterios seleccionados por el usuario. Los criterios son los siguientes:

Tipo de tematización: Estilo proporcional.

Criterios

- Cantidad de residentes por edificio en la residencia uno.
- Cantidad de residentes por edificio en la residencia dos.
- Cantidad de residentes por edificio en la residencia de profesores y especialistas.

Tipo de tematización: Estilo categorizado.

Criterios

- Edificios por el tipo de residencia.
- Edificios libres y ocupados en la residencia uno.
- Edificios libres y ocupados en la residencia dos.
- Edificios libres y ocupados en la residencia de profesores y especialistas.

Tipo de tematización: Gráfico de barras o Gráfico de pastel.

Criterios

- Edificios según su capacidad máxima y capacidad ocupada en la residencia uno.
- Edificios según su capacidad máxima y capacidad ocupada en la residencia dos.
- Edificios según su capacidad máxima y capacidad ocupada en la residencia de trabajadores y especialistas.

2.1.2 Requisitos no funcionales (RnF)

Según (SOMMERVILLE 2011) los requisitos no funcionales son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tipo de respuestas y uso de almacenamiento.

RnF1 Apariencia o interfaz externa.

La interfaz externa del producto debe ser de fácil navegación por el usuario, sencilla y legible. Se debe garantizar la uniformidad en las interfaces de usuario. Los conceptos y términos que se plasman en las interfaces del sistema fueron obtenidos de la experiencia de las personas que más utilizarán este producto.

RnF2 Rendimiento.

El tiempo de respuesta ante diferentes peticiones de los usuarios será inferior a los tres (3) segundos.

RnF3 Usabilidad.

El sistema puede ser utilizado por cualquier usuario que posea conocimientos informáticos básicos. El software tiene una curva de aprendizaje baja, que permite al usuario familiarizarse rápidamente con los elementos del sistema y operarlo de forma correcta en poco tiempo de uso.

Requisitos de Hardware y software.

RnF4 Software:

Para las Terminales Cliente:

- Navegador: Mozilla Firefox (v36) o superior, Google Chrome (v50) o superior, Opera (24) o superior.
- Sistema operativo: GNU/Linux Ubuntu 14.04, Windows y Mac OS.

Para la Terminal Servidor:

- Sistema operativo: GNU/Linux Ubuntu Server 14.04.
- Servidor Web: Node JS v4.4.3
- Sistema Gestor de Base de Datos: PostgreSQL 9.3.
- PostGIS v2.1 como extensión de PostgreSQL para el manejo de datos espaciales.
- MapServer 6.4.

RnF5 Hardware

Para las Terminales Cliente:

- Tarjeta de red.
- RAM: Mínimo 512 MB. Recomendado 1GB.
- Procesador: Mínimo P4 a 1.0 GHz. Recomendado Dual Core a 1.3 GHz.

Para la Terminal Servidor:

- Tarjeta de red.
- RAM: Mínimo 2 GB. Recomendado 4 GB.
- Procesador: Mínimo Core 2 Duo a 1.7 GHz. Recomendado Core i3 a 1.3 GHz.
- Espacio en Disco: Mínimo 40 GB. Recomendado 100 GB.

2.2 Descripción de la solución propuesta

Dicho antes que este módulo es creado con el objetivo de satisfacer las necesidades de los directivos que laboran en el área de residencia de la universidad, ayudándolos en el proceso de toma de decisiones, una vez que el usuario acceda al menú de administración y presione el botón “tematizaciones”, este direccionará a una vista HTML donde tendrá una tabla con todas las tematizaciones definidas para la residencia mediante un criterio. Cuando el usuario elija la acción visualizar, el sistema mostrará la tematización definida mediante el criterio seleccionado y una vez que el usuario quiera eliminar dicha tematización presionará la acción eliminar.

Si el usuario desea añadir una tematización, presionará el botón añadir y elegirá el estilo de tematización que quiere realizar ya sea estilo proporcional, estilo categorizado, estilo gráfico de barras o estilo gráfico de pastel, una vez elegido el tipo de tematización, procederá a introducir los datos eligiendo la capa que desea tematizar y mediante qué criterio quiere tematizar. Una vez realizada la tematización, si el usuario desea modificarla, ya sea para cambiar el color o criterio, presionará la acción modificar y direccionará a la interfaz correspondiente para actualizar los datos.

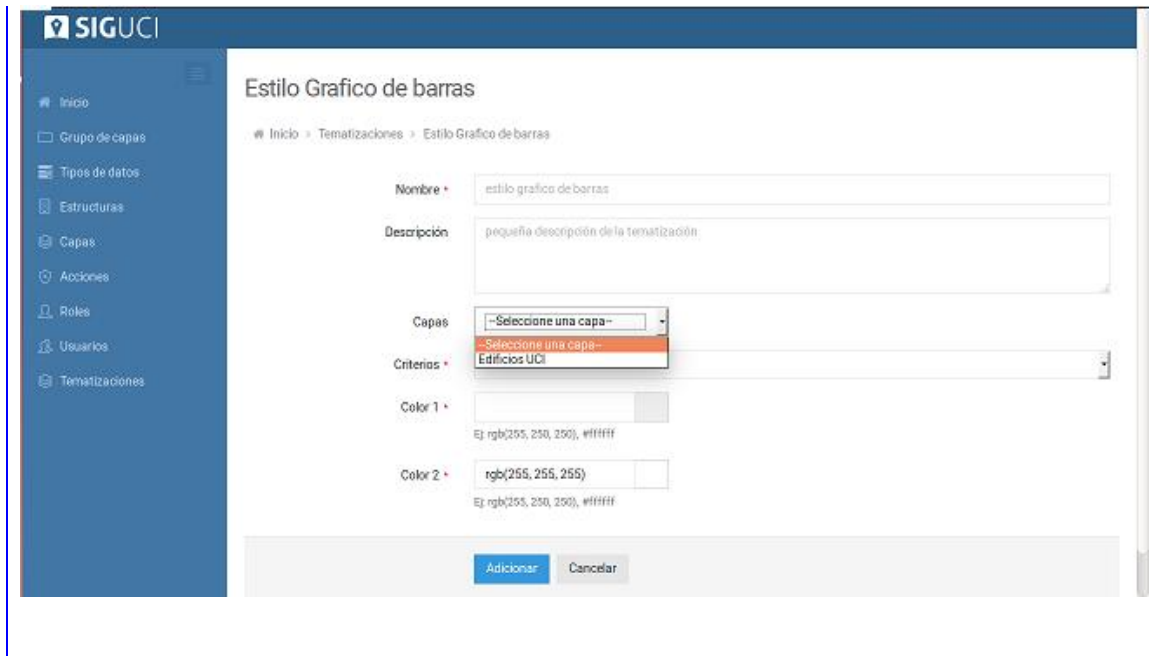
Historia de usuario.

Las historias de usuario son técnicas utilizadas para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en una semana (ISSI 2003). La **Tabla 1** representa la historia de usuario del requisito “Realizar tematización según estilo gráfico de barras.”

Tabla 1: Historia de usuario del requisito “Realizar tematización según estilo gráfico de barras”.

Número: 1	Nombre del requisito: Realizar tematización según estilo gráfico de barras
Programador: Asley Arias Sainz y Luis David González Sarduy	Iteración Asignada: 1 era
Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> Rotura de host donde esté instalado el servidor de Bases de Datos. 	Tiempo Real: 1 semana

<ul style="list-style-type: none">• Problemas eléctricos.• Atrasos generados por el cliente debido a la poca disponibilidad de los mismos.	
<p>Descripción:</p> <p>1- Objetivo:</p> <p>-Permite al usuario insertar una tematización según el estilo gráfico de barras a un área determinada.</p> <p>2- Acciones para lograr el objetivo:</p> <p>Para insertar una tematización hay que:</p> <ul style="list-style-type: none">- Tener en cuenta el tipo de tematización que se desea realizar, (en este caso estilo gráfico de barras), y se deben de llenar los datos que brinda la interfaz dígase capa, criterio, descripción, color y nombre- Debe estar autenticado en el sistema. <p>3- Comportamientos válidos y no válidos:</p> <p>Los campos capa, criterios, descripción, color y nombre son obligatorios:</p> <ul style="list-style-type: none">-Nombre: Se inserta un nombre a la tematización.- Descripción: Campo de texto para describir la tematización.- Capa: botón para seleccionar la capa a tematizar (Edificios UCI).- Criterios: botón para seleccionar el criterio por el cual se desea tematizar.-Color: Se inserta un color a la tematización.	
Observaciones:	
Prototipo de interfaz:	



Las demás descripciones de historias de usuarios se pueden encontrar en el **Anexo 2**

2.3 Análisis y diseño

Análisis y diseño es una disciplina de la metodología de desarrollo de software AUP_UCI. En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de las funcionalidades, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales (SÁNCHEZ 2015).

Diseño.

“El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. Los principios de diseño establecen una filosofía general que guía el trabajo de diseño que debe ejecutarse” (PRESSMAN 2010).

2.3.1 Arquitectura de software

La arquitectura de software de un programa es la estructura o las estructuras del sistema, es lo que comprende a todos los componentes del software, las propiedades externas y visibles y las relaciones que existen entre ellos (PRESSMAN 2010).

2.3.2 Patrón arquitectónico Modelo Vista Controlador

El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (FERNÁNDEZ y DÍAZ 2012).

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo (UNIVERSIDAD DE ALICANTE 2017).

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo (FERNÁNDEZ y DÍAZ 2012).

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo (FERNÁNDEZ y DÍAZ 2012).

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo (FERNÁNDEZ y DÍAZ 2012).

A continuación, se representa en la **Figura 2** como está estructurado este patrón.

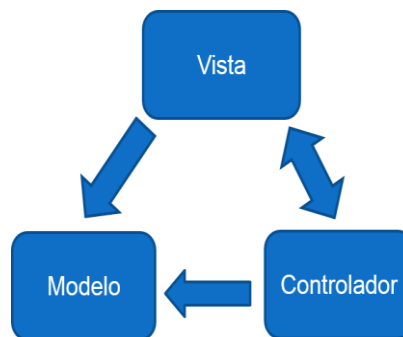


Figura 2: Patrón Modelo Vista Controlador.

2.3.3 Patrones de diseño

Según (PRESSMAN 2010) *“Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del diseño dentro de un contexto específico y entre fuerzas que afectan la manera en la que se aplica y en la que se utiliza dicho patrón”*.

Patrones GRASP.

Los patrones GRASP² describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (LARMAN 1999). Los patrones identificados son:

Alta Cohesión: Asigna a las clases responsabilidades que trabajan sobre una misma área de la aplicación y que no posea mucha responsabilidad, evitando que una clase sea responsable de muchas tareas funcionalmente heterogéneas.

En este caso, se garantiza que cada una de las clases del sistema posean alta cohesión, de manera que las clases posean la característica de tener las responsabilidades estrechamente relacionadas y que no realicen un trabajo enorme. El uso de este patrón permite que se pueda mejorar la claridad y facilidad en que se entiende el diseño, se simplifique el mantenimiento y existan mejoras de funcionalidad.

Bajo Acoplamiento: Se asignan las responsabilidades de forma tal que cada clase comunique con el menor número de clases, minimizando el nivel de dependencia.

El acoplamiento es una medida de la fuerza en que una clase está conectada a otras clases, que las conoce y recurre a ellas. En este caso, se refleja el bajo acoplamiento, en cada una de las clases del sistema con el objetivo de que una clase no dependa de muchas clases, de esta forma, no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar.

Controlador: Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación, en este caso, se encuentra reflejado en la clase *AdminTematizacionesController* **Figura 3** que se encarga de controlar las peticiones realizadas desde la interfaz.

² GRASP: acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades).

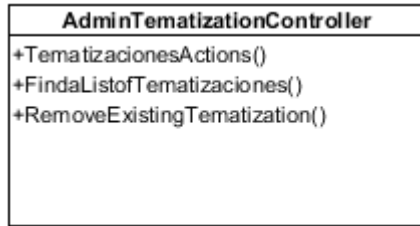


Figura 3: Patrón Controlador.

Experto: Este patrón es evidenciado en las entidades del modelo ya que poseen un grupo de funcionalidades que están estrechamente relacionadas con la entidad que representan y contienen la información necesaria de la tabla que representan. La **Figura 4** representa la creación del modelo de datos

```
var Tematization = sequelize.define('tematization', {
  id_tematization: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    allowNull: false,
    autoIncrement: true,
    unique: true
  },
  tematization_name: {
    type: DataTypes.STRING,
    allowNull: false,
    validate: {
```

Figura 4: Patrón Experto.

Creador: En la clase *MapManager* se evidencia este patrón, ya que aquí se evidencian acciones definidas para el sistema. En dichas acciones se crean objetos, lo que evidencia que la clase *MapManager* es creador. En la **Figura 5** se representa un método de la clase *MapManager*.

```
this.mapManager = new ol.Map({
  layers: [base],
  target: 'map',
  interactions: new ol.interaction.defaults({
    doubleClickZoom: false,
    altShiftDragRotate: false
  }),
  controls: ol.control.defaults({
    attribution: false,
    zoom: false,
    rotate: false
  })
});
```

Figura 5: Patrón Creador.

Patrones GOF.

Los patrones de diseño GOF³ se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento, los patrones **creacionales** están relacionados con los procesos de creación de objetos, los **estructurales** definen estructuras típicas entre clases y objetos, mientras que los patrones de **comportamiento** caracterizan la forma en que los objetos se distribuyen responsabilidades. Los patrones identificados son:

Inyección de dependencias: Este patrón es utilizado ya que el framework Angularjs hace uso de él. Por tanto, las dependencias como los servicios son suministradas por Angularjs. Por ello al crear un componente de Angularjs, se hace necesario que se especifiquen las dependencias que se necesitan. Ejemplo, en la clase controlador del módulo se utilizan servicios que brinda la universidad, esto hace que al crear el controlador debemos especificar las dependencias del servicio y no intentar crear un objeto del servicio. La **Figura 6** muestra un ejemplo de este patrón.

```
ApplicationConfiguration.getAdminModule('tematizaciones')
  .controller('AdminTematizacionesController', ['$scope', '$rootScope', '$q', '$compile', '$filter',
    'DOptionsBuilder', 'DTColumnBuilder', '$stateParams', '$location',
    'Authentication', 'Tematizaciones', 'TematizationTypes', 'Layers', 'MapVisual', 'Services',
    function ($scope, $rootScope, $q, $compile, $filter, $resource,
      DOptionsBuilder, DTColumnBuilder, $stateParams, $location,
      Authentication, Tematizaciones, TematizationTypes, Layers, MapVisual, Services,
```

Figura 6: Patrón inyección de dependencias.

2.3.4 Diagrama de clases del diseño

Un diagrama de clases es empleado para representar y documentar el diseño. Además, sirve para visualizar las clases existentes en el sistema y las relaciones que existen entre ellas. Un diagrama de clases está compuesto por los siguientes elementos: Clases (atributos, métodos y visibilidad) y Relaciones (Herencia, Composición, Agregación, Asociación y Uso). A continuación la **Figura 7** muestra el diagrama de clases del diseño del módulo.

³ GOF: acrónimo que significa Gang of Four, también conocidos como Pandilla de los Cuatro.

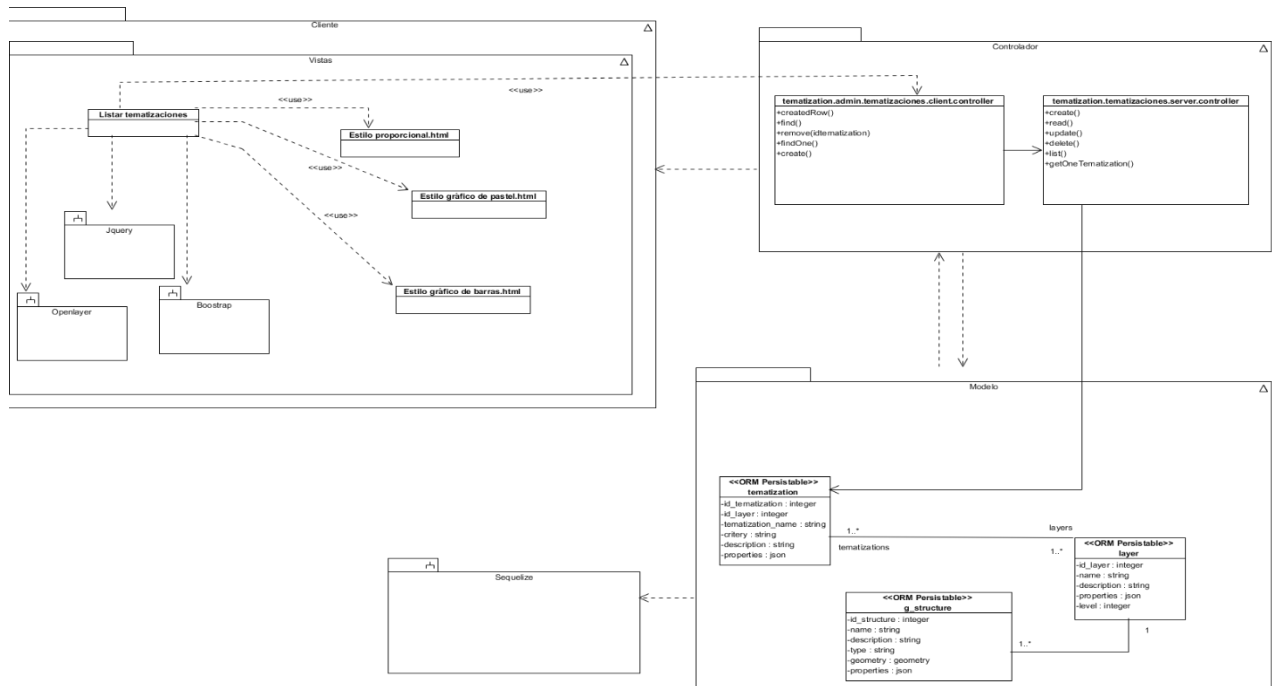


Figura 7: Diagrama de clases del diseño correspondiente al paquete Tematizaciones.

Descripción de las clases del diagrama de clases del diseño.

OpenLayers: Librería que permite cargar los mapas en la web y visualizar las capas tematizadas sobre él.

CP_tematizaciones: Interfaz que brinda los diferentes formularios para añadir las tematizaciones

Tematization.admin.tematizaciones.client.controller: Controlador por el cual se procesan todas las peticiones del usuario.

Tematization.tematizaciones.server.controller: Controlador por la parte del servidor que se encarga de enviar los datos al modelo.

Tematization: Clase del modelo de datos que representa los atributos de cada tematización.

Layer: Clase del modelo de datos que representa los atributos de cada capa.

g_structure: Clase del modelo de datos que representa los atributos de cada estructura.

Estilo_proporcional: Formulario que captura los datos para realizar una tematización según el estilo proporcional.

Estilo_grafico_pastel: Formulario que captura los datos para realizar una tematización según el estilo gráfico de pastel.

Estilo_grafico_barras: Formulario que captura los datos para realizar una tematización según el estilo

gráfico de barras.

2.3.5 Diseño de la base de datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, entre otras (COSTAL 2015).

El modelo entidad - relación es uno de los enfoques de modelación de datos que más se utiliza actualmente por su simplicidad y legibilidad. Su legibilidad se ve favorecida porque proporciona una notación diagramática muy comprensiva. Es una herramienta útil tanto para ayudar al diseñador a reflejar en un modelo conceptual los requisitos del mundo real de interés como para comunicarse con el usuario final sobre el modelo conceptual obtenido y, de este modo, poder verificar si satisface sus requisitos (COSTAL 2015).

El modelo entidad - relación del módulo de análisis temático del SIG_UCI v3.0 cuenta con tres tablas una de ellas es **tematization** donde se encuentran los atributos definidos para cada tipo de tematización, cuenta también con la tabla **layer** que esta va a tener una relación con cada tematización que se va a desarrollar ya que la capa que se pretende tematizar es la residencia, y la tabla **g_structure** donde se va a obtener la geometría de los objetos a tematizar. A continuación, la **Figura 8** representa el modelo de datos del módulo.

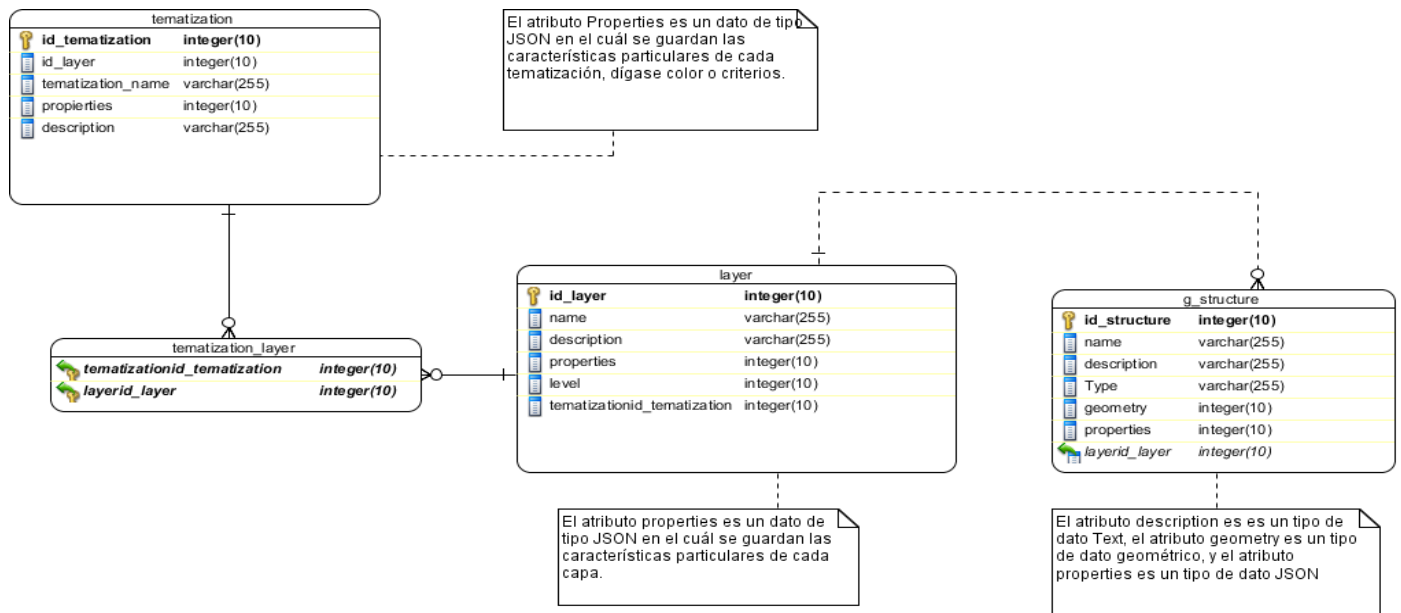


Figura 8: Modelo Entidad Relación.

Descripción de los atributos de la tabla de base de datos.

En la **Tabla 2** se representan los atributos de la tabla *tematization* del módulo.

Tabla 2: Descripción de los atributos de la tabla *tematization*.

Atributo	Tipo de dato	Descripción
Id_tematization	Integer	Etiqueta que identifica la Tematización.
Id_layer	Integer	Etiqueta única que identifica la capa sobre la cual se quiere tematizar (Edificios UCI).
Tematization_name	Varchar	Almacena el nombre de las tematizaciones.
Properties	Json	Almacena las características particulares de cada tematización.
Description	varchar	Almacena una descripción definida por el usuario para cada tipo de tematización.

2.4 Conclusiones parciales

Para el desarrollo del presente módulo, fue necesario realizar un conjunto de operaciones por parte de los desarrolladores.

- Con la entrevista realizada se identificaron los requisitos no funcionales y funcionales del producto, estos últimos fueron encapsulados en historias de usuario.
- Con la arquitectura especificada se identificaron los elementos que producen un impacto en la estructura del sistema y así reducir los riesgos asociados con la construcción del mismo.
- Los patrones de diseño sirvieron de apoyo para desarrollar el módulo ya que estos solucionan problemas para la reutilización y efectividad del código.
- Con la realización del diagrama de clases del diseño se obtuvo una visión más exacta del sistema en términos de implementación.

- La creación del modelo de datos permitió visualizar la estructura de la base de datos del módulo, siendo de gran utilidad ya que el mismo guardará los datos de interés para el sistema.

CAPÍTULO 3: Implementación y Prueba

Una de las disciplinas críticas en el desarrollo de software es la implementación, pues es aquí donde se materializa el análisis y diseño. Para llevar a cabo la implementación del producto se hace necesario definir estándares de codificación para establecer un lenguaje común entre las personas que trabajan en él. Además, es necesario construir un diagrama de componentes para una mejor comprensión de los elementos que intervienen en la implementación. La confección del diagrama de despliegue ayuda a conocer el entorno donde se utilizará dicho módulo. Una vez que el producto esté listo se realizan pruebas para demostrar que la solución es conforme con su especificación, que da respuesta a los requisitos del cliente, y así corroborar que se ha obtenido un resultado satisfactorio.

3.1 Diagrama de componentes

Un componente de software es una unidad de composición con interfaces especificadas contractualmente y sólo con dependencias de contexto explícitas. Un componente de software puede implementarse de manera independiente y está sujeto a composición por terceras partes (SOMMERVILLE 2011).

En la **Figura 9** se representan los componentes del módulo.

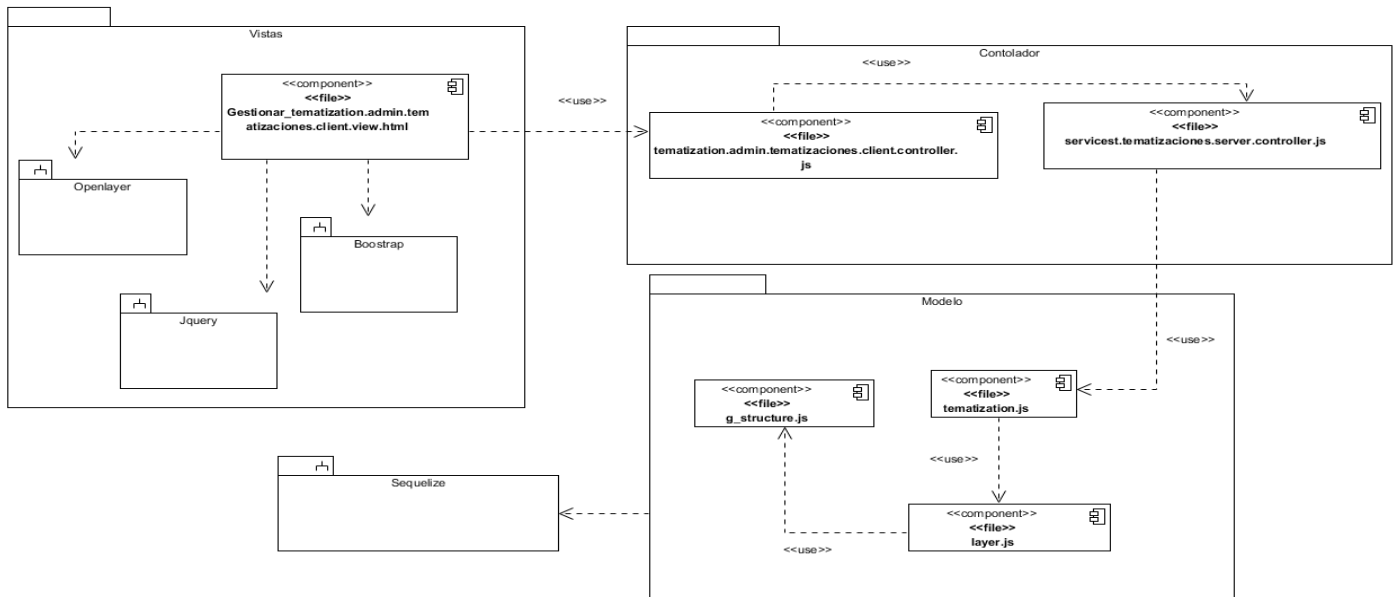


Figura 9: Diagrama de componentes de la solución.

3.2 Diagrama de despliegue

Un **diagrama de despliegue** es un diagrama estructurado que muestra la arquitectura del sistema desde

el punto de vista de la distribución de los **artefectos** del software en los **destinos de despliegue**. Los artefactos representan elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. El destino de despliegue está generalmente representado por un nodo que es o bien de los dispositivos de hardware o algún entorno de ejecución de software. Los nodos pueden ser conectados a través de vías de comunicación para crear sistemas en red de complejidad arbitraria (SARMIENTO 2013).

A continuación se representa en la **Figura 10** el diagrama de despliegue de la solución

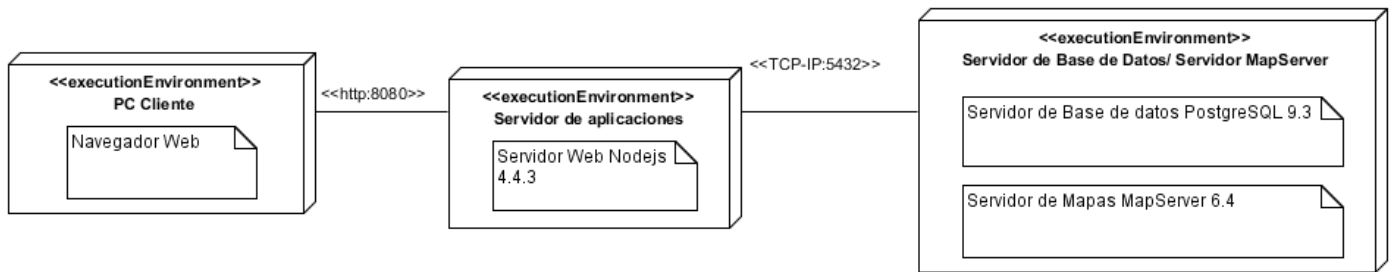


Figura 10: Diagrama de despliegue de la solución.

3.3 Estándares de codificación

Para llevar a cabo un proyecto es necesario definir estándares de codificación para que el código sea más legible y establecer un lenguaje común entre las personas que en él trabajan y el producto final tenga una mejor calidad.

Un **estándar de codificación** comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (MICROSOFT 2017).

La Línea de Productos de Software Aplicativos SIG para la creación del SIG_UCI v3.0 tiene sus propios estándares de codificación para que el código quede claro y entendible para todos sus miembros. A continuación, se muestran algunas pautas que el equipo de desarrollo sigue para la construcción del módulo.

Márgenes, indentación, líneas y espacios.

- La indentación se realizará siempre con tabuladores y no con cuatro puntos, cuidando de no mezclar un estilo con el otro.

- Se dejará una línea en blanco inmediatamente después del inicio de un bloque de código, dígame bloques de funciones, bloques condicionales, bucles, entre otros.
- Se dejará una línea en blanco inmediatamente después del bloque de declaración de variables.
- Se dejará un espacio en blanco antes y después de cada operador, dígame operadores aritméticos, operadores booleanos, paréntesis, llaves, entre otros.
- La llave de apertura de un bloque de código se coloca después de la instrucción anterior y no en una línea en blanco independiente.
- La llave de clausura de un bloque de código tendrá el mismo nivel de indentación que la línea de la llave de apertura.
- Cada bloque de código anidado tendrá un nivel de indentación más que el bloque padre exceptuando las llaves de apertura y cierre.

```
exports.getOneTematizations = function (req, res, next, id) {  
  if ((id % 1 === 0) === false) { //check if it's integer  
    return res.status(404).send({  
      message: 'La tematizacion no es válidaaaaa' + id  
    });  
  }  
  Tematization.find({  
    where: {  
      id_tematization: id  
    },  
  }).then(function (tematization) {  
    if (!tematization) {  
      return res.status(404).send({  
        message: 'No se ha encontrado la tematizacion solicitada'  
      });  
    } else {  
      req.tematization = tematization;  
      next();  
    }  
  }).catch(function (err) {  
    return next(err);  
  });  
};
```

Figura 11: Estándares de codificación para márgenes, indentación, líneas y espacios.

Variables, identificadores y parámetros.

- Los identificadores se escribirán en idioma inglés evitando el uso de contracciones y empleando términos comunes cercanos al inglés técnico.
- Todas las variables de un bloque de código se declararán al inicio del método en cuestión.
- Las variables se declararán por línea y empleando la palabra reservada `<var>` que nunca podrá omitirse, aunque el lenguaje lo permita.
- Todas las variables deben ser inicializadas en el momento de declaración para indicar la naturaleza de su contenido. En caso de no estar disponible datos para su inicialización se emplea el cero (0) para enteros, para números de coma flotante (0.0), doble comilla para cadenas, corchetes para arreglos y llaves para objetos.

```

exports.getCantResidentsByEdif = function (req, res) {
  process.env['NODE_TLS_REJECT_UNAUTHORIZED'] = '0';
  var url = 'http://ias.uci.cu:9763/services/ResidenciaWS?wsdl';

  soap.createClient(url, function (err, client) {
    if (err)
      return res.status(503).send({
        message: 'No se pudo establecer la comunicación con el servicio externo'
      });
    else
      var critery = req.body.tematization.properties.critery;
      var numedif = [];
      var edif = [];
  }
}

```

Figura 12: Estándares de codificación para variables, identificadores y parámetros.

3.4 Pruebas de la propuesta de solución

A todo software, una vez concluida su implementación, es necesario aplicarles un conjunto de pruebas para verificar su correcto funcionamiento.

“Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán” (PRESSMAN 2010).

El único instrumento adecuado para determinar el *status* de la calidad de un producto de software es el **proceso de pruebas**. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos (SOFTWARE P.D 2005).

El **proceso de pruebas** es una actividad en la cual se ejecuta un sistema o uno de sus componentes, los resultados se observan, se registran y se realiza una evaluación de algún aspecto. El objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente pueden demostrar que existen defectos en el software, por eso es necesario diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo posibles.

Una buena prueba de software debe de tener los siguientes objetivos:

- Probar si el software no hace lo que debe hacer.
- Probar si el software hace lo que no debe hacer.

Para llevar a cabo el proceso de prueba existen varios niveles, entre ellos se encuentran:

- Pruebas a nivel de unidad.
- Pruebas a nivel de integración.

- Pruebas a nivel de sistema.
- Pruebas de aceptación.

3.4.1 Definición y descripción de las pruebas

Para el buen funcionamiento del módulo se definió por parte de los desarrolladores hacer pruebas a nivel de sistema.

Pruebas a nivel de sistema.

“La prueba a nivel de sistema es una serie de diferentes pruebas cuyo propósito principal es ejercitar por completo el sistema basado en computadora. Aunque cada prueba tenga un propósito diferente, todo él funciona para identificar que los elementos del sistema se hayan integrado de manera adecuada y que se realicen las funciones asignadas” (PRESSMAN 2010).

Según (PRESSMAN 2010) para este nivel de prueba existen diferentes tipos de pruebas que se pueden aplicar, entre ellas se tienen:

- Pruebas de contenido.
- Pruebas de funcionalidad.
- Pruebas de estructuras.
- Pruebas de usabilidad.
- Pruebas de navegabilidad.
- Pruebas de rendimiento.
- Pruebas de compatibilidad.
- Pruebas de interoperabilidad.
- Pruebas de seguridad.

Para realizar estos tipos de pruebas, se utilizan los métodos de caja negra que tratan de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y se produce el resultado esperado, y el método de caja blanca que estudia los detalles del procesamiento interno del componente y hace modificaciones en el nivel de código para eliminar cualquier conflicto (PRESSMAN 2010). De estos métodos el utilizado para el proceso de pruebas del módulo fue el método de caja negra.

Pruebas Funcionales o de Caja Negra.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software. Las técnicas de pruebas de caja negra permiten derivar un conjunto de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Según

(PRESSMAN 2010) en las pruebas de caja negra se intentan encontrar errores en las categorías siguientes:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en la estructura de datos o en el acceso a bases de datos externas.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

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 las habilidades 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 las pruebas de caja negra del módulo de análisis temático para el SIG_UCI v3.0 se utilizó la técnica de Partición de equivalencia.

Casos de pruebas.

Un **caso de prueba** es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de prueba especifican una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Se realizan con el fin de asegurar que el producto es operativo.

Con los **casos de prueba** de caja negra se pretende demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene (SOFTWARE P.D 2005).

Para validar que el módulo funcione correctamente se definieron un conjunto de casos de pruebas que reúnen la cantidad de escenarios posibles. A continuación la **Tabla 3** representa el caso de prueba *Realizar tematización según estilo gráfico de barras*. Para ver los demás casos de pruebas remitirse al **Anexo 3**.

Descripción General: Permitir incluir una tematización estilo gráfico de barras en el sistema.

Condición de ejecución: Para incluir una tematización en el sistema hay que estar autenticado con el rol Gestor de tematizaciones.

Tabla 3: Caso de prueba “Realizar tematización según estilo gráfico de barras”.

Escenario	Descripción	Nombre	Descripción	Capa	Criterios	Color	Respuesta del sistema	Flujo central
EC 1.1 Opción de Adicionar los datos	El usuario selecciona la funcionalidad	V	V	V	V	V	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".
EC 1.2: Introducir datos incorrectos.	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización gráfico de barras, completa los campos del formulario y presiona el botón adicionar.	tematizació	Se realiza	Edificios	Capacidad	rojo	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Gráfica de barras" luego muestra un formulario para introducir los datos de la tematización. Si el usuario adiciona los datos incorrectos en cualquier campo muestra un mensaje de error que diga "Datos incorrectos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción "Adicionar". 6. Seleccionar la opción "Gráfica de barras".
		tematizació	Se realiza	Edificios	Capacidad	N/A		
		tematizació	Se realiza	Edificios	Capacidad	rojo		
		tematizació	Se realiza	Edificios	N/A	rojo		
		tematizació	Se realiza	N/A	Capacidad	rojo		
		tematizació	Se realiza	Edificios	Capacidad	rojo		
		tematizació	Se realiza	Edificios	Capacidad	rojo		
		tematizació	Se realiza	N/A	Capacidad	rojo		
		tematizació	Se realiza	Edificios	Capacidad	rojo		
		tematizació	Se realiza	N/A	N/A	N/A		
		tematizació	Se realiza	N/A	N/A	N/A		
		tematizació	Se realiza	N/A	N/A	N/A		
		tematizació	Se realiza	N/A	N/A	N/A		
		tematizació	Se realiza	N/A	N/A	N/A		
		EC 1.3: Dejar campos obligatorios en blanco	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización gráfico de barras, completa los campos del formulario y presiona el botón adicionar.	V	V	V		
tematizació	Se realiza	Edificios	Capacidad	N/A				
tematizació	Se realiza	Edificios	Capacidad	rojo				
tematizació	Se realiza	Edificios	N/A	rojo				
tematizació	Se realiza	N/A	Capacidad	rojo				
tematizació	Se realiza	Edificios	Capacidad	rojo				
tematizació	Se realiza	Edificios	Capacidad	rojo				
tematizació	Se realiza	N/A	Capacidad	rojo				
tematizació	Se realiza	Edificios	Capacidad	rojo				
tematizació	Se realiza	N/A	N/A	N/A				
tematizació	Se realiza	N/A	N/A	N/A				
tematizació	Se realiza	N/A	N/A	N/A				
tematizació	Se realiza	N/A	N/A	N/A				
tematizació	Se realiza	N/A	N/A	N/A				
EC 1.4: Opción de Cancelar.	El usuario selecciona la funcionalidad	N/A	N/A	N/A	N/A	N/A	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".

Tabla 4: Descripción de variables del caso de prueba "Realizar tematización según estilo gráfico de barras".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Cadena de caracteres (no admite espacios en blanco o escribir solo espacios, no admite caracteres especiales) Longitud máxima 250 caracteres.
2	Descripcion	Campo de texto	No	Cadena de caracteres
3	Capa	Lista desplegable	No	Campo desplegable que contiene las capas que conforma el sistema, en este caso por defecto tiene la capa Edificios UCI.
4	Criterios	Lista desplegable	No	Lista desplegable que contiene
5	Color	Campo de selección	No	Campo de selección que elige el

Resultados de las pruebas Funcionales o de Caja negra.

Durante la realización de las pruebas de caja negra se identificaron varias no conformidades asociadas a las funcionalidades y los errores ortográficos, las cuales fueron disminuyendo considerablemente con cada iteración realizada.

Tabla 5: Resultados de las pruebas de caja negra.

Iteraciones	Funcionalidades	Errores ortográficos
1	9	6
2	3	2
3	0	0

En la **Figura 13** se aprecia una gráfica con los resultados anteriores.

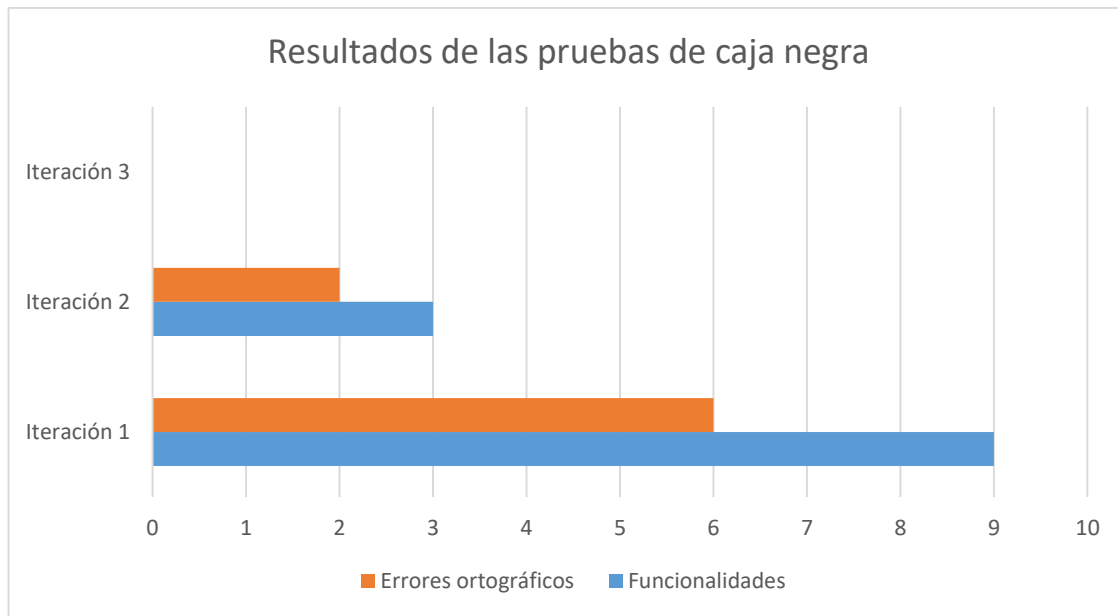


Figura 13: Gráfica de los resultados de las pruebas de caja negra.

Pruebas de rendimiento.

Las pruebas de rendimiento deben diseñarse para garantizar que el sistema procese su carga pretendida. Esto implica que el sistema tenga una serie de pruebas en la que la carga se va incrementando considerablemente hasta que el rendimiento del sistema sea inaceptable. Al igual que otras pruebas, se encargan de demostrar que el sistema cumple con sus requerimientos, como de descubrir problemas y defectos en el sistema (SOMMERVILLE 2011).

Pruebas de Carga y Estrés.

Consisten en realizar pruebas que se vayan acercando a la cantidad máxima de carga del diseño del sistema, hasta que falla. Según (SOMMERVILLE 2011) estas pruebas tienen dos funciones.

- Prueba el comportamiento de fallo en el sistema, donde surgen circunstancias a través de una combinación no esperada de eventos donde la carga que hay sobre el sistema supere la carga máxima anticipada. Las pruebas de estrés demuestran que la sobrecarga del sistema hace que falle poco en vez de colapsar bajo su carga.
- Sobrecarga el sistema y hace que se descubran defectos que normalmente no serían descubiertos.

Para la ejecución de este tipo de pruebas se utilizó la herramienta JMeter v2.12, para ello se tuvo en cuenta el siguiente entorno de desarrollo:

1 PC cliente con 8 GB de memoria RAM con sistema operativo GNU/Linux Ubuntu 14.04 y 107,7 GB de espacio en disco duro y microprocesador Intel Core i5, cuenta con una PC servidora de 4 GB de memoria

RAM, 500 GB de disco duro, microprocesador i3, y sistema operativo GNU/Linux Ubuntu 14.04.

Pruebas de Carga.

Para las pruebas de carga el tiempo promedio de una ejecución debe ser de 3 segundos. A continuación, se muestran las variables analizadas.

Usuarios: Cantidad de usuarios conectados concurrentemente.

Muestra: Cantidad de peticiones realizadas por cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Min: Tiempo mínimo que demora un hilo en acceder a una página.

Max: Tiempo máximo que demora un hilo en acceder a una página.

% de Error: Por ciento de error en las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): Se mide por cantidad de solicitudes por segundo.

Kb/Seg: El rendimiento se mide en cantidad de Kilobytes por segundo.

En la **Tabla 6** se representan los resultados que arrojaron las pruebas de rendimiento.

Tabla 6: Resultado de las pruebas de carga.

Usuarios	Muestra	Media	Min	Max	% Error	Rendimiento	Kb/Seg
20	150	1202	2	1506	0	9.4	40.1
40	300	2034	3	2054	0	7.6	30.4
80	600	2308	3.9	2495	0	5.3	22.2
120	920	3089	4	3175	0	3.2	11.7

Las pruebas realizadas demuestran que el sistema es capaz de responder a 150 peticiones realizadas por 20 usuarios conectados simultáneamente en un tiempo promedio de 1202 milisegundos con un 0% de error.

Por otra parte, se realizaron 300 peticiones por 40 usuarios, en este caso el sistema respondió en un promedio de tiempo de 2034 milisegundos con un 0% de error.

Se realizaron 600 peticiones por parte de 80 usuarios y el sistema respondió en un tiempo promedio de 2308 milisegundos con un 0% de error.

Por último, se realizaron 920 peticiones iniciadas por 120 usuarios donde el sistema respondió en un tiempo

promedio de 3089 milisegundos con un 0% de error.

Pruebas de Estrés.

Esta prueba se realizó con un total de 200 usuarios arrojando un total de 1200 peticiones, para esta muestra el sistema demostró poca efectividad ya que el tiempo máximo de respuesta demoró más de 11 segundos.

3.5 Conclusiones parciales

Mediante los resultados obtenidos en las disciplinas de implementación y pruebas internas de la metodología AUP_UCI los autores concluyen que:

- A través de la implementación se podrá materializar una solución que responda a las especificaciones y necesidades del cliente.
- La implementación del modelo de componentes permite conocer cómo interactúan los distintos subsistemas del módulo.
- La confección del modelo de despliegue brinda información de cómo está distribuido el entorno donde se utilizará dicho módulo.
- La definición de los estándares de codificación estableció en el equipo de desarrollo un lenguaje común entre sus miembros.
- Se realizaron pruebas funcionales o de caja negra y pruebas de rendimiento, demostrando que las funciones del módulo son efectivas y producen un resultado satisfactorio.

CONCLUSIONES GENERALES

Como resultado del trabajo realizado se arribaron a las siguientes conclusiones:

- Mediante el estudio de los referentes teóricos y metodológicos que sustentan la investigación se pudo conocer los distintos tipos de mapas temáticos existentes y la forma de representar estos en los SIG.
- El análisis de los distintos sistemas que hacen uso de los mapas temáticos permitió conocer cómo representar la información que el cliente necesita y así favorecer el proceso de toma de decisiones en la universidad.
- La confección de los artefactos permitió una mejor visión del funcionamiento del sistema, logrando un entendimiento común entre los implicados.
- Para realizar las tematizaciones se obtuvieron 10 criterios por los cuales tematizar, mediante estos se pudo conocer el tipo de información que el cliente necesita y de esta forma saber mediante que tematización representarlos en el mapa.
- La definición y descripción de los requisitos ejemplificaron las particularidades que el sistema debe cumplir para satisfacer las necesidades del cliente.
- La aplicación de las pruebas posibilitó evaluar el producto, comprobándose el correcto funcionamiento del módulo y la ausencia de errores.

RECOMENDACIONES

Luego de realizada la presente investigación los autores recomiendan:

- Representar la información de la universidad mediante otros tipos de mapas temáticos, ejemplo de ello los mapas de puntos y los mapas de cartogramas.
- Extender la investigación al resto de las áreas de la universidad.

Bibliografía.

1. SPHINX, 2017. Acerca de — MapServer 7.0.6 documentation. *MapServer* [en línea]. [Consulta: 29 diciembre 2016]. Disponible en: <http://www.mapserver.org/es/about.html#about>.
2. BALDOQUÍN, N. y VALLÍN, A., 2015. Componente para la mensajería instantánea y notificaciones sobre Node.js. S.I.:
3. BAÑOS, O., 2012. Análisis del módulo de Admisnistración de la Plataforma Atlas. . La Habana:
4. BOOCH, J. y RUMBAUGH, J., 2005. *El Lenguaje Unificado de Modelado*. S.I.: s.n.
5. CARTOGRÁFICA DE CANARIAS, 2015. Mapas Temáticos | Mapas de Canarias. [en línea]. [Consulta: 20 noviembre 2016]. Disponible en: <https://www.grafcan.es/mapas-tematicos>.
6. CASADO, C., 2013. Programación web » WebStorm. [en línea]. [Consulta: 20 diciembre 2016]. Disponible en: <http://multimedia.uoc.edu/blogs/pw/es/2013/04/16/webstorm/>.
7. CIENCIA GEOGRÁFICA, 2014. ¿Qué son los mapas geográficos? [en línea]. [Consulta: 20 noviembre 2016]. Disponible en: <http://cienciageografica.carpetapedagogica.com/2011/08/que-son-los-mapas-geograficos.html>.
8. COSTAL, D., 2015. Introducción al diseño de bases de datos. . S.I.: P01/75002/00578.
9. CUADRADO, D.G., 2016. Apuntes Mapas Topográficos. . S.I.:
10. EGUÍLUZ, J., 2013. Introducción a JavaScript. . S.I.: s.n.,
11. FERNÁNDEZ, Y. y DÍAZ, Y., 2012. Patrón Modelo-Vista-Controlador. . S.I.:
12. GUTIÉRREZ, J.E., 2012. *Cartografía Temática – Recopilación Apuntes*. S.I.: s.n.
13. HERRERA, E. y COCA, L.L., 2015. *Módulo de análisis espacial estadístico descriptivo para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas (SIG_UCI)*. S.I.: s.n.
14. IBÁÑEZ, S., MORENO, H. y GISBERT, J.M., 2012. Escalas en un mapa. . S.I.:
15. ISSI, G., 2003. Metodologías Ágiles en el desarrollo de software. ,
16. KEVANORT, E. por, 2015. Sequelize : ORM para tecnologías Node.js e IO.js | TAWS. [en línea]. [Consulta: 3 marzo 2017]. Disponible en: <http://blog.espol.edu.ec/taws/2015/11/24/sequelize-orm-para-tecnologias-node-js-y-io-js/>.
17. LARMAN, C., 1999. *UML y patrones. Una introducción al análisis y diseño orientado*. Madrid: s.n.

18. MICROSOFT, J., 2017. Revisiones de código y estándares de codificación. [en línea]. [Consulta: 13 mayo 2017]. Disponible en: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx).
19. OLAYA, V., 2014. *Sistemas de Información Geográfica*. S.l.: CreateSpace Independent Publishing Platform (Amazon).
20. PANTOJA, Y., 2010. *MaGIStral: Sistema Integral de gestión de recursos para el proyecto GENESIG*. La Habana: s.n.
21. PRESSMAN, R., 2010. *Ingeniería de Software. Un enfoque práctico. Séptima edición*. México: s.n. 9701054733
22. QUEZADA, J.A., 2007. *Pruebas de software*. S.l.: s.n.
23. QUIÑONES, E., 2017. PostgreSQL: The world's most advanced open source database. [en línea]. [Consulta: 22 marzo 2017]. Disponible en: <https://www.postgresql.org/>.
24. RAMSEY, P., 2017. PostGIS — Spatial and Geographic Objects for PostgreSQL. [en línea]. [Consulta: 21 diciembre 2017]. Disponible en: <http://postgis.net/>.
25. RODRÍGUEZ, A., 2010. *Servicio de mapas temáticos*. La Habana: s.n.
26. SÁNCHEZ, T.R., 2015. Metodología de desarrollo para la Actividad productiva de la UCI. . La Habana:
27. SANTAMARÍA, V.H., 2015. *NUEVAS HERRAMIENTAS PARA EL ANÁLISIS DE OPINIÓN EN FLUJOS DE TEXTO*. Pamplona: s.n.
28. SARMIENTO, F., Johana, 2013. Visión General de los Diagramas de Despliegue. *UML* [en línea]. [Consulta: 13 mayo 2017]. Disponible en: <http://umldiagramadespliegue.blogspot.com/>.
29. SOFTLAND, 2017. Visual Paradigm for UML Standard 8.0 Free Download. [en línea]. Disponible en: <http://visual-paradigm-for-uml-standard.soft112.com/>.
30. SOFTWARE P.D, 2005. Las pruebas de software forman el único instrumento adecuado para determinar la calidad de software. [en línea]. [Consulta: 5 mayo 2017]. Disponible en: <http://www.pruebasdesoftware.com/laspruebasdesoftware.htm>.
31. SOMMERVILLE, I., 2011. *Ingeniería de Software*. México: s.n. 978-607-32-0603-7
32. UDIG, 2010. Software UDig | Sistemas de Información Geográfica. [en línea]. [Consulta: 4 diciembre 2016]. Disponible en: <http://sig.cea.es/udig>.
33. VISUAL PARADIGM, 2010. Visual Paradigm 8.0 (formerly VP-UML 8.0) Released. [en línea]. [Consulta: 20 diciembre 2016]. Disponible en: <https://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

GLOSARIO DE TÉRMINOS

A

Acote: Limitar un terreno con cotos para reservar su uso.

B

Bifurcan: División en dos partes iguales.

C

Capa: Elemento mediante el cual se representa un evento sobre el mapa.

Cartografía: Ciencia que estudia los mapas y cartas geográficas y cómo realizarlos.

D

Demografía: Disciplina social que estudia estadísticamente a la población humana.

E

Estándares OGC: Estándares asociados a las interfaces, codificación, perfiles, esquemas de aplicación y documentos de buenas prácticas que permiten crear sistemas de información que pueden fácilmente intercambiar información geográfica e instrucciones con otros sistemas de información.

F

Fenómeno: Manifestación de una actividad que se produce en la naturaleza y se percibe a través de los sentidos.

G

Geoespacial: Concede información de datos geográficos, así como su ubicación.

I

Infraestructura: Conjunto de medios técnicos, servicios e instalaciones necesarios para el desarrollo de una actividad.

L

Longitudinales: Es utilizado para referirse a la distancia de un punto a otro.

M

Migraciones: Desplazamiento de la población que se produce desde un lugar de origen a otro destino.

Módulo: Porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realiza comúnmente una de dichas tareas.

P

Plataforma: Es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software.

R

Regionalización: Delimitación geográfica que se

realiza en consideración de elementos comunes, sean económicos, sociales, culturales, geográficos, administrativos y/o políticos.

Relieve: Formas que presenta la corteza terrestre o litosfera en la superficie, tanto en relación con las tierras emergidas como en el relieve oceánico

T

Tematizaciones: Conjunto de características que permiten evaluar un determinado evento sobre la superficie de un mapa.

Topografía: Conjunto de principios y procedimientos que tienen como objetivo la representación gráfica de la superficie terrestre, con sus formas y detalles.

ANEXOS

Anexo 1

1. ¿Cuáles son los procesos que se llevan a cabo en el área de residencia?
2. ¿Qué tipo de información se maneja en su área?
3. ¿Cómo se realizan los procesos para obtener esa información?
4. ¿Existe algún sistema que maneje y gestione este tipo de información?
5. ¿Cuál es la información más importante para usted?
6. ¿Quiénes deben ser los involucrados que pueden tener acceso a dicha información?
7. Algún dato importante que desee añadir.

Anexo 2

Tabla 7: Historia de usuario del requisito "Realizar tematización según estilo proporcional."

Número: 1	Nombre del requisito: Realizar tematización según estilo proporcional
Programador (es): Asley Arias Sainz y Luis David González Sarduy	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Rotura de host donde esté instalado el servidor de Bases de Datos. • Problemas eléctricos. • Atrasos generados por el cliente debido a la poca disponibilidad de los mismos. 	Tiempo Real: 1 semana
Descripción:	

1- Objetivo:

-Permite al usuario insertar una tematización según el estilo proporcional a un área determinada.

2- Acciones para lograr el objetivo:

Para insertar una tematización hay que:

- Tener en cuenta el tipo de tematización que se desea realizar, (en este caso estilo proporcional), y se deben de llenar los datos que brinda la interfaz dígase capa, criterio y descripción, nombre, color.
- Debe estar autenticado en el sistema.

3- Comportamientos válidos y no válidos:

Los campos capa, criterios, descripción, nombre y color son obligatorios

-Nombre: Se inserta un nombre a la tematización.

- Descripción: Campo de texto para describir la tematización.

- Capa: botón para seleccionar la capa a tematizar (Edificios UCI).

- Criterios: botón para seleccionar el criterio por el cual se desea tematizar.

-Color: Se especifica el color de la tematización.

Observaciones:**Prototipo de interfaz:**

Tabla 8: Historia de usuario del requisito "Listar tematizaciones."

Número: 6	Nombre del requisito: Listar Tematizaciones
Programador (es): Asley Arias Sainz y Luis David González Sarduy	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Rotura de host donde esté instalado el servidor de Bases de Datos. • Problemas eléctricos. • Atrasos generados por el cliente debido a la poca disponibilidad de los mismos. 	Tiempo Real: 1 semana
Descripción:	

1- Objetivo:

-Permite al usuario listar las tematizaciones

2- Acciones para lograr el objetivo:

Para listar una tematización hay que:

-Estar autenticado en el sistema.

-el usuario debe crear una tematización

3- Comportamientos válidos y no válidos:

Los campos nombres, descripción y tipo que brinda la interfaz son obligatorios

-Nombre: Se inserta un nombre a la tematización.

- Descripción: Campo de texto para describir la tematización.

-Tipo: Se inserta el tipo de tematización que es.

-Acciones: Se puede visualizar o eliminar la tematización.

Observaciones:**Prototipo de interfaz:**

The screenshot shows the 'Tematizaciones' page in the SIGUCI system. The interface includes a sidebar menu with options like 'Inicio', 'Grupo de capas', 'Tipos de datos', 'Estructuras', 'Capas', 'Acciones', 'Roles', 'Usuarios', and 'Tematizaciones'. The main content area displays a table of thematizations with the following data:

Nombre	Descripción	Tipo	Acciones
1	descr	Estilo proporcional	✎ 👁 ✕
2	des	Grafico de barra	✎ 👁 ✕
3	descr	Grafico de pastel	✎ 👁 ✕

Additional interface elements include a search bar with the text 'Criterio de búsqueda', pagination controls showing 'Mostrando de 1 a 3 de 3 entradas', and a footer with the copyright notice '©2016 Universidad de las Ciencias Informáticas'.

Tabla 9: Historia de usuario del requisito "Realizar tematización según estilo gráfico de pastel."

Número: 5	Nombre del requisito: Realizar tematización según estilo gráfico de pastel
Programador (es): Asley Arias Sainz y Luis David González Sarduy	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Rotura de host donde esté instalado el servidor de Bases de Datos. • Problemas eléctricos. • Atrasos generados por el cliente debido a la poca disponibilidad de los mismos. 	Tiempo Real: 1 semana
Descripción: 1- Objetivo: -Permite al usuario insertar una tematización según el estilo gráfico de pastel a un área determinada. 2- Acciones para lograr el objetivo: Para insertar una tematización hay que: Tener en cuenta el tipo de tematización que se desea realizar, (en este caso estilo gráfico de pastel), y se deben de llenar los datos que brinda la interfaz dígase capa, criterio y descripción, color, nombre Debe estar autenticado en el sistema. 3- Comportamientos válidos y no válidos: Los campos capa, criterios y descripción, color y nombre son obligatorios -Nombre: Se inserta un nombre a la tematización. - Descripción: Campo de texto para describir la tematización. - Capa: botón para seleccionar la capa a tematizar (Edificios UCI). - Criterios: botón para seleccionar el criterio por el cual se desea tematizar.	

-Color: Se inserta un color a la tematización.

Observaciones:

Prototipo de interfaz:

The screenshot displays the 'Estilo Grafico de pastel' configuration page within the SIGUCI system. The interface includes a left-hand navigation menu with the following items: Inicio, Grupo de capas, Tipos de datos, Estructuras, Capas, Acciones, Roles, Usuarios, and Tematizaciones. The main content area is titled 'Estilo Grafico de pastel' and contains the following fields:

- Nombre ***: A text input field containing 'estilo grafico de pastel'.
- Descripción**: A text input field containing 'pequeña descripción de la tematización'.
- Capas ***: A dropdown menu with the selected option '- Selecciona una capa -'.
- Criterios**: A dropdown menu with the selected option 'Edificios UC'.
- Color 1 ***: A color selection field with a color swatch and the text 'Ej. rgb(255, 250, 250), #FFFFFF'.
- Color 2 ***: A color selection field with a color swatch and the text 'Ej. rgb(255, 250, 250), #FFFFFF'.

At the bottom of the form, there are two buttons: 'Adicionar' (highlighted in blue) and 'Cancelar'.

Tabla 10: Historia de usuario del requisito "Realizar tematización según estilo categorizado."

Número: 2	Nombre del requisito: Realizar tematización según estilo categorizado
Programador (es): Asley Arias Sainz y Luis David González Sarduy	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Rotura de host donde esté instalado el servidor de Bases de Datos. • Problemas eléctricos. • Atrasos generados por el cliente debido a la poca disponibilidad de los mismos. 	Tiempo Real: 1 semana
<p>Descripción:</p> <p>1- Objetivo:</p> <p>-Permite al usuario insertar una tematización según el estilo categorizado a un área determinada.</p> <p>2- Acciones para lograr el objetivo:</p> <p>Para insertar una tematización hay que:</p> <ul style="list-style-type: none"> - Tener en cuenta el tipo de tematización que se desea realizar, (en este caso estilo categorizado), y se deben de llenar los datos que brinda la interfaz dígame capa, criterio y descripción, color, nombre - Debe estar autenticado en el sistema. <p>3- Comportamientos válidos y no válidos:</p> <p>Los campos capa, criterios y descripción, color y nombre son obligatorios</p> <ul style="list-style-type: none"> -Nombre: Se inserta un nombre a la tematización. - Descripción: Campo de texto para describir la tematización. - Capa: botón para seleccionar la capa a tematizar (Edificios UCI). - Criterios: botón para seleccionar el criterio por el cual se desea tematizar. -Color: Se inserta un color a la tematización. 	

Observaciones:**Prototipo de interfaz:**

The screenshot displays the SIGUCI web application interface. On the left is a dark blue sidebar with the following menu items: Inicio, Grupo de capas, Tipos de datos, Estructuras, Capas, Acciones, Roles, Usuarios, and Tematizaciones. The main content area shows the breadcrumb path: Inicio > Tematizaciones > Estilo Categorizado. The form contains the following fields:

- Nombre:** Input field containing the value "4".
- Descripción:** Text area containing the value "gestor".
- Capas:** Dropdown menu with "Edificios UCI" selected.
- Criteria:** Dropdown menu with "--Seleccione un Criterio--" selected.
- Color 1:** Input field with "rgb(76, 12, 12)" and a dark red color swatch. Below it is the example text "Ej: rgb(255, 250, 250), #ffffff".
- Color 2:** Input field with "rgb(82, 20, 20)" and a dark red color swatch. Below it is the example text "Ej: rgb(255, 250, 250), #ffffff".
- Color 3:** Input field with "rgb(186, 131, 131)" and a reddish-brown color swatch. Below it is the example text "Ej: rgb(255, 250, 250), #ffffff".

At the bottom of the form are two buttons: "Adicionar" (highlighted in blue) and "Cancelar".

Anexo 3

Tabla 11: Caso de prueba del requisito "Realizar tematización según estilo gráfico de pastel"

Escenario	Descripción	Nombre	Descripción	Capa	Criterios	Color	Respuesta del sistema	Flujo central
EC 1.1 Opción de Adicionar los datos	El usuario selecciona la funcionalidad	V	V	V	V	V	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".
EC 1.2: Introducir datos incorrectos.	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización gráfico de pastel, completa los campos del formulario y presiona el botón adicionar.	V	V	V	V	I	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Gráfica de Pastel" luego muestra un formulario para introducir los datos de la tematización. Si el usuario adiciona los datos incorrectos en cualquier campo muestra un mensaje de error que diga "Datos incorrectos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción. 5. Seleccionar la opción "Tematizaciones". 6. Seleccionar la opción "Adicionar".
EC 1.3: Dejar campos obligatorios en blanco	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización gráfico de pastel, completa los campos del formulario y presiona el botón adicionar.	V	V	V	V	I	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Gráfica de pastel" luego muestra un formulario para introducir los datos de la tematización. Si el usuario deja algún campo en blanco muestra un mensaje de error que diga "Datos incompletos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción. 5. Seleccionar la opción "Tematizaciones". 6. Seleccionar la opción "Adicionar".
EC 1.4: Opción de Cancelar.	El usuario selecciona la funcionalidad	N/A	N/A	N/A	N/A	N/A	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".

Tabla 12: Caso de prueba del requisito "Realizar tematización según estilo categorizado"

Escenario	Descripción	Nombre	Descripción	Capa	Criterios	Color	Respuesta del sistema	Flujo central	
EC 1.1 Opción de Adicionar los datos	El usuario selecciona la funcionalidad	V	V	V	V	V	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".	
EC 1.2: Introducir datos incorrectos.	estilo caegorizado	tematizació	Se realiza	Edificios	Edificios	azul	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Estilo categorizado" luego muestra un formulario para introducir los datos de la tematización. Si el usuario adiciona los datos incorrectos en cualquier campo muestra un mensaje de error que diga "Datos incorrectos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción "Adicionar". 6. Seleccionar la opción "Estilo categorizado".	
		V	V	V	V	I			
		V	V	V	I	V			
		tematizació	Se realiza	Edificios	Edificios	N/A			azul
		V	V	I	V	V			
		tematizació	Se realiza	N/A	Edificios	Edificios			azul
		V	I	V	V	V			
		tematizació	Se realiza	Edificios	Edificios	Edificios			azul
		I	V	V	V	V			
		.;][ap	Se realiza	Edificios	Edificios	Edificios			azul
		I	I	V	V	V			
		.;][ap	Se realiza	Edificios	Edificios	Edificios			azul
		I	I	I	V	V			
		.;][ap	Se realiza	N/A	Edificios	Edificios			azul
		I	I	I	I	V			
		.;][ap	Se realiza	N/A	N/A	N/A			azul
		V	I	I	I	I			I
tematizació	Se realiza	N/A	N/A	N/A	N/A				
V	V	I	I	I	I				
tematizació	Se realiza	N/A	N/A	N/A	N/A				
V	V	V	I	I	I				
tematizació	Se realiza	Edificios	Edificios	N/A	N/A				
EC 1.3: Dejar campos obligatorios en blanco	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización estilo categorizado, completa los campos del formulario y presiona el botón adicionar.	V	V	V	V	V	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Estilo categorizado" luego muestra un formulario para introducir los datos de la tematización. Si el usuario deja algún campo en blanco muestra un mensaje de error que diga "Datos incompletos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción "Adicionar". 6. Seleccionar la opción "Estilo categorizado".	
tematizació	Se realiza	Edificios	Edificios	Edificios	N/A	azul			
V	V	V	I	V	V				
tematizació	Se realiza	Edificios	Edificios	N/A	azul				
V	V	I	V	V					
tematizació	Se realiza	N/A	Edificios	Edificios	azul				
V	I	V	V	V					
tematizació	Se realiza	Edificios	Edificios	Edificios	azul				
I	V	V	V	V					
Se realiza	Edificios	Edificios	Edificios	Edificios	azul				
I	I	V	V	V					
I	I	I	V	V					
I	I	I	I	V					
I	I	N/A	Edificios	Edificios	azul				
I	I	I	I	I	I				
tematizació	Se realiza	N/A	N/A	N/A	N/A				
V	V	I	I	I	I				
tematizació	Se realiza	N/A	N/A	N/A	N/A				
V	V	V	I	I	I				
tematizació	Se realiza	Edificios	Edificios	N/A	N/A				
I	I	I	I	I	I				
I	I	N/A	N/A	N/A	N/A				
EC 1.4: Opción de Cancelar.	El usuario selecciona la funcionalidad	N/A	N/A	N/A	N/A	N/A	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".	

Tabla 13: Caso de prueba del requisito "Listar tematizaciones"

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1.1 Opción de listar las tematizaciones.	Se listan los datos de una tematización	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción tematizaciones.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones".
EC 1.2 Opción de visualizar tematización	Selecciona la opción Visualizar tematización.	El sistema brinda la posibilidad de visualizar la tematización. <u>Ver CP: "Visualizar tematización".</u>	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción visualizar
EC 1.3 Eliminar una tematización del listado.	Selecciona la opción Eliminar tematización.	El sistema brinda la posibilidad de eliminar la tematización. <u>Ver CP: "Eliminar tematización".</u>	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción eliminar

Tabla 14: Caso de prueba del requisito "Visualizar tematizaciones"

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción de visualizar tematizaciones.	Selecciona la opción de visualizar una tematización.	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción tematizaciones, y luego si hay alguna creada , este podrá visualizarla presionando el botón visualizar.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la acción visualizar.
EC 1.2 Opción de cancelar.	El usuario selecciona la opción de Cancelar.	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción tematizaciones, y luego si hay alguna creada , este podrá visualizarla presionando el botón visualizar. Una vez visualizada la tematización podrá cancelarla y presionando el botón cancelar y regresará al listado de tematizaciones.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción

Tabla 15: Caso de prueba del requisito "Eliminar tematizaciones"

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Eliminar tematización	Selecciona la opción de eliminarr una tematización.	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción tematizaciones, y luego si hay alguna creada , este podrá eliminarla presionando el botón eliminar.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la acción eliminar.
EC 1.2 Opción Aceptar	Selecciona la opción Aceptar.	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción tematizaciones, y luego si hay alguna	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración".
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración el usuario elige la opción	<ol style="list-style-type: none"> 1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción

Tabla 16: Caso de prueba del requisito "Realizar tematización según estilo proporcional"

Escenario	Descripción	Nombre	Descripción	Capa	Criterios	Color	Respuesta del sistema	Flujo central	
EC 1.1 Opción de Adicionar los datos	El usuario selecciona la funcionalidad	V	V	V	V	V	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".	
EC 1.2: Introducir datos incorrectos.	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización estilo proporcional, completa los campos del formulario y presiona el botón adicionar.	tematizació	Se realiza	Edificios	Cantidad	N/A	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Estilo proporcional" luego muestra un formulario para introducir los datos de la tematización. Si el usuario adiciona los datos incorrectos en cualquier campo muestra un mensaje de error que diga "Datos incorrectos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción "Adicionar". 6. Seleccionar la	
		V	V	V	V	I			
		V	V	V	I	V			
		tematizació	Se realiza	Edificios	Cantidad	N/A			amarillo
		V	V	I	V	V			
		tematizació	Se realiza	N/A	Cantidad	amarillo			
		I	I	V	V	V			
		.;][ap	Se realiza	Edificios	Cantidad	amarillo			
		I	I	V	V	V			
		.;][ap	.;][klnn-	Edificios	Cantidad	amarillo			
		I	I	I	V	V			
		.;][ap	.;][klnn-	N/A	Cantidad	amarillo			
		I	I	I	I	V			
		.;][ap	.;][klnn-	N/A	N/A	amarillo			
		V	I	I	I	I			
tematizació	.;][klnn-	N/A	N/A	N/A					
V	V	I	I	I					
tematizació	Se realiza	N/A	N/A	N/A					
V	V	V	I	I					
tematizació	Se realiza	Edificios	N/A	N/A					
EC 1.3: Dejar campos obligatorios en blanco	El usuario selecciona la funcionalidad "Tematizaciones", luego elige la opción adicionar y elige la tematización estilo proporcional, completa los campos del formulario y presiona el botón adicionar.	V	V	V	V	I	El sistema muestra un formulario para iniciar sesión, otro para autenticar usuario, posteriormente muestra la opción para ir al módulo de administración, una vez en la vista administración, el usuario elige la opción tematizaciones, luego el sistema muestra el listado de tematizaciones, el usuario elige la opción adicionar y se despliega una lista con los tipos de tematizaciones, el usuario elige la opción "Estilo proporcional" luego muestra un formulario para introducir los datos de la tematización. Si el usuario deja algún campo en blanco muestra un mensaje de error que diga "Datos incompletos".	1. Seleccionar la opción "Iniciar Sesión". 2. Autenticar usuario con el rol que posee los permisos de administración y presionar el botón "Entrar". 3. Seleccionar la opción "Administración". 4. Seleccionar la opción "Tematizaciones". 5. Seleccionar la opción "Adicionar". 6. Seleccionar la opción "Estilo proporcional".	
		tematizació	Se realiza	Edificios	Cantidad	N/A			
		V	V	V	I	V			
		tematizació	Se realiza	Edificios	N/A	amarillo			
		V	V	I	V	V			
		tematizació	Se realiza	N/A	Cantidad	amarillo			
		V	I	V	V	V			
		tematizació		Edificios	Cantidad	amarillo			
		I	V	V	V	V			
			Se realiza	Edificios	Cantidad	amarillo			
		I	I	V	V	V			
				Edificios	Cantidad	amarillo			
		I	I	I	V	V			
				N/A	Cantidad	amarillo			
		I	I	I	I	V			
		N/A	N/A	amarillo					
V	I	I	I	I					
tematizació		N/A	N/A	N/A					
V	V	I	I	I					
tematizació	Se realiza	N/A	N/A	N/A					
V	V	V	I	I					
tematizació	Se realiza	Edificios	N/A	N/A					
I	I	I	I	I					
		N/A	N/A	N/A					
EC 1.4: Opción de Cancelar.	El usuario selecciona la funcionalidad	N/A	N/A	N/A	N/A	N/A	El sistema muestra un formulario para iniciar sesión, otro para autenticar	1. Seleccionar la opción "Iniciar Sesión".	