

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.**

TÍTULO: Módulo de impresión de mapas para la Plataforma GeneSIG.

AUTOR: Lázaro Dyron Delgado de la Cruz

TUTORES: MsC. Gerdys Ernesto Jiménez Moya, Ing. Grethell Castillo Reyes

COTUTOR: Ing. Adrián Gracia Águila.

La Habana, 2015

“Año 57 de la Revolución”

Declaración de Autoría



Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría en la Habana a los ____ días del mes ____ del año _____.

Lázaro Dyron Delgado

Firma del Autor

Grethell Castillo Reyes

Firma del Tutor

Gerdys Jiménez

Firma del Tutor





"En este futuro brillante, no puedes olvidar tu pasado"

Bob Marley

Datos de Contacto



Datos de Contacto

TUTOR: MsC. Gerdys Ernesto Jiménez Moya (email: gejimenez@uci.cu)

TUTOR: Ing. Grethell Castillo Reyes (email: gcreyes@uci.cu)

Agradecimientos

A mis padres, porque a pesar de mi forma de actuar y de todos los errores que he cometido nunca han dejado de sentirse orgullosos de mí.

A mi tía Yuleisi, por tenerme siempre presente.

A mi tío Carlos, por la actitud que muestra cuando se trata de mí.

A mi primo Papo, por toda la ayuda.

A mi novia, por todo lo demostrado en ocho años.

A todas aquellas personas en la Universidad que sin necesidad de mencionarlas saben que les estoy agradecido.

A Yoan Mandina, Alain León y Yampier Medina por la ayuda.

A mi tutora Grethell Castillo, por ayudarme a tal punto que no tengo palabras para agradecerle.

Dedicatoria

A mis Padres.

A mis hermanos.

A mis abuelos Eusebio, Iberis y Regla.

A mi primo Adrián.

A mis "colegas" de Mariano, quienes dijeron en más de una ocasión: "...tu sabes que estudiar no es pa' nosotros, salte ya de ese canal..."

Resumen

En la UCI (Universidad de las Ciencias Informáticas) se desarrolló una plataforma para agilizar el desarrollo futuro de SIG (Sistemas de Información Geográfica) en la WEB, partiendo de la reutilización de componentes y funcionalidades. Actualmente, la plataforma posee un componente para exportar al formato PDF (Formato de Documento Portable, del término en inglés *Portable Document Format*) un área seleccionada del mapa que se está visualizando. Dicho componente tiene varias limitaciones que traen como consecuencia que, como resultado final no se genere un mapa de impresión personalizado, que además carece de una descripción completa y detallada de objetos espaciales que en él se manipulan, esto impide obtener salidas similares a los mapas convencionales impresos. Todo esto, a su vez, afecta el análisis realizado sobre la información geoespacial, ya que el componente no brinda flexibilidad en cuanto a la inclusión y configuración de elementos en el mapa que se está consultando. Por esta razón, la presente investigación tuvo como objetivo principal desarrollar un módulo para el diseño flexible de mapas de impresión en la Plataforma GeneSIG. El diseño flexible consiste en brindar a los usuarios la posibilidad de modificar a fondo todos y cada uno de los elementos que incluya en el mapa que imprimirá posteriormente, alcanzando de esta forma un mayor nivel de personalización en el diseño realizado. Se llegó a la conclusión de que mediante la elaboración del mencionado módulo de impresión se habilitó un servicio que puso en manos de los usuarios la posibilidad de obtener imágenes en formato PDF del mapa o la zona del mapa determinada tras haber especificado los parámetros de impresión deseados.

Palabras claves: GeneSIG, impresión, mapa, objetos espaciales.

Abstract

In the University of the Computer Sciences a platform was developed to speed up the future development of GIS (Geographic Information Systems) in the WEB, starting from the use of components and functionalities. At the moment, the platform possesses a component to export to the format PDF (Portable Document Format) a selected area of the map that is visualizing. This component has several limitations that produce as a final result a personalized impression map is not generated, it also lacks a complete and detailed description of space objects that are manipulated in him, and this prevents to obtain similar exits to conventional maps. All this, in turn, affects the analysis carried out about the space information, since the component doesn't offer flexibility as for the inclusion and configuration of elements in the map that is consulting. For this reason, the present investigation had as main objective to develop a component for the flexible design of impression maps in the Platform GeneSIG. The flexible design consists on offering the users the possibility to modify thoroughly all and each one of the elements that includes in the map that will print later on, reaching this way a bigger personalization level in the carried out design. You reached the conclusion that by means of the elaboration of the mentioned impression module a service was enabled that put in the users' hands the possibility to obtain images in format PDF of the map or the area of the map determined after having specified the wanted impression parameters.

Keywords: *GeneSIG, impression, map, space objects.*

Índice de Contenido

| | |
|--|-----------|
| Introducción..... | 1 |
| Estructura del documento..... | 5 |
| 1 Fundamentos de la impresión de mapas..... | 6 |
| 1.1 Conceptos relacionados con el dominio del problema..... | 6 |
| 1.1.1 Mapa..... | 6 |
| 1.1.2 Leyenda..... | 6 |
| 1.1.3 Escala gráfica..... | 7 |
| 1.1.4 Mapa de referencia..... | 7 |
| 1.1.5 Sistemas de Información Geográfica..... | 7 |
| 1.1.6 Dato espacial..... | 8 |
| 1.1.7 Mapas de Impresión..... | 8 |
| 1.2 Diseño de Mapas para Impresión..... | 8 |
| 1.2.1 Principales elementos de un Mapa de Impresión..... | 9 |
| 1.3 Situación Problemática..... | 10 |
| 1.4 Análisis de Soluciones Existentes..... | 10 |
| 1.5 Plataforma GeneSIG..... | 13 |

| | | |
|--------------|--|----|
| 1.5.1 | <i>Especificaciones técnicas</i> | 14 |
| 1.5.2 | <i>Plugins y Coreplugins</i> | 14 |
| 1.5.3 | <i>Infraestructura</i> | 15 |
| 1.5.4 | <i>Recursos</i> | 15 |
| 1.5.5 | <i>Servicios</i> | 16 |
| 1.5.6 | <i>Integración con plataformas externas</i> | 16 |
| 1.6 | Herramientas y tecnologías | 16 |
| 1.6.1 | <i>Metodología de desarrollo de software</i> | 16 |
| 1.6.2 | <i>Framework ExtJS 3.1</i> | 18 |
| 1.6.3 | <i>IDE utilizado</i> | 19 |
| 1.6.4 | <i>Visual Paradigm 8.0 Enterprise Edition</i> | 19 |
| 1.6.5 | <i>Lenguaje de Modelado</i> | 19 |
| 1.6.6 | <i>Lenguaje del Servidor</i> | 20 |
| 1.6.7 | <i>Servidor Web</i> | 20 |
| 1.6.8 | <i>Servidor de Mapas</i> | 21 |
| 1.6.9 | <i>Openlayers</i> | 21 |
| 1.7 | Conclusiones parciales | 22 |
| 2 | Descripción de la solución propuesta | 23 |

| | | |
|--------------|--|----|
| 2.1 | Modelo de dominio | 23 |
| 2.1.1 | <i>Descripción del modelo de dominio</i> | 24 |
| 2.1.2 | <i>Glosario de términos del dominio</i> | 25 |
| 2.2.1 | <i>Requisitos no funcionales</i> | 26 |
| 2.2.2 | <i>Requisitos funcionales</i> | 28 |
| 2.3 | Conclusiones parciales | 36 |
| 3 | Implementación y validación | 37 |
| 3.1 | Arquitectura de Software | 37 |
| 3.1.2 | <i>Patrones de Diseño</i> | 38 |
| 3.2 | Elementos del Diseño de Clases | 39 |
| 3.3 | Modelo de Implementación | 40 |
| 3.4 | Modelo de Despliegue | 41 |
| 3.5 | Modelo de Pruebas | 42 |
| 3.5.1 | <i>Prueba de aceptación</i> | 43 |
| 3.5.2 | <i>Prueba de Caja Negra</i> | 43 |
| 3.5.3 | <i>Prueba de Caja Blanca</i> | 46 |
| 3.6 | Conclusiones Parciales | 48 |
| | Conclusiones Generales | 49 |

Índice de Contenido

| | |
|---|----|
| Recomendaciones | 50 |
| Glosario de Términos | 51 |
| Referencias Bibliográficas | 53 |
| Anexos | 56 |

Índice de Tablas

| | |
|--|-----------|
| Tabla 1.1 Elementos principales de un mapa de impresión..... | 9 |
| Tabla 1.2 Comparación de soluciones existentes..... | 12 |
| Tabla 2.1 RF 8: Adicionar Etiqueta de Texto:..... | 29 |
| Tabla 2.2 RF 10: Adicionar Escala:..... | 32 |
| Tabla 3.1: Secciones a probar en el RF Adicionar Etiqueta de Texto..... | 44 |
| Tabla 3.2: Secciones a probar en el RF Deshacer..... | 44 |
| Tabla 3.3: Secciones a probar en el RF Adicionar Imagen..... | 45 |
| Tabla 3.4: Secciones a probar en el RF Rehacer..... | 45 |

Índice de Figuras

| | |
|--|-----------|
| Fig. 1.1: Estructura física de GeneSIG. | 15 |
| Fig. 1.2: Fases del ciclo de vida del proyecto. | 18 |
| Fig. 2.3: Diagrama de clases del dominio | 24 |
| Fig. 2.4: Prototipo de IU Propiedades del elemento/Etiqueta | 32 |
| Fig. 2.5: Prototipo de IU Propiedades del elemento/Escala | 36 |
| Fig. 3.6: Diagrama de Clases del Diseño | 40 |
| Fig. 3.7: Diagrama de Componentes | 41 |
| Fig. 3.8: Diagrama de Despliegue | 42 |
| Fig. 3.9: Grafo de flujo de datos de la función crear imagen. | 47 |
| Fig. 10 Anexo 1: Crear Imagen | 56 |

Introducción

Los mapas digitales son ampliamente superiores a los físicos, la existencia de la web y el amplio uso de las Tecnologías de la Informática y las Comunicaciones (TIC) hacen posible que se encuentren a disposición de los usuarios a lo largo del mundo. De esta forma dichos usuarios se benefician de las ventajas que trae consigo consultar los mapas digitales, ya que están enriquecidos con una cantidad de información increíble. Los mapas digitales están relacionados con la tecnología espacial, esta tecnología está presente en situaciones cotidianas como es el caso de revisar el pronóstico meteorológico, ver televisión por la antena parabólica o utilizar un dispositivo GPS (Sistema de Posicionamiento Global, del término en inglés *Global Positioning System*).

“La tecnología espacial nos da las nuevas maneras de integrar, acceder y analizar datos, permite a las organizaciones trabajar más eficazmente y tomar bien las decisiones. El mapa digital tiene una serie amplia de soluciones espaciales para ayudar a las organizaciones a aprovechar el poder de la tecnología espacial”.(Product, 2013).

Las personas han sabido sacar provecho de las funcionalidades de estos mapas, los usan a diario y no solo en empresas que necesitan de apoyo digital para su funcionamiento, los usuarios comunes los utilizan con regularidad (gracias a la integración de los mapas con la tecnología móvil) para funciones simples pero importantes. Dichas funciones pueden ser acordar una cita en un lugar señalado, enviar su ubicación a un amigo que intenta localizarlo o incluso encontrar en las cercanías locales que brinden algún tipo de servicio, dígame restaurantes, bares, cines, gasolineras.

Hay que resaltar que estos mapas son ampliamente útiles, su uso está presente en numerosos campos científicos que se nutren de las funcionalidades que estos poseen. A partir del año 2009, los campos de la ciencia que utilizan tecnología de mapas digitales incluyen geología, ingeniería, arquitectura, topografía, minería, silvicultura, medio ambiente, y arqueología.(E-CENTRO, 2014).

Existen en la actualidad sistemas computacionales que facilitan a los usuarios el trabajo con la información geográfica y que desde su aparición constituyeron elementos definitivos en el desarrollo de la cartografía digital, ya que permiten a los usuarios realizar consultas y configuraciones sobre la información visualizada. Esta es una razón por la que estos sistemas están ganando popularidad entre las personas,

organizaciones y sectores que precisan de información geográfica para dar solución a alguna tarea determinada o para auxiliarse en la toma de alguna decisión.

Los sistemas mencionados anteriormente son conocidos como SIG y se pueden definir como *“un conjunto de hardware, software, datos geográficos y personal capacitado, organizados para capturar, almacenar, consultar, analizar y presentar todo tipo de información que pueda tener una referencia geográfica. Un SIG es esencialmente una base de datos espacial, lo que le otorga una cualidad incomparable en el desarrollo de análisis enfocados a resolver problemas reales que afectan el espacio geográfico”*.(Geográfica, 2013).

Las características mencionadas anteriormente hacen que los SIG sean ampliamente usados a nivel mundial. *“Los SIG tienen aplicaciones en disímiles campos como lo son las evaluaciones ambientales, emergencias médicas, producción cartográfica y el estudio de la presencia de recursos en territorios determinados. Lo anterior se hace posible porque estos sistemas están capacitados para el análisis espacial de la información, la creación de mapas y la edición de los datos de estos”*.(Antunez, 2012).

Lo anterior constituye un claro factor de por qué Cuba necesita de esta tecnología y del conocimiento y dominio de la misma. En la UCI a lo largo de los años se ha adquirido gran experiencia en el desarrollo de grandes proyectos contribuyendo de una forma u otra en la formación de profesionales con grandes conocimientos en áreas de la informática. Uno de los aportes más significativo de la UCI a este tema es el desarrollo de la Plataforma Soberana GeneSIG. Con esta plataforma se persigue fortalecer la experiencia en la realización de SIG y abrir un espacio sólido en el mercado de aplicaciones de esta rama, reutilizando los componentes y funcionalidades para personalizar los productos en cualquier negocio que lo requiera.

“La Plataforma GeneSIG es un producto encaminado a realizar la representación y análisis geoespacial de información geográfica y su estructura arquitectónica permite personalizar sus funcionalidades para la mayoría de los negocios que lo requieran. Puede ser considerado como un SIG único y extensible, basado en estándares OpenGIS que incluye funcionalidades operativas de las aplicaciones de esta tecnología”.(Pantoja Zaldívar, 2011).

Sin embargo, incluso cuando la tecnología pone estos avances en manos de las personas, no es razón para deshacerse de los mapas tradicionales, los cuales se pueden definir como la representación geográfica de la Tierra o parte de ella en una superficie plana. (RAE, 2010). La razón por la que sería

conveniente continuar su uso es que este tipo de mapas también brinda ventajas, ejemplo de ello se ve en el sector educativo, donde se hace factible el uso de atlas para impartir clases en lugares que no disponen de suficientes medios electrónicos. Esto no es aplicable solo al sector educativo, en el sector militar la utilización de los mapas tradicionales tampoco ha sido abandonada, ya que los mismos constituyen un elemento clave en el desarrollo de tácticas de combate, estrategias y misiones de reconocimiento.

Una vez alcanzado este punto se puede deducir que la mejor forma de sacar partido al trabajo con información geográfica sería contando con una combinación entre las ventajas que brindan los mapas digitales y los tradicionales.

Dentro de la cartografía digital, existe un tipo de mapa conocido como mapa de impresión que permite al usuario la obtención de la información visualizada en formato duro. El mapa de impresión proporciona el diseño de capas y el aumento de las capacidades de impresión. Permite agregar los elementos, etiquetas de texto, imágenes, leyendas, escala, formas básicas, flechas, tablas de atributos y marcos HTML (Lenguaje de Marcas de Hipertexto, del término en inglés *Hypertext Marks Language*). También permite clasificar según el tamaño, agrupar, alinear y posicionar cada elemento, además de ajustar las propiedades para crear las capas. El diseño puede imprimirse o puede exportarse a formatos de imagen. (Documentación QGIS 2.0, 2006).

Actualmente, la plataforma GeneSIG cuenta con un componente para exportar al formato PDF un área seleccionada del mapa que se está visualizando, permitiendo añadir al documento de impresión elementos tales como: la leyenda, la escala gráfica y el mapa de referencia correspondiente. Sin embargo, existen limitaciones que afectan el resultado final y por ende los análisis para los que son objetos los documentos generados. Estas limitaciones se reflejan en el desempeño de funciones relacionadas de forma general con la inclusión y configuración de elementos en el mapa que se está consultando. De forma específica el componente actual no permite ubicar el mapa en el lugar deseado en el documento, no permite la inclusión de elementos descriptivos ni permite imprimir otra sección del mapa que no sea la que se esté visualizando en ese momento. Todo esto constituye un obstáculo para el enriquecimiento de la información a imprimir.

Estas limitaciones traen como consecuencia que, como resultado final no se genere un mapa de impresión personalizado, que además carece de una descripción completa y detallada de los objetos espaciales que en él se manipulan, lo que impide obtener salidas similares a los mapas convencionales

impresos. De no obtener dichas salidas no se garantiza la efectividad de las decisiones tomadas en base a los documentos generados ya que los usuarios no contarían con un documento con todos los detalles gráficos necesarios para el correcto trabajo sobre información geográfica. Todo esto, a su vez, afecta el análisis realizado sobre la información geoespacial y por tanto influye de forma negativa sobre los resultados provenientes de dichos análisis. Conocido esto se puede determinar como **problema de la investigación** que, la construcción adecuada de mapas en formato duro es una tarea compleja y de un elevado nivel de perfección, por lo que hacerlo de manera manual o estática conlleva por lo general a no cumplir las exigencias establecidas. El **objeto de estudio** seleccionado es el proceso de diseño de mapas de impresión en SIG, por otra parte el **campo de acción** es el proceso de diseño de mapas de impresión en SIG en la Web. Se hace necesario el planteamiento de un **objetivo general** con el fin de dar solución al problema planteado, dicho objetivo es desarrollar un módulo que permita el diseño flexible de mapas para impresión en la Plataforma GeneSIG.

Para darle cumplimiento al objetivo antes planteado se proponen las siguientes **preguntas científicas**:

1. ¿Cuáles son los principales SIG que cuentan con módulos para el diseño de mapas de impresión?
2. ¿Qué características imprescindibles debe tener el módulo para el diseño de mapas de impresión?
3. ¿Cuáles son las pautas a seguir para definir el diseño del sistema a desarrollar?
4. ¿Cómo enfocar la construcción del módulo hacia las necesidades de la plataforma GeneSIG?
5. ¿Cómo validar la solución construida con el fin de garantizar el correcto funcionamiento de la misma?

El proceso de investigación viene guiado por las siguientes **tareas de la investigación**, las cuales han sido definidas con el fin de dar cumplimiento al objetivo anteriormente mencionado:

1. Identificar las variantes de solución existentes en el tratamiento del problema.
2. Realizar el análisis y diseño del módulo de impresión de mapas para la plataforma GeneSIG.
3. Implementar el módulo de impresión para la plataforma GeneSIG.
4. Realizar las pruebas las pruebas de software al módulo de impresión para la plataforma GeneSIG.

Una vez llevada a cabo la realización de estas tareas se estima una serie de **posibles resultados**:

1. El diseño y la implementación de un módulo que permita el diseño flexible de mapas para impresión desde la plataforma GeneSIG.
2. La documentación técnica asociada al desarrollo del módulo.

La investigación científica desempeñada para dar cumplimiento a la problemática tiene sus bases en varios **métodos investigativos** definidos en (Ramírez Gonzalez, 2008), los cuales son: el **análisis documental** y **entrevista** que clasifican como métodos empíricos, el primero para el estudio de carácter literario que se lleva a cabo en la investigación y el segundo para obtener respuestas verbales a las interrogantes planteadas sobre el problema propuesto. Para un análisis más exhaustivo del material bibliográfico empleado se utilizan los siguientes métodos teóricos: **analítico – sintético**, que permite sintetizar conceptos e identificar elementos claves para dar solución a la problemática y por último, el método **histórico – lógico**, que se usa con el objetivo de realizar comparaciones con resultados anteriormente alcanzados.

Estructura del documento

El presente documento se encuentra dividido en tres capítulos:

Capítulo 1. Fundamentos de la impresión de mapas: se lleva a cabo la descripción de los elementos asociados a la investigación, de esta forma se brindan elementos esenciales que ayudan a lograr un mejor entendimiento de la solución propuesta. También se justifica la elección de las tecnologías y herramientas usadas durante el proceso de desarrollo y se realiza un análisis detallado de la situación problemática y el objeto de estudio de la investigación.

Capítulo 2. Descripción de la solución propuesta: en este capítulo se tienen en cuenta las indicaciones de la metodología de desarrollo utilizada para realizar la descripción de los artefactos relacionados con el levantamiento de requisitos y la modelación del dominio.

Capítulo 3. Implementación y validación: durante este capítulo se trata todo lo relacionado con los procesos de diseño, implementación y validación de la solución propuesta. Además se realizan un conjunto de pruebas a fin de comprobar el correcto funcionamiento del sistema.

Capítulo 1

1 Fundamentos de la impresión de mapas

En este capítulo se explican los conceptos fundamentales relacionados con el dominio del problema planteado con el objetivo de apoyar a la comprensión del alcance de la investigación desarrollada. También se brinda una descripción explícita de la situación problemática existente para un mejor entendimiento de la necesidad vigente. Además se presentan las tecnologías y herramientas seleccionadas para el desarrollo de la solución, justificando su elección. También se lleva a cabo un análisis detallado de los referentes teóricos y prácticos que se manifiestan en la realización del presente trabajo con el fin de esclarecer su objeto de estudio.

1.1 Conceptos relacionados con el dominio del problema

1.1.1 *Mapa*

Un mapa es un dibujo o esquema que representa una cierta franja de un territorio sobre una determinada superficie de dos dimensiones. Existe una gran variedad de tipos de mapas según lo que nos muestren y según el objetivo que persigan cumplir. (Definiciones, 2014).

Por otra parte, según (Antunez, 2012) un mapa es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos.

1.1.2 *Leyenda*

En un mapa se denomina leyenda a la explicación que se provee sobre los símbolos y colores que se presentan en el mapa. En la leyenda se suele dibujar cada uno de los símbolos utilizados en el mapa y se coloca una explicación sobre su significado. (RAE, 2010).

En la leyenda se reflejan todos los símbolos que se utilizan en el mapa, y proporciona la clave para la interpretación de los mismos. Este elemento en particular es de suma importancia a la hora de consultar

un mapa ya que no comprender la información plasmada en dicho mapa tendría un alto grado de complejidad. (Definiciones, 2014).

1.1.3 Escala gráfica

La escala es un gráfico consistente en una línea o barra graduada que indica la longitud correspondiente en el plano. (RAE, 2010).

En cartografía, se utiliza para señalar una relación, ya sea numérica o gráfica, entre la realidad y el dibujo. Por ejemplo el número de kilómetros reales entre un lugar y otro de la superficie terrestre están representados en un mapa en dimensiones menores, pero que equivalen a las reales. En (Antunez, 2012) se define la escala gráfica como la relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre.

1.1.4 Mapa de referencia

Los mapas de referencia, también conocidos como los mapas de navegación, muestran diversos aspectos de las ubicaciones. Algunas de ellas ofrecen la información o los datos necesarios para trazar las rutas de viaje extra. (mapas, 2014).

Por otra parte, en los SIG se podrían definir de una forma más explícita como un mapa diseñado para mostrar donde se encuentra algún elemento geográfico en relación con los otros elementos que conforman el mapa visualizado.(Esri, 2015).

1.1.5 Sistemas de Información Geográfica

Un SIG es un sistema compuesto por hardware, software, procedimientos y equipo humano para capturar, manejar, manipular, transformar, analizar y modelar datos geográficos, permitiendo representar los objetos del mundo real en términos de posición, atributos y de interrelaciones espaciales , con el objeto de analizar estos datos y de resolver problemas de gestión y planificación. Un SIG debe tener como funciones la entrada y salida de datos, la gestión de datos (modificar, eliminar, etc.) y funciones de análisis y consulta. (Dávila Martínez).

Un SIG es un software específico que permite a los usuarios crear consultas interactivas, integrar, analizar y representar de una forma eficiente cualquier tipo de información geográfica referenciada asociada a un territorio, conectando mapas con bases de datos.

De forma general se puede concluir en que un SIG es todo aquel software o programa que al relacionarse con la información brinda como resultado la obtención de datos relacionados con el espacio físico.

1.1.6 Dato espacial

Los datos espaciales representan información sobre la ubicación física y la forma de objetos geométricos. Estos objetos pueden ser ubicaciones de punto u objetos más complejos como países, carreteras o lagos. Su nombre se debe a que poseen una ubicación en el espacio, razón por la que tienen gran importancia en el campo de la ubicación geográfica. Cabe resaltar que este tipo de dato se define por elementos esenciales como lo son la relación, la localización y la descripción. (Castillo Reyes, 2012).

1.1.7 Mapas de Impresión

Si un usuario se encuentra visualizando un mapa, debería ser capaz de imprimirlo o exportarlo a un documento. Los archivos de mapas de los SIG no son imágenes. Estos archivos más bien guardan el estado del programa, con referencias a todas las capas, etiquetas y colores. Así que para alguien que no tenga los datos o el mismo programa SIG, el archivo del mapa será inútil. Afortunadamente, los SIG pueden exportar el archivo del mapa a un formato que cualquier ordenador pueda leer, así como imprimir el mapa si se tiene una impresora conectada. La realización de estas acciones es labor del módulo o compositor de impresión. (Documentación QGIS 2.0, 2006).

1.2 Diseño de Mapas para Impresión

A lo largo del mundo el trabajo sobre la información geográfica ha crecido considerablemente. Incluso cuando la mayor parte de dicha información se encuentra en formato digital es de suma importancia que los usuarios puedan contar con la misma en formato físico para su posterior utilización en circunstancias en que no se cuente con medios tecnológicos para acceder a dicha información. Para dar solución a esto, los SIG incluyen funcionalidades de impresión que permiten a los usuarios llevar la información visualizada a formato PDF, para luego realizar la impresión de la misma. Esta operación es llevada a cabo

por un módulo específico dentro de los SIG, dicho módulo es conocido como módulo de impresión o compositor de mapas de impresión.

Los módulos o compositores de impresión presentan características específicas en dependencia de como hayan sido configurados. Además de encargarse de la impresión de la información, posee funcionalidades que brindan a los usuarios la posibilidad de realizar modificaciones sobre la información que desean imprimir con el fin de enriquecer la misma.

1.2.1 Principales elementos de un Mapa de Impresión

Con el objetivo de obtener un acabado del software que cumpla con los requisitos necesarios que aseguren el correcto funcionamiento del mismo se realiza una entrevista a las personas involucradas con el proyecto para conocer su opinión con respecto al tema.

Elementos tales como la Leyenda, Escala Gráfica, Mapa de Referencia, Textos Descriptivos y Símbolos fueron descritos por estas personas como elementos claves con que debe contar el producto final. Además resalta la importancia que tiene para el usuario la presencia de opciones configurables de cada uno de estos elementos.

A continuación se muestra una tabla con la descripción de los principales elementos con los que debe contar un mapa para impresión:

Tabla 1.1: Elementos principales de un mapa de impresión.

| Elementos | Función |
|--------------------|---|
| Leyenda | Elemento que permite a los usuarios comprender la información plasmada en el mapa. |
| Escala Gráfica | Elemento que muestra a los usuarios información geográfica en dimensiones menores, pero que equivalen a las reales. |
| Mapa de Referencia | Elemento que brinda a los usuarios información adicional sobre el mapa consultado. |

| | |
|---------------------|---|
| Textos Descriptivos | Elemento que permite a los usuarios crear etiquetas de texto para la identificación de los elementos presentes en el mapa consultado. |
| Símbolos | Elemento que permite a los usuarios enriquecer la información consultada para su posterior impresión. |

1.3 Situación Problemática

En la Universidad de las Ciencias Informáticas se desarrolló una plataforma para agilizar el desarrollo de SIG en la web, partiendo de la reutilización de componentes y funcionalidades. Actualmente, la plataforma posee un componente para exportar al formato PDF un área seleccionada del mapa que se está visualizando. Este componente permite además añadir al documento de impresión elementos tales como: la leyenda, la escala gráfica y el mapa de referencia correspondiente. Sin embargo, existen limitaciones que afectan el resultado final y por ende los análisis para los que son objetos los documentos generados.

Estas limitaciones se reflejan en el desempeño de funciones relacionadas de forma general con la inclusión y configuración de elementos en el mapa que se está consultando, constituyendo un obstáculo para el enriquecimiento de la información a imprimir.

Esto trae como consecuencia que, como resultado final no se genere un mapa de impresión personalizado, que además carece de una descripción completa y detallada de los objetos espaciales que en él se manipulan, lo que impide obtener salidas similares a los mapas convencionales impresos. De no obtener dichas salidas no se garantizaría la efectividad de las decisiones tomadas en base a los documentos generados ya que los usuarios no contarían con un documento con todos los detalles gráficos necesarios para el correcto trabajo sobre información geográfica. Todo esto, a su vez, afecta el análisis realizado sobre la información geoespacial y por tanto influye de forma negativa sobre los resultados provenientes de dichos análisis.

1.4 Análisis de Soluciones Existentes

Actualmente, existe un conjunto numeroso de SIG que incluyen variantes para el diseño de mapas de impresión, tal es el caso de QuantumGIS, Mapinfo, Kosmo y ArcGIS:

Kosmo cuenta con un módulo que permite modificar las Opciones de Impresión por defecto, dichas opciones son accesibles a través de una ventana que brinda la posibilidad de mantener el tamaño de los elementos - opción que permite no modificar el tamaño de los elementos al modificar la configuración de página – y la de mantener la proporción existente entre el objeto y el papel (opción por defecto) que mantiene la relación existente entre el ancho y alto de página y el alto y ancho del componente. También tiene la herramienta Configuración de Página que permite modificar los parámetros de configuración de la página donde se va a imprimir el mapa. Una vez hechas las modificaciones, accediendo a la herramienta Imprimir se obtiene una ventana que permite seleccionar la impresora que se utilizará, así como el número de copias a realizar etc... Es importante resaltar que Kosmo permite guardar mapas como ficheros XML (Lenguaje de Marcas Extensible) lo que hace posible guardar un mapa como plantilla para usarlo en otros proyectos. (SAIG, 2009).

Por otra parte QuantumGIS ha implementado la Rotación de Elemento lo que implica que todo tipo de elemento en el diseñador de impresión se puede rotar, incluyendo barra de escala, tablas y leyendas. Por ejemplo, puede rotar una etiqueta en el diseñador de manera que se ajuste mejor a su diseño de página. También se han realizado mejoras a la herramienta Añadir Escala en el Diseñador y Regla, lo que implica que las apariencias de las reglas hayan sido mejoradas mediante el ajuste de la escala lógica y al añadir divisiones más pequeñas de la regla. Por último QuantumGIS permite la Generación de Archivo Mundial, que permite en el diseñador, crear mapas georreferenciados. (Documentación QGIS 2.0, 2006).

Sobre ArcGIS se conoce que ArcGIS 10.1 para servidores incluye un servicio de geoprocesamiento llamado *PrintingTools*. Cuando se desarrollan aplicaciones web, se puede invocar el servicio *PrintingTools* para imprimir y obtener una imagen de alta calidad cartográfica. Este es un flujo de trabajo avanzado que no es necesario si está utilizando la API web. *PrintingTools* viene con un conjunto predefinido de diseños de mapa para elegir. Si desea utilizar su propio diseño de mapa, puede publicar su propio servicio para la impresión del mapa web similar a *PrintingTools* y elegir su propia carpeta de diseños. (Esri, 2013).

MapInfo brinda una GUI (*Graphic User Interface*) sencilla donde el usuario puede interactuar con el sistema. Al abrir la ventana correspondiente a la configuración del funcionamiento del módulo se presentan las opciones para escoger las vistas deseadas de la información visualizada, pudiendo escoger entre una vista o todas las vistas abiertas (esta opción es particularmente útil para la confección de mapas

en mosaico o mural). Seguido a esto se obtiene una página con múltiples elementos: mapa, título, leyenda, buscador y gráfico. Seguido se muestra una ventana equipada para realizar configuraciones a la impresora (impresora a utilizar, tipo y tamaño de impresión, calidad de impresión). También en esta ventana se puede seleccionar la escala en que se desea hacer la impresión y reorganizar los elementos, se puede incluso redefinir el tamaño de la página. (Arce Mesén, 2000).

Tras realizar un análisis de las soluciones existentes, se ha hecho una selección de funcionalidades que los mismos poseen. La razón es que se considera que podrían implementarse algunas similares para el módulo que se quiere desarrollar con el fin de mejorar su funcionamiento.

Estos sistemas han sido también comparados atendiendo a otras características:

Libre/Propietario: Expresa el tipo de software del sistema analizado.

Nivel de Requisitos: Este aspecto brinda clasificaciones de tipo bajo, medio o alto en dependencia de las propiedades que debe poseer la estación de trabajo donde se ejecute el sistema.

Compatibilidad: Muestra los Sistemas Operativos donde se puede ejecutar el sistema.

Licencia: Se muestran las licencias pertenecientes a cada uno de los sistemas analizados.

Tabla 1.2: Comparación de soluciones existentes.

| SIG | ArcGIS | MapInfo | Quantum GIS | Kosmo |
|----------------------------|----------------------------------|--|--|-------------------|
| Libre/Propietario | Propietario | Propietario | Libre | Libre |
| Nivel de Requisitos | Medio | Bajo | Bajo | Bajo |
| Compatibilidad | Windows, Android, Mac OS X | Microsoft Windows 9X, Microsoft Office. | GNU/Linux, BSD, Unix, Mac OSX, Windows y Android | Windows/GNU/Linux |
| Licencia | Comercial de tres niveles | Comercial | GNU GPL | GNU GPL |

Tras concluir el análisis realizado a dichos sistemas se llega a la conclusión de que QuantumGIS es el que posee la mayor cantidad de funcionalidades adaptables a la solución propuesta debido a que incluye en su diseñador de mapas la posibilidad de trabajar con múltiples elementos y se han añadido configuraciones para añadir y redimensionar varios elementos simultáneamente.

Por esta razón se han desarrollado las principales funcionalidades de dicho sistema basadas en las existentes en QuantumGIS. De forma más específica se han tomado las funcionalidades Adicionar Imagen del Mapa, Adicionar Imagen, Adicionar Etiqueta de Texto, Adicionar Leyenda y Adicionar Escala. Además de las funcionalidades principales se han desarrollada otras, tales como Deshacer Cambios y Rehacer Cambios. Vale añadir que QuantumGIS también posee un diseño más completo que permite a los usuarios realizar modificaciones a los elementos añadidos de forma más detallada.

1.5 Plataforma GeneSIG

La plataforma GeneSIG es un sistema de información geográfica desarrollado en la UCI, a continuación se brinda una descripción que permite conocer los detalles de dicho sistema:

GeneSIG es un sistema de software con una arquitectura basada en componentes. Estos son las unidades de modelado, diseño e implementación. Las interfaces están separadas de las implementaciones, y conjuntamente con sus interacciones son el centro de incumbencias en el diseño arquitectónico. Los componentes soportan algún régimen de introspección, de modo que su funcionalidad y propiedades puedan ser descubiertas y utilizadas en tiempo de ejecución. Entonces se puede decir que las características principales son la modularidad, la reusabilidad y compatibilidad. (Antunez, 2012).

La plataforma GeneSIG al ser una aplicación enfocada a la web, brinda facilidades en cuanto al acceso por parte de los usuarios que pueden utilizar sus servicios desde cualquier ubicación, siempre y cuando posean una conexión a internet. Existen otros aspectos además de la portabilidad que constituyen ventajas de las aplicaciones web sobre las de escritorio, ejemplo de esto son: La ligereza que presentan este tipo de aplicaciones, como se accede a ellas a través de un navegador que no contiene el programa, no es necesario la presencia de grandes prestaciones en el ordenador para trabajar con ellas. También son fáciles de actualizar y mantener, además pueden ser muy seguras, dependiendo del desarrollador. Su

funcionalidad es independiente del sistema operativo instalado en el ordenador del usuario y no existe incompatibilidad de versiones, puesto que todos los usuarios trabajan con la misma.

1.5.1 Especificaciones técnicas

En la plataforma se siguen los principios de la Programación Orientada a Objetos (POO), está programada en PHP 5 y está enfocada al desarrollo de aplicaciones web en el mismo lenguaje de programación. La versión mínima de PHP (del término en inglés *Hypertext Preprocessor*) requerida para ejecutar GeneSIG es PHP 5.2. Por otra parte se utiliza JavaScript en su forma *client-side* y también ExtJS como framework para la implementación de la interfaz, aprovechando que brinda la posibilidad de realizar una interfaz dinámica que resulte más interesante a los usuarios que interactúen con el sistema.

Utiliza MapServer como servidor de cartografía digital, que provee cartografía a través de la red tanto en modo vectorial como con imágenes. Puede utilizarse para crear servicios de información basados en mapas dinámicos, imágenes satelitales, datos SIG o un conjunto de estos. Permite ejecutar una visualización sobre cualquier mapa publicado que esté accesible desde Internet. El software trabaja incorporando otras aplicaciones OpenSource las cuales en su conjunto ofrecen una solución tecnológica de alto nivel para el tratamiento de la información geográfica. Una característica muy importante es que MapServer puede funcionar sobre la mayor parte de versiones de Sistemas operativos, UNIX/LINUX, Microsoft Windows XP/NT/98/95 y hasta MacOS. Además se utiliza PostgreSQL como servidor de base de datos, ya que es un proyecto de software libre distribuido bajo licencia BSD (del término en inglés *Berkeley Software Distribution*), de gran estabilidad y creado con el aporte de varios colaboradores y auspiciantes a nivel mundial bajo los estándares de ANSI-SQL 92/99.(Antunez, 2012).

1.5.2 Plugins y Coreplugins

En GeneSIG los *Plugins* son paquetes modulares de archivos (clases PHP, plantillas HTML, imágenes) que se utilizan para llevar a cabo una acción especializada: el formato de mapa principal, la interfaz de las capas de la navegación, la exploración del mapa (zoom, paneo), las consultas, la autenticación y las interfaces de búsqueda.

Por otra parte los *Coreplugins* son considerados aquellos *Plugins* de "bajo nivel" que tienen una implicación de carácter vertical afectando la mayoría de los componentes de la plataforma. Estos llevan a cabo acciones como la manipulación del tamaño del mapa, herramientas de navegación, la selección de capas. (Antunez, 2012).

1.5.3 Infraestructura

La estructura física que compone la plataforma GeneSIG, en principio es la misma definida por el framework CartoWeb.

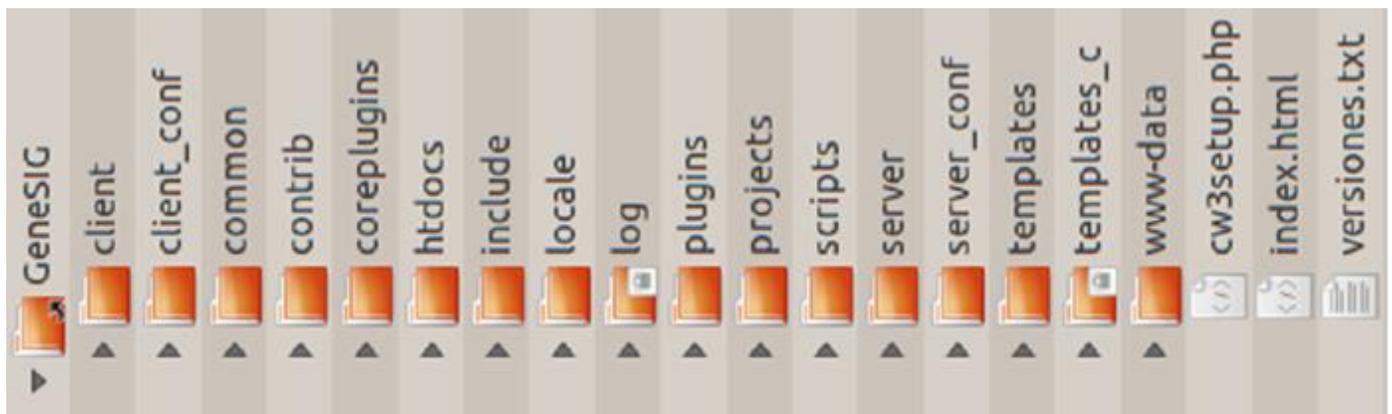


Fig. 1.1: Estructura física de GeneSIG.

La analogía que existe entre la estructura física de un *plugin* o *coreplugin* con un proyecto específico y a su vez con el framework en sentido general es algo intencionado, en función de minimizar el cúmulo de conceptos que intervienen en el desarrollo de software sobre esta plataforma.(Antunez, 2012).

1.5.4 Recursos

La plataforma GeneSIG posee una serie de recursos que implementan una gran cantidad de comportamientos que son generalizados y con una marcada tendencia a influir en los restantes módulos de forma horizontal, dichos recursos son los siguientes: Base, Layer, Location, Query, Auth, Priolus y Manipulación de Mensajes.(Antunez, 2012).

1.5.5 Servicios

Uno de los mecanismos de integración que se están potenciando en la plataforma GeneSIG es el basado en Servicio Web. Este podría interactuar con servicios remotos a través de los métodos Get y Post de HTTP (del término en inglés *HyperText Transfer Protocol*), pero SOAP (del término en inglés *Simple Object Access Protocol*) es mucho más robusto y flexible. Permitiendo el paso de parámetros y comandos entre clientes y servidores de HTTP, independientemente de las plataformas y aplicaciones existentes entre ambas partes. Los parámetros y los comandos se codifican utilizando XML (Lenguaje de Marcas Extensible, del término en inglés *Extensible Markup Language*). SOAP es un protocolo liviano, basado en lenguaje XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Sin embargo no define la aplicación, ni la semántica de implementación. (Antunez, 2012).

1.5.6 Integración con plataformas externas

Cada día aparecen nuevas tendencias que aumentan la complejidad de los entornos de las tecnologías de la información.

Los productos de software hoy en día se encuentran en el núcleo de las empresas y de esta forma siguen el mismo modelo de actividad que éstas se imponen, es decir, deben ser reactivos frente a los cambios. Con la integración se persigue poner en relación los diferentes sistemas para que funcionen de una forma heterogénea, permitiendo conseguir una solución global que responda a las necesidades del usuario. (Antunez, 2012).

1.6 Herramientas y tecnologías

Tras la definición de la propuesta de solución al problema planteado y una vez descritas las características principales de la Plataforma GeneSIG, se procede a la selección de las tecnologías y herramientas que se usan. El desarrollo de la solución se realiza haciendo uso de tecnologías y herramientas libres, en su mayoría definidas por la base arquitectónica de GeneSIG. En esta sección se justifica la elección de las mismas, brindando información detallada al respecto.

1.6.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software se utilizan con el fin de crear un diseño previo del camino a seguir para obtener un correcto producto final, razón por la cual es de suma importancia hacer uso de estas antes de comenzar con la implementación. Las metodologías de software establecen pautas a seguir, que de cumplirlas se estaría garantizando la calidad del software desarrollado y el acabado del mismo dentro de los límites de tiempo requeridos.

Diversas tendencias y metodologías de desarrollo de software han aparecido en años recientes, buscando resolver los problemas que proyectos más tradicionales, no han conseguido enfrentar. Entre ellas están los frameworks de proyectos, las metodologías ágiles y los modelos de medición de madurez. Junto con estos marcos de trabajo, ciertas estrategias específicas han permitido a los equipos de desarrollo producir software más robusto, predecible, reutilizable y de fácil mantenimiento.(UCID, 2012).

La metodología que se utiliza para el desarrollo del módulo es PRODESOF, ya que constituye la más adecuada debido a que es la utilizada por la plataforma GeneSIG, para la cual se desarrollará dicho módulo.

La metodología PRODESOF se plantea como Modelo de Desarrollo de Software una combinación entre el modelo Basado en Componentes y el Iterativo e Incremental, permitiendo el primero alcanzar un mayor nivel de reutilización de software, en contextos distintos a aquellos para los que fue diseñado. El segundo plantea la descomposición del ciclo de vida del software en iteraciones, lo que hace posible la eliminación de los riesgos más significativos para el proyecto con cada nueva iteración, mientras los incrementos manejan la estrategia en la que se obtienen los resultados.(UCID, 2012).

El ciclo de vida de los proyectos que utilizan esta metodología viene definido por cinco fases establecidas por PRODESOF. Según (UCID, 2012) dichas fases son Inicio, Modelación, Construcción, Explotación Experimental y Despliegue. A continuación se presenta una descripción gráfica del ciclo de vida de los proyectos:



Fig. 1.2: Fases del ciclo de vida del proyecto (UCID, 2012).

Las fases del proyecto son divisiones dentro del mismo proyecto, donde es necesario ejercer un control adicional para gestionar eficazmente la conclusión de un entregable mayor. Las fases del proyecto suelen completarse de manera secuencial, pero en determinadas situaciones pueden superponerse.

1.6.2 Framework ExtJS 3.1

ExtJS 3.1 es una biblioteca de JavaScript que hace relativamente cómodo o sencillo crear interfaces de usuario de estilo escritorio para aplicaciones web. ExtJS posee comunicación asíncrona, este factor permite que la información sea cargada y procesada de forma tal que el usuario no lo perciba, además permite distribuir la carga de procesamiento entre cliente y servidor (Frederick, 2010).

ExtJS permite construir aplicaciones complejas en internet además de flexibilizar el manejo de componentes de la página como el DOM (del término en inglés *Document Object Model*) y peticiones AJAX (del término en inglés *Asynchronous JavaScript And XML*).

Este *framework* fue utilizado para la creación de la interfaz de usuario, ya que a través de ExtJS se puede generar una interfaz que resulte atractiva para los usuarios, al tiempo que brinda una apariencia profesional y un diseño gráfico sencillo.

1.6.3 IDE utilizado

Los IDE (del término en inglés *Integrated Development Environment*) o Entornos de Desarrollo Integrado son herramientas que facilitan la creación de aplicaciones de software. El IDE Netbeans en su versión v7.0 brinda soporte para el trabajo con PHP como lenguaje de programación, permitiendo la depuración y compilación del mismo. Netbeans posee otras características ventajosas, como es el caso del autocompletamiento de código, además se debe resaltar que es un software libre sin restricciones de uso.

1.6.4 Visual Paradigm 8.0 Enterprise Edition

Este software constituye una herramienta de modelado adecuada para el trabajo con software libre. El mismo permite automatizar la metodología de software, facilitando el diseño y la documentación de las actividades que se llevan a cabo durante el proceso de desarrollo de software. Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computadora, del término en inglés *Computer Aided Software Engineering*) que posee una interfaz amigable que brinda a los usuarios la posibilidad de interactuar con el sistema de forma fácil y sencilla.

Visual Paradigm fue utilizado para la realización de los diagramas empleados durante la investigación, lo que permitió alcanzar un mayor nivel de comprensión de la solución propuesta.

1.6.5 Lenguaje de Modelado

El Lenguaje de Modelado Unificado de diagrama o notación (UML, de las siglas en inglés *Unified Modeling Language*) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. Este lenguaje no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está diseñado para su uso con software orientado a objetos.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas que representan alguna parte o punto de vista del sistema. (Javier Garzas, 2013).

UML es el lenguaje de modelado utilizado por Visual Paradigm para esquematizar el sistema y brindar información visual del mismo.

1.6.6 Lenguaje del Servidor

PHP5 es un lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con este lenguaje. Con este lenguaje se puede procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Es utilizado tanto por pequeñas páginas web como por grandes empresas. También puede utilizar y presentar resultados en otros estándares de datos o lenguajes propios de los desarrollos web, como XHTML (del término en inglés *Extensible HyperText Markup Language*) y cualquier otro tipo de ficheros XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla, utilizando estos ficheros para generar contenido dinámico. Es decir, el contenido dinámico puede surgir de otros sitios además de desde bases de datos. También se puede interactuar con otros servidores usando cualquier protocolo. (APR, 2015).

PHP es el lenguaje de programación que fue usado en la aplicación para la realización de las funciones en el lado del servidor.

1.6.7 Servidor Web

Apache 2.2 Server es un servidor de código abierto para plataformas UNIX (Sistema operativo portable, multitarea y multiusuario). Este servidor es usado primariamente para enviar páginas web estáticas y dinámicas en la *World Wide Web* (WWW). Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o utilizan características propias de este servidor web. Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma

segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. (Apache, 2015).

Este servidor fue utilizado durante el proceso de desarrollo de la aplicación y será también el utilizado durante el proceso de despliegue.

1.6.8 Servidor de Mapas

MapServer 5.6 es una plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas interactivas para la web. Originalmente desarrollado a mediados de los 90's en la Universidad de Minnesota, MapServer es publicado bajo una Licencia tipo MIT (del término en inglés *Massachusetts Institute of Technology*), y funciona en los principales sistemas operativos (Windows, Linux, Mac OS X). Es importante resaltar que no es un sistema con todas las funcionalidades de un sistema SIG, ni tampoco inspira serlo. La posibilidad de ser utilizado como servidor de mapas por terceros programas, ha llevado a la creación de aplicaciones web basadas en MapServer para la publicación de datos geoespaciales, CartoWeb es una de ellas. (MapServer, 2015).

Este servidor fue utilizado para la creación de la aplicación, apoyándose en él para la visualización, análisis y consulta de la información geográfica brindada.

1.6.9 Openlayers

Openlayers 2.12 es una librería JavaScript. Al trabajar del lado del cliente, constituye un visor de mapas en JavaScript. La descarga de estos mapas se realiza directamente desde el navegador a través de Ajax, por lo que no genera tráfico en el servidor. Los mapas se descargan directamente del servidor de mapas que suele ser una pieza diferenciada a nivel de sistemas. Openlayers permite sobreponer distintas capas sobre una básica, añadir indicadores o puntos en el mapa con leyendas, así como polígonos, además proporciona su propio API (del término en inglés *Application Programming Interface*) para dibujarlos de una manera sencilla. (Sánchez Suárez, 2015).

1.7 Conclusiones parciales

Tanto el trabajo con información geográfica, como los medios para la realización del mismo son de gran importancia en la actualidad. Por esta razón este trabajo va encaminado al desarrollo de un módulo que representa un elemento definitivo para la obtención de mapas personalizados, brindando resultados superiores a los obtenidos mediante el uso de tecnologías ya existentes. Para lograr esto se realizó un estudio de algunos sistemas y se tomó lo mejor de cada una de ellos con el fin de aplicarlo al módulo que se quiere desarrollar, de esta forma dicho módulo cuenta con funcionalidades básicas de las principales herramientas presentes en la actualidad. Producto de esto se espera lograr la realización de un módulo apoyado en MapInfo, ArcGIS, Kosmos y principalmente QuantumGIS que permita la elaboración de mapas de impresión personalizados para dar solución a las necesidades actuales. Se hace posible la implementación de la solución ya que los aspectos abordados durante el capítulo proveen la base de conocimientos necesaria para dicha cuestión.

Capítulo 2

2 Descripción de la solución propuesta

En este capítulo se lleva a cabo la elaboración de la solución propuesta, describiendo los principales artefactos relacionados con la modelación del dominio y el levantamiento de requisitos, atendiendo a la metodología PRODESOF. Se incluyen representaciones del modelo de dominio y los conceptos asociados a él, además de los requisitos funcionales y no funcionales que debe poseer el sistema.

2.1 Modelo de dominio

Solamente se realiza el Modelo de Dominio para determinar las deficiencias existentes en el sistema y proceder con la informatización de las mismas debido a que el negocio es muy amplio y no se tienen claros todos los procesos que intervienen. Dicho modelo se realiza con el objetivo de comprender el sector de negocio al cual va a servir el sistema.

En este modelo se capturan los objetos más importantes que intervienen en el sistema, los cuales representan los eventos que tienen lugar en el entorno en que trabaja el sistema (Jacobson, 2000). A través del lenguaje UML se realiza la representación de las clases relacionadas con el problema para obtener una mejor comprensión de la estructura de la organización.

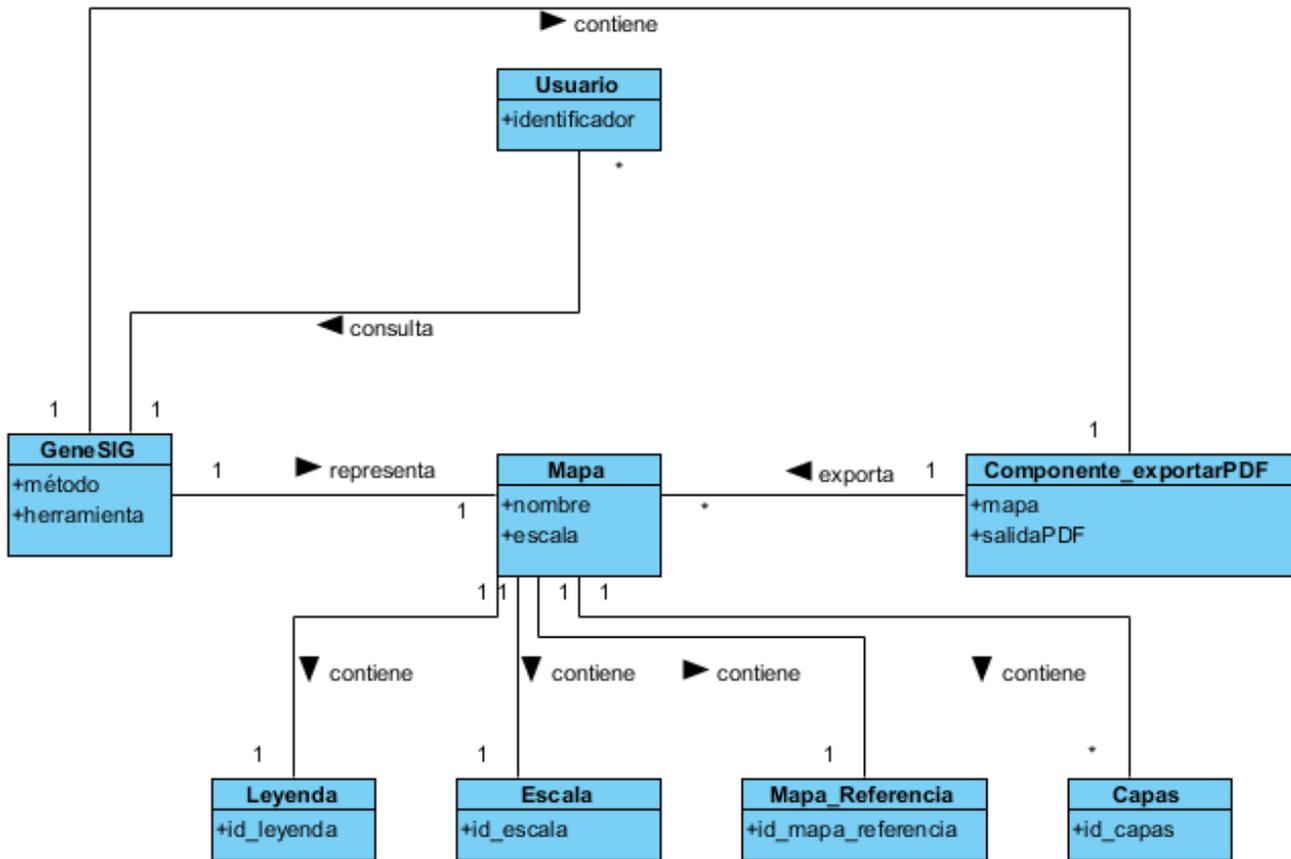


Fig. 2.3: Diagrama de clases del dominio

El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Cuando se realiza la programación orientada a objetos, el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio podría constituir en algunos casos una primera versión del sistema.

2.1.1 Descripción del modelo de dominio

La plataforma GeneSIG representa mapas que al ser consultados por los usuarios, los mismos pueden añadir a estos mapas objetos de tipo escala, mapa de referencia y leyenda. La obtención en formato duro de los mapas de impresión creados por los usuarios constituye la función del componente para la

impresión de mapas que posee GeneSIG, dicha función brinda a los usuarios la posibilidad de utilizar los documentos obtenidos en tareas donde sería más conveniente contar con mapas en formato duro.

2.1.2 Glosario de términos del dominio

GeneSIG: Plataforma para la creación de SIG que se encarga de la gestión de la información referenciada geográficamente.

Usuario: Persona u organismo que acceda al sistema con el objetivo de generar un mapa de impresión enriquecido con elementos incluidos por el mismo usuario.

Mapa: Dibujo o esquema que representa una cierta franja de un territorio sobre una determinada superficie de dos dimensiones, puede ser de tipo bidimensional o esférico.

Capas: La información temática que se usa para elaborar los SIG.

Componente Exportar a PDF: Aplicación encargada de exportar a formato PDF una zona seleccionada del mapa visualizado tras haber realizado sobre este las modificaciones necesarias por parte de los usuarios para lograr la adaptación a la cuestión existente.

Escala: Gráfico que consiste en una línea o barra graduada que indica la longitud correspondiente en el plano.

Leyenda: Explicación que se provee sobre los símbolos y colores que se presentan en el mapa.

Mapa de Referencia: Elementos que muestran diversos aspectos de las ubicaciones.

2.2 Modelo del sistema

Una vez realizado el modelo de dominio y la descripción de los elementos que en el intervienen, con el objetivo de encaminar de forma correcta el desarrollo se presentan a continuación las especificaciones de los requisitos funcionales y no funcionales con el fin de establecer las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe trabajar. Los requisitos deben ser claros, correctos, inequívocos, específicos y comprobables.

2.2.1 Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (UCID, 2012).

A continuación se muestran los RNF que debe poseer el sistema:

Usabilidad:

- **RNF 1** El sistema podrá ser usado por personas con conocimientos básicos en el manejo de Computadoras.
- **RNF 2** El módulo podrá ser usado por personas con conocimientos básicos de la plataforma GeneSIG.
- **RNF 3** El sistema mantendrá al usuario informado sobre las funcionalidades de los elementos mediante el uso de *Tooltips*.

Eficiencia:

- **RNF 4** El sistema informará al usuario mediante la muestra de mensajes sobre los cambios relevantes que este realice.
- **RNF 5** El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Utilizando como umbral un valor de 5 segundos.

Restricciones de Diseño:

- **RNF 6** El producto de software final debe diseñarse sobre una arquitectura que permita una integralidad con la plataforma GeneSIG.

- **RNF 7** Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará para la plataforma GeneSIG.

Interfaz de Usuario:

- **RNF 8** Tener una apariencia profesional y un diseño gráfico sencillo con la utilización de las tonalidades de los colores representativos de la plataforma GeneSIG.

Hardware: En este caso el módulo presenta los mismos requisitos de hardware de la plataforma GeneSIG que a continuación se enuncian:

Para las PC clientes:

- **RNF 9** Se requiere tengan tarjeta de red.
- **RNF 10** Al menos 128 MB de memoria RAM.
- **RNF 11** Procesador 512 MHz como mínimo.

Para los servidores:

- **RNF 12** Se requiere tengan tarjeta de red.
- **RNF 13** El Servidor de la aplicación tenga como mínimo 2GB de RAM y 80 GB de disco duro.
- **RNF 14** El Servidor de BD tenga como mínimo 2GB de RAM y 80 GB de disco duro.
- **RNF 15** El servidor de mapas tenga como mínimo 2GB de RAM.
- **RNF 16** Procesador 3 GHz como mínimo.

Software:

Para las PC clientes:

- **RNF 17** Un navegador como Mozilla Firefox u otro navegador que cumpla con los estándares W3C.
- **RNF 18** Sistema operativo: GNU/Linux o Windows.

Para los Servidores:

- **RNF 19** Sistema operativo: Debian Squeeze 6.0.
- **RNF 20** Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- **RNF 21** Servidor de mapas MapServer 5.6.
- **RNF 22** PostgreSQL como Sistema Gestor de Base de Datos.

Licencia:

- **RNF 23** De acuerdo a las licencias de las herramientas que se proponen a utilizar para el desarrollo del módulo se puede decir que el mismo posee una arquitectura de modelo libre, que permitirá la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la Autorización de sus respectivos titulares.

2.2.2 *Requisitos funcionales*

Los requisitos funcionales (RF) definen las condiciones o capacidades que el sistema será capaz de realizar. Estos describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Estos requisitos, al tiempo que avanza el proyecto de software, se convierten en los algoritmos, la lógica y gran parte del código del sistema.(UCID, 2012).

Seguidamente se plantean los requisitos funcionales que debe poseer el sistema:

- **RF 1** Guardar documento: Permite guardar el mapa en formato PDF tras haber modificado el mismo.
- **RF 2** Deshacer cambios en el documento: Permite retroceder en las acciones de modificación realizadas sobre el mapa de impresión.
- **RF 3** Rehacer cambios en el documento: Permite avanzar en las acciones de modificación realizadas sobre el mapa de impresión.
- **RF 4** Adicionar imagen del mapa: Permite añadir al mapa de impresión imágenes existentes en el sistema.

- **RF 5** Adicionar imagen: Permite añadir al mapa de impresión imágenes personalizadas.
- **RF 6** Adicionar etiqueta de texto: Permite añadir al mapa de impresión etiquetas de texto para facilitar la identificación de los objetos que en él se reflejan.
- **RF 7** Adicionar leyenda: Permite añadir al mapa de impresión una leyenda con el fin de comprender la información mostrada.
- **RF 8** Adicionar escala: Permite añadir al mapa de impresión una escala gráfica con el fin de conocer las distancias verdaderas de los lugares representados.
- **RF 9** Adicionar figura geométrica: Permite añadir al mapa de impresión figuras geométricas existentes en el sistema con el fin de enriquecer la información mostrada.
- **RF 10** Modificar elementos adicionados: Permite personalizar los elementos adicionados al mapa
- **RF 11** Eliminar Etiqueta: Permite eliminar el objeto de tipo etiqueta añadido al mapa.
- **RF 12** Eliminar Leyenda: Permite eliminar el objeto de tipo leyenda añadido al mapa.
- **RF 13** Eliminar Escala: Permite eliminar el objeto de tipo escala añadido al documento.
- **RF 14** Eliminar figura geométrica: Permite eliminar el objeto de tipo figura geométrica añadido al documento.
- **RF 15** Eliminar Imagen: Permite eliminar el objeto de tipo imagen añadido al documento.

A continuación se muestra la descripción de dos de estos requisitos:

Tabla 2.1RF 6: Adicionar Etiqueta de Texto:

| Conceptos tratados | Conceptos | Atributos |
|--------------------|--|---------------|
| | No procede | No procede |
| Precondiciones | Precondiciones | Pre-requisito |
| | No procede | No procede |
| Descripción | <ol style="list-style-type: none"> 1. Se selecciona la opción del menú principal Edición/ Agregar texto o mediante acceso directo en la barra de herramienta representado por el ícono (). Se activa la herramienta para la adición del texto. 2. Se muestra activa la pestaña Propiedades del elemento ubicado en el panel izquierdo de la interfaz, en el cual se encuentran ubicadas las propiedades del texto. Ver Figura 2.1. | |

| | |
|--|---|
| | <ol style="list-style-type: none">3. Se da clic en el área de trabajo donde se encuentran ubicadas las páginas que conforman el documento y se arrastra presionando el clic izquierdo, para conformar el tamaño del recuadro donde se ubica el texto.4. Se introducen en el campo Texto, los datos de la etiqueta de texto que se desea adicionar. Se actualiza en el área de trabajo el texto introducido.5. Se modifican o se introducen los datos en el <i>fieldset</i> Apariencia:<ul style="list-style-type: none">• Tipo de letra: se selecciona el tipo de letra con la cual se desea mostrar el texto, se muestran los formatos que se encuentran registrados en el sistema operativo que se tiene instalado.• Estilo del tipo de letra: se selecciona el o los estilos para el tipo de letra, ya sea Normal, Negrita, Cursiva o Subrayada. Si se desea seleccionar más de un estilo se marca el <i>checkbox</i> correspondiente al elemento.• Alineación: se selecciona la alineación para el texto agregado ya sea Izquierda, Centro, Derecha o Justificado.• Tamaño: se introduce el tamaño de fuente para el texto, inicialmente se muestra por defecto 11.• Color: se modifica el color del texto, inicialmente se muestra por defecto el color negro.6. Se modifican los campos que se corresponden con el tamaño y la posición del texto desplegando el <i>fieldset</i> Tamaño y posición:<ul style="list-style-type: none">• Posición en X: se modifica el valor de la posición del texto en X. Esta opción se puede realizar seleccionando la imagen y ubicando el cursor encima de la misma, se da clic y se arrastra hacia la posición deseada.• Posición en Y: se modifica el valor de la posición del texto en Y. Esta opción se puede realizar seleccionando la imagen y ubicando el cursor encima de la misma, se da clic y se arrastra hacia la posición deseada.• Ancho: se modifica el ancho del recuadro que conforma el texto. Esta opción se puede realizar seleccionando el texto y ubicando el cursor sobre los lados del recuadro que lo conforman, se muestra una flecha que indica la dirección, se da clic sobre la misma y se arrastra el cursor hasta la posición deseada.• Altura: se modifica la altura del recuadro que conforma el texto. Esta opción se puede realizar seleccionando el texto y ubicando el cursor sobre la parte superior e inferior del recuadro que lo conforma, se muestra una flecha que |
|--|---|

| | |
|----------------------------|---|
| | <p>indica la dirección, se da clic sobre la misma y se arrastra el cursor hasta la posición deseada.</p> <ol style="list-style-type: none"> Se modifican los campos que se corresponden con el ancho y el borde de del recuadro del texto desplegando el <i>fieldset</i> Ancho y borde: <ul style="list-style-type: none"> Color de fondo: se modifica el color del fondo del recuadro que conforma el texto. Color de borde: se modifica el color del borde del recuadro que conforma el texto. Ancho de borde: se modifica el grosor del borde del recuadro que conforma el recuadro texto. Estilo de línea: se selecciona el estilo de línea del borde del recuadro que conforma el texto. Se muestra en el área de trabajo el texto teniendo en cuenta las propiedades modificadas. |
| <p>Validaciones</p> | <ol style="list-style-type: none"> Si se dejan vacíos campos obligatorios se muestra el mensaje de error en el <i>tooltip</i> del campo correspondiente: <i>"Este campo es obligatorio."</i> El campo Tamaño solo permite números enteros positivos entre 8 y 72. Si se introducen datos inválidos se muestra el mensaje de error en el <i>tooltip</i> del campo: <i>"Solo admite números enteros positivos entre 8 y 72."</i> El campo Color, solo admite números hexadecimales. Si se introducen datos inválidos se muestra el mensaje de error en el <i>tooltip</i> del campo: <i>"El color debe tener formato hexadecimal."</i> Los campos Posición en X, Posición en Y, Ancho y Altura son obligatorios. Admiten números reales positivos. Si se introducen datos incorrectos se muestra el mensaje de error en el <i>tooltip</i> del campo: <i>"Solo admite números reales positivos."</i> Los campos Color de fondo, Color de borde, Estilo de línea y Ancho de borde son obligatorios. El campo Ancho de borde solo permite números reales positivos entre 0 y 50. Si se introducen datos inválidos se muestra el mensaje de error en el <i>tooltip</i> del campo: <i>"Solo admite números reales positivos entre 0 y 50."</i> 5, 7 Los campos Color, Color de fondo y Color de borde, solo admite números hexadecimales. Si se introducen datos inválidos se muestra el mensaje de error |

| | |
|-------------------------|---|
| | en el <i>tooltip</i> del campo correspondiente: <i>“El color debe tener formato hexadecimal.”</i> . |
| Post-condiciones | Se ha adicionado una etiqueta de texto. |
| Post-requisito | No procede. |



Fig. 2.4: Prototipo de IU Propiedades del elemento/Etiqueta

Tabla 2.2RF 8: Adicionar Escala:

| Conceptos tratados | Conceptos | Atributos |
|--------------------|-----------|-----------|
|--------------------|-----------|-----------|

| | | |
|-----------------------|--|----------------------------|
| | No procede | No procede |
| Precondiciones | Precondiciones | Pre-requisito |
| | No procede | No procede |
| | Se debe haber añadido al menos una imagen del mapa. | Adicionar imagen del mapa. |
| Descripción | <ol style="list-style-type: none"> Se selecciona la opción del menú principal Edición/ Agregar escala o mediante acceso directo en la barra de herramienta representado por el ícono (). Se activa la herramienta para la adición de la escala. Se muestra activa la pestaña Propiedades del elemento ubicado en el panel izquierdo de la interfaz, en el cual se encuentran ubicada las propiedades de la escala. Ver Figura 2.2. Se da clic en el área de trabajo donde se encuentran ubicadas las páginas que conforman el documento y se arrastra presionando el clic izquierdo, para conformar el tamaño del recuadro donde se ubica la escala. Se modifican o se introducen los datos en el <i>fieldset</i> Propiedades principales: <ul style="list-style-type: none"> Mapa: se selecciona el mapa con el cual se desea relacionar la leyenda desplegando el campo Mapa. Inicialmente se muestra por defecto la denominación de la última imagen del mapa que ha sido añadida. Estilo: se selecciona el estilo en el que se va a mostrar la escala. Los estilos registrados son: Numérico, Recuadro simple, Recuadro doble, Línea con marcas arriba y Línea con marcas abajo. Se muestra por defecto el estilo numérico. Unidad: se selecciona la unidad en la que se desea mostrar la escala, que pueden ser: Unidades del mapa, Metros, Pies o Millas náuticas. Inicialmente se muestra por defecto Unidades del mapa. Cantidad de segmentos. se introducen la cantidad de segmentos que se desean mostrar en la escala. Transparencia: campo que permite aumentar o disminuir la transparencia de la escala. Se modifican o se introducen los datos en el <i>fieldset</i> Letras: | |

| | |
|--|---|
| | <ul style="list-style-type: none">• Fuente: campo que permite modificar la fuente de la letra para la escala.• Estilo: se selecciona el o los estilos para el tipo de letra, ya sea Normal, Negrita, Cursiva o Subrayada. Si se desea seleccionar más de un estilo se marca el <i>checkbox</i> correspondiente al elemento.• Tamaño: se introduce el tamaño de fuente, inicialmente se muestra por defecto• Color: se modifica el color del texto, inicialmente se muestra por defecto el color negro. <p>6. Se modifican los campos que se corresponden con el tamaño y la posición de la escala, desplegando el <i>fieldset</i> Tamaño y posición:</p> <ul style="list-style-type: none">• Posición en X: se modifica el valor de la posición de la escala en X. Esta opción se puede realizar seleccionando la escala y ubicando el cursor encima de la misma, se da clic y se arrastra hacia la posición deseada.• Posición en Y: se modifica el valor de la posición de la escala Y. Esta opción se puede realizar seleccionando la escala y ubicando el cursor encima de la misma, se da clic y se arrastra hacia la posición deseada.• Ancho: se modifica el ancho del recuadro que conforma la escala. Esta opción se puede realizar seleccionando la escala y ubicando el cursor sobre los lados del recuadro que lo conforman, se muestra una flecha que indica la dirección, se da clic sobre la misma y se arrastra el cursor hasta la posición deseada.• Altura: se modifica la altura del recuadro que conforma la escala. Esta opción se puede realizar seleccionando la leyenda y ubicando el cursor sobre la parte superior e inferior del recuadro que lo conforma, se muestra una flecha que indica la dirección, se da clic sobre la misma y se arrastra el cursor hasta la posición deseada. <p>7. Se modifican los campos que se corresponden con el ancho y el borde de del recuadro del texto desplegando el <i>fieldset</i> Ancho y borde:</p> <ul style="list-style-type: none">• Color de fondo: se modifica el color del fondo del recuadro que conforma la escala.• Color de borde: se modifica el color del borde del recuadro que conforma la escala.• Ancho de borde: se modifica el grosor del borde del recuadro que conforma la escala.• Estilo de línea: se selecciona el estilo de línea del borde del recuadro que |
|--|---|

| | |
|-------------------------|--|
| | <p>conforma la escala.</p> <p>8. Se muestra en el área de trabajo la escala, teniendo en cuenta las modificaciones realizadas.</p> |
| Validaciones | <ol style="list-style-type: none"> 1. Si no existe al menos una imagen del mapa añadida, la opción agregar escala se muestra inactiva. 2. El campo Cantidad de segmentos es un campo numérico obligatorio que solo admite números enteros positivos, su valor máximo es 100. Si se introducen datos inválidos o mayores que 100 se muestra el mensaje de error en el tooltip del campo: <i>“Solo admite números enteros positivos entre 1 y 100.”</i>. 3. Si se dejan vacíos campos obligatorios se muestra el mensaje de error en el tooltip del campo correspondiente: <i>“Este campo es obligatorio.”</i>. 4. El campo Tamaño solo permite números enteros positivos entre 8 y 72. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo: <i>“Solo admite números enteros positivos entre 8 y 72.”</i>. 5. El campo Color, solo admite números hexadecimales. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo: <i>“El color debe tener formato hexadecimal.”</i>. 6. Los campos Posición en X, Posición en Y, Ancho y Altura son obligatorios. Admiten números reales positivos. Si se introducen datos incorrectos se muestra el mensaje de error en el tooltip del campo: <i>“Solo admite números reales positivos.”</i>. 7. Los campos Color de fondo, Color de borde, Estilo y Ancho de borde son obligatorios. 8. El campo Ancho de borde solo permite números reales positivos entre 0 y 50. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo: <i>“Solo admite números reales positivos entre 0 y 50.”</i>. 9. Los campos Color de fondo y Color de borde, solo admite números hexadecimales. Si se introducen datos inválidos se muestra el mensaje de error en el tooltip del campo correspondiente: <i>“El color debe tener formato hexadecimal.”</i>. |
| Post-condiciones | Se ha adicionado una etiqueta de texto. |
| Post-requisito | No procede. |

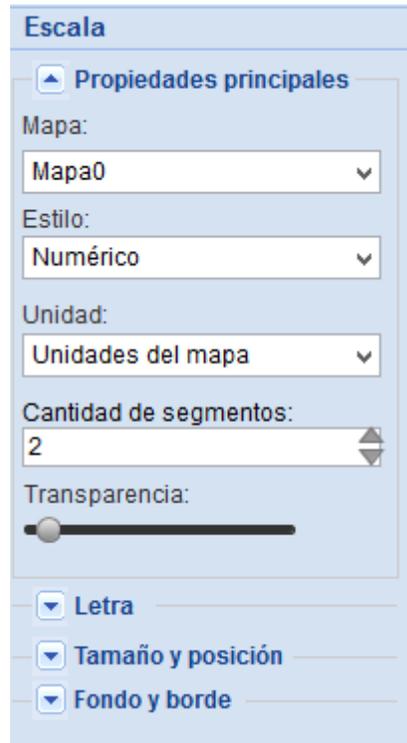


Fig. 2.5: Prototipo de IU Propiedades del elemento/Escala

En el documento de Especificación de Requisitos de Software (ERS) del proyecto se brinda una descripción más detallada de los RF asociados a la solución propuesta.

2.3 Conclusiones parciales

En este capítulo se desarrolló el modelo de dominio asociado al sistema, permitiendo identificar los problemas existentes. Tras la realización del levantamiento de requisitos se obtuvieron las funcionalidades con las que debe contar el sistema, permitiendo conocer el comportamiento de los usuarios durante su interacción con el mismo. Este capítulo brinda una amplia documentación referente a los conceptos relacionados con el sistema y a las restricciones que se deben tener en cuenta durante su implementación.

Capítulo 3

3 Implementación y validación

Este capítulo se basa en el proceso de desarrollo de la solución propuesta y en el transcurso del mismo se abordarán aspectos tales como los patrones de diseño utilizados, el modelo de despliegue, el modelo de datos, el modelo de diseño y la ejecución de los casos de pruebas para detectar las no conformidades presentes en el software.

3.1 Arquitectura de Software

En una definición tal vez demasiado amplia, David Garlan (Carnegie Mellon University, 2013) establece que la Arquitectura de Software constituye un puente entre el requisito y el código, ocupando el lugar que en los modelos antiguos se reservaba para el diseño. Por otra parte en el documento de IEEE Std 1471 – 2000 (Committee, 2000), adoptada también por Microsoft, se define la arquitectura como *“la organización fundamental de un sistema encarnado en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución”*.

La arquitectura enfoca el diseño del software desde varias perspectivas, de la calidad de estos procesos dependerá el éxito del diseño detallado, implementación e integración de la aplicación.

3.1.1 Patrones de Arquitectura

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. (Alonso, y otros, 2004). Los patrones arquitectónicos que se muestran a continuación fueron utilizados en la solución:

1. Arquitectura Orientada a Objetos:

Los componentes de este estilo son los objetos, o más bien instancias de los tipos de datos abstractos. Se basa en los principios de encapsulamiento, herencia y polimorfismo. Son así mismo las unidades de modelado, diseño e implementación, y los objetos y sus interacciones

son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. Además las interfaces están separadas de las implementaciones.(Romero, y otros, 2013).

2. Arquitectura Basada en Componentes:

Un componente es una pieza de código pre-elaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándares. Este modelo posee características del modelo espiral, es evolutivo por naturaleza y exige un enfoque interactivo para la creación de software, además de que posibilita alcanzar un alto nivel de reutilización de software. Este patrón está dividido en 4 etapas: la de análisis y comparación del proceso de desarrollo de software basado en componente, análisis arquitectural, identificación de componentes y especificación de los principales componentes.(Romero, y otros, 2013).

La plataforma GeneSIG está desarrollada sobre el framework CartoWeb, el cual posee una arquitectura Orientada a Objetos y Basada en Componentes. Producto de esto, la arquitectura aplicada a la solución propuesta será una mezcla de las antes mencionada, debido a que dicha solución se desarrolla para la plataforma GeneSIG.

3.1.2 Patrones de Diseño

Para la solución propuesta se seleccionan los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades, del término en inglés *General Responsibility Assignment Software Patterns*) ya que los mismos representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones.

A continuación se describen los patrones empleados en la solución propuesta:

Bajo Acoplamiento: Este patrón se utiliza con el fin de disminuir el impacto del cambio en las clases involucradas en el sistema. El acoplamiento viene dado por la fuerza de la conexión existente entre las clases, con este patrón se disminuye dicho acoplamiento ya que vela por que las clases se conecten entre ellas solo cuando sea necesario, viéndose de esta forma menos afectadas por los cambios realizados a otros componentes.

Alta Cohesión: Mediante este patrón se logra una simplificación de las mejoras y el mantenimiento realizado a la aplicación. Se pone en práctica al otorgar a cada elemento una labor única en el sistema, de

esta forma se evita la sobrecarga de las clases ya que las mismas colaboran entre ellas para llevar a cabo las tareas. Hay que resaltar que la presencia de una alta cohesión en los elementos del sistema facilita la reutilización de los mismos.

Experto: Se encarga de velar por que la implementación de que un método se realice en la clase que cuente con la información requerida para la ejecución de este método. La utilización de este patrón está presente en métodos sencillos como el `eliminarElemento()`, donde la única información que debe conocer el sistema es el elemento actual o seleccionado para proceder con su eliminación.

Además de los patrones GRASP se utilizan también los patrones GOF (del término en inglés *Gang Of Four*), que están dirigidos al desarrollo de los sistemas orientados a objetos y que se clasifican en tres categorías: creacionales, estructurales y de comportamiento.

Singleton: Este patrón se encarga del acceso a las clases y su función es velar por que haya una sola instancia mediante la cual se pueda acceder a una clase. Utilizando para esto un mecanismo de acceso global. Este patrón se pone de manifiesto en la clase `ServerContext` donde el método `getmapobject()` crea una instancia del mapa, la cual es única en todo el sistema.

Patrón Command: Este patrón se manifiesta cuando se realiza alguna petición de información por parte del cliente. Su funcionamiento se basa en la vinculación de un comando a un objeto, para que pueda ser procesado como si fuera uno, es decir que puede ser utilizado en métodos, almacenado e incluso devuelto. En este caso, la clase `AjaxHelper` se encarga de comunicar las interfaces con el servidor.

3.2 Elementos del Diseño de Clases

En el diseño de clases se resume la definición de las clases que se pueden implementar en el software, se visualizan las relaciones entre ellas y se muestra gráficamente la interacción de los objetos para comunicarse entre sí. (UCID, 2012).

Seguido se muestra el diagrama de clases del diseño de la funcionalidad Adicionar Etiqueta:

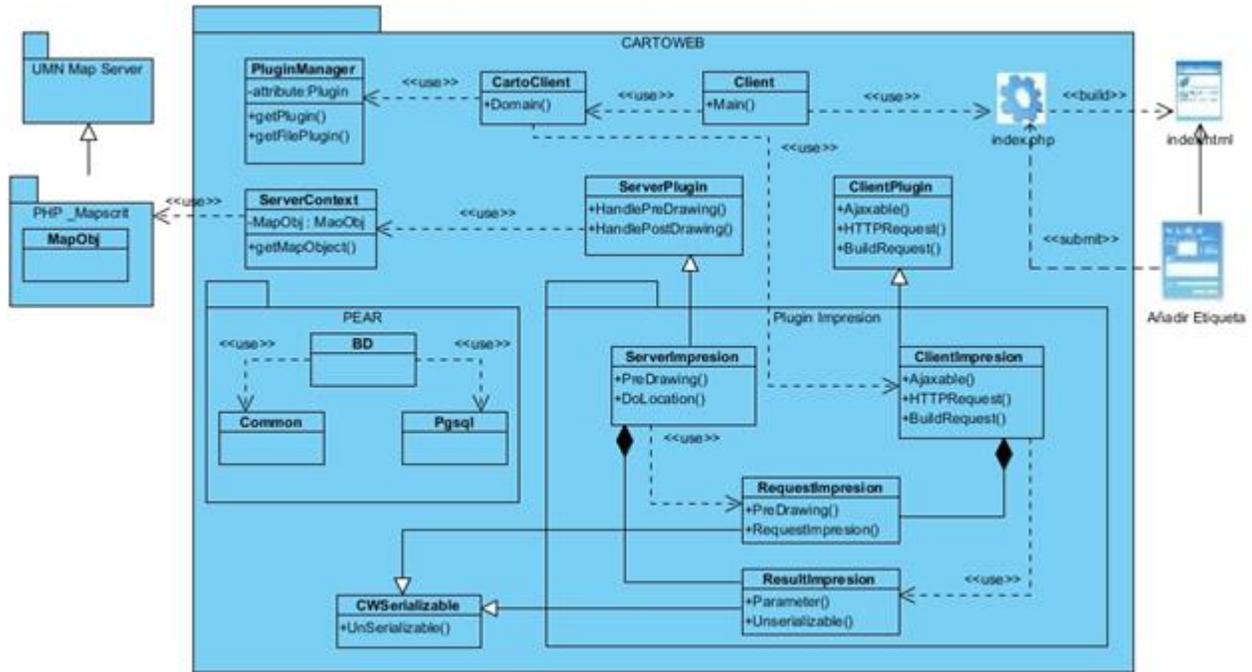


Fig. 3.6: Diagrama de Clases del Diseño

3.3 Modelo de Implementación

Este modelo describe cómo se organizan los componentes atendiendo a los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje de programación utilizado. Con la realización de dicho diagrama se brinda una descripción detallada de cómo los elementos que lo conforman se implementan en términos de componentes, como se puede apreciar a continuación:

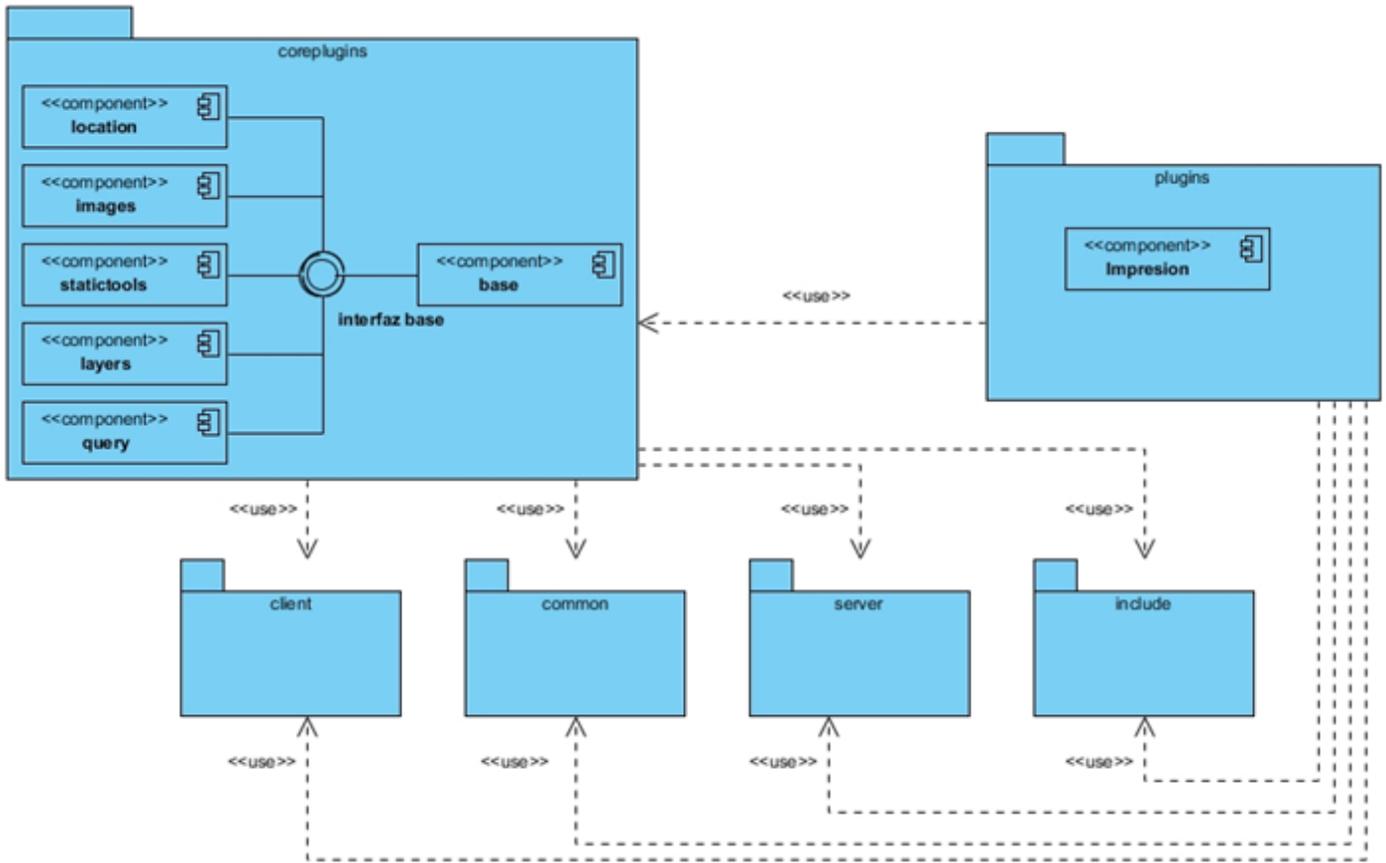


Fig. 3.7: Diagrama de Componentes

3.4 Modelo de Despliegue

Este modelo está conformado por componentes, nodos y relaciones. Se utiliza tanto para visualizar la distribución de los elementos que conforman el *software* del sistema en los nodos físicos como para representar el *hardware* utilizado en la implementación del sistema.

A continuación se muestran los elementos de hardware que intervienen en el componente a desarrollar, así como los servidores, requisitos que los mismos deben cumplir y la forma de comunicación entre ellos:

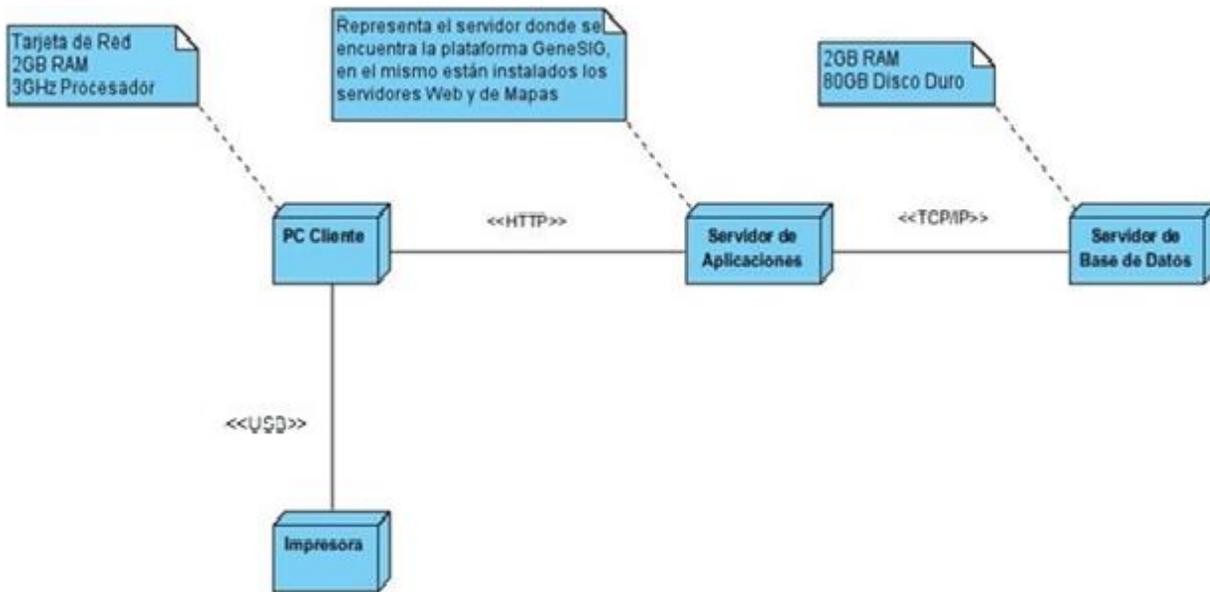


Fig. 3.8: Diagrama de Despliegue

3.5 Modelo de Pruebas

Las pruebas del software son la actividad más común de control de la calidad realizada en los proyectos para asegurar el correcto funcionamiento del software. Tienen como objetivos la verificación de la correcta implementación de los requisitos explícitamente establecidos, la adecuada integración de los componentes que conforman el sistema y la ejecución de casos de prueba que permitan detectar el mayor número de no conformidades y corregirlas antes de la entrega del software al cliente.(UCID, 2012).

Es importante destacar que las pruebas reducen la probabilidad de que aparezcan defectos ocultos en el software, pero incluso si no se encuentra ningún defecto, nunca será una garantía de su corrección.

3.5.1 Prueba de aceptación

Las pruebas de aceptación constituyen un escenario de utilización del sistema para comprobar el comportamiento que se espera por parte de este, teniendo en cuenta el punto de vista del sistema externo o del usuario que interactúe con la aplicación y también del cliente.

El objetivo de la realización de esta prueba es validar que el sistema cumpla con las funcionalidades esperadas por el cliente así como brindar a los directivos del proyecto la posibilidad de expresar su conformidad en cuanto a la solución propuesta.

Líder de proyecto (Aplicativos SIG): Ing. Alain León Companioni

Líder de proyecto (GeneSIG): Ing. Grethell Castillo Reyes

Analista principal: Ing. Leiny Ruíz Cervera.

Líder de los desarrolladores (Aplicativos SIG): Ing. Alain León Companioni

Líder de los desarrolladores (GeneSIG): Ing. Grethell Castillo Reyes

Administrador de configuraciones: Ing. Miosotis Aida troche

Programadores: Ing. Yoan Mandina

Ing. Arlettys Silvera Boffill

Ing. Camilo Madariaga Dozaret

A partir de la realización de estas pruebas el proyecto emitió un acta de aceptación del módulo de impresión realizado para la Plataforma GeneSIG.

3.5.2 Prueba de Caja Negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa (Lara García, 2013) razón por la que se denomina **caja negra** a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca (Pressman, 5ta Edición).

Los objetivos perseguidos con las pruebas de caja negra son demostrar que:

1. Las funciones del software son operativas.

2. La entrada se acepta de forma correcta.
3. Se produce una salida correcta.
4. La integridad de la información externa se mantiene.

La prueba se basó en el método de Partición Equivalente, el cual consiste en la división del dominio de entrada de un programa en clases de datos que tienden a ejercitar determinadas funciones del *software*. Se realizaron dos iteraciones al sistema, durante la primera se detectó una no conformidad en el requisito funcional 6: Adicionar Etiqueta de Texto, donde se detectaron errores que afectaban el correcto funcionamiento de este. Una vez determinado y solucionado el problema, se realizó la segunda iteración para verificar la no persistencia de este y para comprobar si el resto del sistema funcionaba correctamente, lo cual concluyó satisfactoriamente. A continuación se muestran los casos de prueba resultantes en la segunda iteración:

Tabla 3.1: Secciones a probar en el RF Adicionar Etiqueta de Texto.

| Nombre de la sección | Escenario | Acción realizada | Respuesta del sistema | Resultado de la prueba |
|-----------------------------------|--|---|--|--------------------------|
| SC 1: Adicionar Etiqueta de Texto | EC 1.1: Modificar propiedades de la etiqueta | El usuario introduce los valores personalizados de la etiqueta. | El sistema captura los valores introducidos por el usuario y modifica la etiqueta. | Resultado Satisfactorio. |

Tabla 3.2: Secciones a probar en el RF Deshacer.

| Nombre de la sección | Escenario | Acción realizada | Respuesta del sistema | Resultado de la prueba |
|-------------------------|---|---|--|--------------------------|
| SC 1: Deshacer etiqueta | EC 1.1: Deshacer las modificaciones realizadas a la etiqueta. | El usuario regresa paso por paso hacia las modificaciones realizadas a la etiqueta. | El sistema actualiza la etiqueta con los valores obtenidos en cada paso. | Resultado Satisfactorio. |
| | EC 1.2: Se | El usuario llega al | El sistema elimina la | Resultado |

| | | | | |
|--|---|--|---|--------------------------|
| | deshacen todas las modificaciones realizadas a la etiqueta. | estado inicial de la etiqueta antes de ser modificada. | etiqueta. | Satisfactorio. |
| | EC 1.3: Se elimina la etiqueta. | El usuario elimina la etiqueta previamente modificada. | El sistema oculta el panel de modificación de dicha etiqueta. | Resultado Satisfactorio. |

Tabla 3.3: Secciones a probar en el RF Adicionar Imagen.

| Nombre de la sección | Escenario | Acción realizada | Respuesta del sistema | Resultado de la prueba |
|------------------------|---------------------------------------|--|---|--------------------------|
| SC 1: Adicionar Imagen | EC 1.1: Seleccionar la imagen deseada | El usuario introduce la url de la imagen deseada | El sistema carga la imagen en el panel de edición | Resultado Satisfactorio. |
| | EC 1.2: Seleccionar la imagen deseada | El usuario introduce la url de la imagen deseada | El sistema muestra un mensaje comunicando que la url introducida no es válida | Resultado Satisfactorio. |

Tabla 3.4: Secciones a probar en el RF Rehacer.

| Nombre de la sección | Escenario | Acción realizada | Respuesta del sistema | Resultado de la prueba |
|----------------------|---|--|--|--------------------------|
| SC 1: Rehacer Imagen | EC 1.1: Rehacer las modificaciones realizadas a la imagen | El usuario avanza paso por paso hacia las modificaciones realizadas a la imagen. | El sistema actualiza la imagen con los valores obtenidos en cada paso. | Resultado Satisfactorio. |
| | EC 1.2: Se rehacen | El usuario llega al | El sistema | Resultado |

| | | | | |
|--|---|--|---|----------------|
| | todas las modificaciones realizadas a la imagen | estado final de la imagen después de ser modificada. | restablece la imagen al punto de modificación especificado. | Satisfactorio. |
|--|---|--|---|----------------|

3.5.3 Prueba de Caja Blanca

Según (Lara García, 2013) las pruebas de caja blanca garantizan que:

1. Se ejecuten al menos una vez todos los caminos independientes de cada módulo.
2. Se utilicen las decisiones en su parte verdadera y en su parte falsa.
3. Se ejecuten todos los bucles en sus límites.
4. Se utilicen todas las estructuras de datos internas.

Esta prueba constituye un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Según (Pressman, 5ta Edición) el seguimiento de estos pasos es crucial para la correcta realización de esta prueba:

1. Generar grafo de flujos de datos.
2. Cálculo de complejidad ciclomática, $V(G)$.

$$V(G) = NA \text{ (Número de Aristas)} - NN \text{ (Número de Nodos)} + 2.$$

$$V(G) = P \text{ (Nodos predicados)} + 1.$$

$$V(G) = R \text{ (Número de regiones)}.$$

Determinar los caminos independientes o básicos.

3. Generar un caso de prueba para cada camino de ejecución.

Las pruebas de caja blanca utilizan el método del camino básico, el cual permite al diseñador de casos de prueba estimar la complejidad lógica de un diseño procedimental y apoyarse en esa estimación como guía para la definición de un conjunto básico de caminos de ejecución. Este método posibilita la detección de

errores no detectados anteriormente en el código. De igual manera el mantenimiento del sistema sería de forma más efectiva y certera.

Método Crear Imagen de la clase controladora (ver Anexo 1):

Paso1: Generar el grafo.

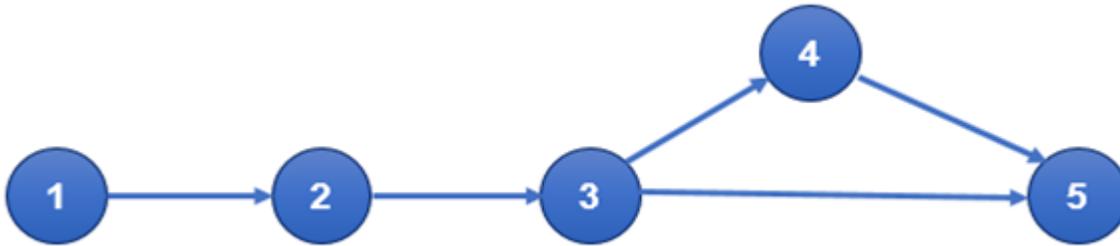


Fig. 3.9: Grafo de flujo de datos de la función crear imagen.

Paso2: Calcular la complejidad.

$$V(G) = NA \text{ (Número de Aristas)} - ND \text{ (Número de Nodos)} + 2$$

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = 1 + 1$$

$$V(G) = 2$$

$$V(G) = R \text{ (Número de regiones)}$$

$$V(G) = 2$$

Paso3: Determinar los caminos básicos.

CB1: 1-2-3-4-5

CB2:1-2-3-5

Paso4: Caso de prueba para el camino básico.

Caso de prueba para camino básico 1.

Entrada: posElemActual == 0.

Resultado esperado: Oculta panel de configuración.

Resultado de la prueba: Satisfactorio.

Caso de prueba para camino básico 2.

Entrada:posElemActual != 0.

Resultado esperado: Mostrar panel de configuración.

Resultado de la prueba: Satisfactorio.

3.6 Conclusiones Parciales

Con el objetivo de representar cada componente que interviene en el desarrollo del módulo para el diseño de mapas de impresión, se realizó el diagrama de componentes, con este se describen los principales elementos del diseño en términos de componentes. La realización del diagrama de despliegue permitió representar la distribución física de los nodos para la instalación del sistema y las dependencias existentes entre estos. Además, con el fin de garantizar el correcto funcionamiento del sistema, se realizaron pruebas para probar dicho funcionamiento. Basado en los resultados obtenidos mediante estas se puede afirmar que el sistema no presenta errores funcionales.

Conclusiones Generales

Se logró de forma satisfactoria el desarrollo de un módulo que permite el diseño flexible de mapas para impresión en la plataforma GeneSIG, por lo que se consideran cumplidos los objetivos propuestos con la realización de esta aplicación.

El módulo desarrollado permite a los usuarios realizar un amplio número de configuraciones sobre los elementos que conforman el mapa que se encuentran diseñando. Con esto se consigue alcanzar un mayor nivel de personalización de dichos mapas, lo que aumenta la capacidad de adaptación a determinado negocio que lo requiera.

Analizando los resultados obtenidos se puede concluir que:

- La utilización de herramientas y tecnologías libres en la implementación permite el cumplimiento de la política de soberanía tecnológica y migración a software libre que lleva a cabo el país actualmente.
- El proceso de desarrollo seleccionado para el desarrollo del módulo sirvió de guía en la implementación y documentación del mismo.
- La plataforma GeneSIG cuenta con un módulo para el diseño flexible de mapas para impresión.
- Los usuarios de dicho módulo pueden realizar todo tipo de configuraciones sobre estos mapas.
- Los usuarios pueden generar el mapa diseñado en formato PDF.

Recomendaciones

Una vez alcanzado el objetivo de la investigación, y basándose en las experiencias obtenidas, se recomienda:

- La inclusión en el módulo de funcionalidades que permitan la obtención del diseño de mapas para impresión en formatos .jpg y .doc.
- La inclusión en el módulo de una funcionalidad que permita exportar el diseño realizado como plantilla .xml.

Glosario de Términos

SIG (Sistemas de Información Geográfica)

HTML Lenguaje de Marcas de Hipertexto (del término en inglés *HyperText Markup Language*)

PDF Formato de Documento Portable (del término en inglés *Portable Document Format*)

XML Lenguaje de Marcas Extensible (del término en inglés *Extensible Markup Language*)

GUI Interfaz Gráfica de Usuario (del término en inglés *Graphic User Interface*)

OOP Programación Orientada a Objetos (del término en inglés *Object-Oriented Programming*)

PHP Preprocesador de Hipertexto (*Hypertext Preprocessor*)

BSD (*Berkeley Software Distribution*),

HTTP Protocolo de Transferencia de Hipertexto (del término en inglés *HyperText Transfer Protocol*)

SOAP Protocolo de Acceso Simple a Objetos (del término en inglés *Simple Object Access Protocol*)

DOM Modelo de Objetos del Document o (del término en inglés *Document Object Model*)

AJAX JavaScript asíncrono y XML (del término en inglés *Asynchronous JavaScript And XML*)

CASE Ingeniería de Software Asistida por Computadora (del término en inglés *Computer Aided Software Engineering*)

UML Lenguaje Unificado de Modelado (del término en inglés *Unified Modeling Language*)

MIT (del término en inglés *Massachusetts Institute of Technology*)

API Interfaz de Programación de Aplicaciones (del término en inglés *Application Programming Interface*)

GRASP Patrones Generales de Software para Asignación de Responsabilidades (del término en inglés *General Responsibility Assignment Software Patterns*)

GOF Pandilla de Cuatro (del término en inglés *Gang Of Four*)

Referencias Bibliográficas

- **ALEGSA. 2011.** [En línea] 2011. <http://www.alegsa.com.ar/>.
- **Alonso, G, y otros. 2004.** *Web Services Concepts, Architectures and Applications*. s.l. : Springer, 2004.
- **Antunez, Romanuel Ramón. 2012.** *Desarrollo de Sistemas de Información Geográficos sobre una plataforma soberana*. Habana : s.n., 2012.
- **Apache, Group. 2015.** Apache Server Foundation. [En línea] 2015. <http://www.apache.org/>.
- **APR. 2015.** Aprende A Programar. [En línea] 2015. <http://www.aprenderaprogramar.com/>.
- **Arce Mesén, Rafael. 2000.** *Guía para el curso de Mapinfo*. Costa Rica : s.n., 2000.
- **Carnegie Mellon University. 2013.** School of Computer Science. [En línea] 2013. <http://www.cs.cmu.edu/>.
- **Castillo Reyes, Grethell . 2012.** Biblioteca. [En línea] Mayo de 2012. [Citado el: 15 de Noviembre de 2014.] <http://biblioteca.uci.cu/>.
- **Catalán Vega, Marcos Antonio. 2011.** *Metodologías de evaluación de Interfaces Gráficas de Usuario*. Madrid : s.n., 2011.
- **Committee, Software Engineering Standards. 2000.** *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. New York : s.n., 2000.
- **Confederación de Empresarios de Andalucía. 2014.** IDEAndalucía. [En línea] 2014.
- **Dávila Martínez, Francisco J.** *Introducción a los Sistemas de Información Geográfica*. s.l. : Cartoteca.
- **Definiciones. 2014.** definicion.de. [En línea] 2014. <http://definicion.de/mapa/>.
- **Documentación QGIS 2.0. 2006.** github.com. [En línea] 2006. <https://github.com/qgis>.

- **E-CENTRO. 2014.** Centrodeartigos.com. [En línea] 2014. [http://centrodeartigo.com/revista-digital-educacion-tecnologia-educativa/Cartografía digital, Historia, Recopilación de Datos, Funciones y Uso.ht](http://centrodeartigo.com/revista-digital-educacion-tecnologia-educativa/Cartografía%20digital,%20Historia,%20Recopilación%20de%20Datos,%20Funciones%20y%20Uso.ht).
- **Esri. 2013.** ArcGis Resources. [En línea] 2013. <http://resources.arcgis.com/es/help/main/10.1/index.html#//0154000004w5000000>.
- **—. 2015.** GIS Dictionary. [En línea] 2015. <http://support.esri.com/en/knowledgebase/Gisdictionary/browse>.
- **Frederick, Shea. 2010.** *Learning ExtJS*. 2010.
- **Friña Iglesias, Verónica, Luaces, Miguel y Trillo, David.** *Servicio de impresión de información geográfica en forma de mapas siguiendo el estándar OGC WPS*. Coruña : Campus de Elviña.
- **Geográfica, Sistemas de Información. 2013.** sites.google.com. [En línea] 2013. <https://sites.google.com/site/sigarcgis/home>.
- **Jacobson, I. 2000.** *El proceso unificado de desarrollo software*. Madrid : Series Editors, 2000.
- **Javier Garzas. 2013.** [En línea] 2013. <http://www.javiergarzas.com/2013/04/que-es-uml-diagramas-uml.html>.
- **Lara García, Marisol. 2013.** *Ingeniería de Software 2.[Presentación electrónica]*. 2013.
- **mapas, Los tipos de referencia de. 2014.** Travel. [En línea] 2014. <http://es.kllvx.com/travel-preparation/maps/1008005217.html>.
- **MapServer. 2015.** MapServer Open Source Web Mapping. [En línea] 2015. <http://mapserver.org/>.
- **Pantoja Zaldívar, Yoenis. 2011.** GeoNews. [En línea] 13 de Diciembre de 2011. [Citado el: 13 de Noviembre de 2014.] <http://revistageonews.wordpress.com>.
- **Pressman, Roger S. 5ta Edición.** *Ingeniería de Software. Un enfoque práctico. 5ta Edición*.
- **Product, Digital Map. 2013.** DigMap. [En línea] 2013. <http://www.digitalmapproducts.com/>.
- **RAE. 2010.** Diccionario de Lengua Española. [En línea] 2010. [Citado el: 15 de Noviembre de 2014.] <http://www.rae.es/rae.html>.

- **Ramírez Gonzalez, Alberto. 2008.***Metodología de la Investigación Científica*. Colombia : s.n., 2008.
- **Romero, Blanca Mayra Lastres y Rico Blanco, Luis Orlando. 2013.***Módulo para la edición de una ontología del dominio geográfico en la plataforma GeneSIG*. La Habana : s.n., 2013.
- **SAIG. 2009.***Kosmo Desktop v2.0*. Sevilla : s.n., 2009.
- **Sánchez Suárez, José Manuel. 2015.** Adictos Al Trabajo. [En línea] 2015. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=openlayers>.
- **TELMEX. 2013.** Academica Comunidad Digital de Conocimiento. [En línea] 2013. <http://www.academica.mx/>.
- **UCID, PDS. 2012.***Proceso de Desarrollo y Gestión de Proyectos de Software (Versión 1.5)*. 2012.

Anexos

```
public function crearImagen($elemento)
{
    1  if($elemento->transparencia==100)
    {
    2      $t=1.0;
    }
    else
    {
    3      $t= $elemento->transparencia/100;
    }
    //print_r($elemento);
    //die();
    4  $imagen='<div style="position:absolute; left:'. $elemento->x.'px; top:'. $elemento->y.'
    4  $imagen.=' src="'. $elemento->icono.'";
    4  $imagen.=' width="'. $elemento->anchura.'";
    4  $imagen.=' height="'. $elemento->altura.'";
    4  $imagen.=' style="opacity: '.$t.'";';

    4  $imagen.=' ></div>';
    5  return $imagen;
}
```

Fig. 10 Anexo 1: Crear Imagen