



**Universidad de las Ciencias Informáticas**

**Facultad 6**

**Título: Desarrollo de un sistema de información geográfica sobre  
Dracones.**

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

**Autores**

Yeneisy Santoya Chávez

Mónica Rodríguez Amador

**Tutor**

Ing. Romanuel Ramón Antunez

*La Habana 2011*

---

*El conocimiento nos hace responsables...*

*Ché.*

---

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de diploma y reconocemos al Centro de Geoinformática y Señales Digitales de la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio del año 2011.

Yeneisy Santoya Chávez

Ing. Romanuel Ramón Antunez

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

Mónica Rodríguez Amador

\_\_\_\_\_  
Firma del Autor

---

## OPINIÓN DEL TUTOR

El tutor del presente Trabajo de Diploma considera que durante su ejecución las estudiantes mostraron las cualidades que a continuación se detallan:

Presentaron un alto grado de independencia al asumir las tareas individuales asignadas. La dedicación y seriedad con que asumieron la investigación trajeron como resultado que su trabajo se convirtiera en referencia para otros estudiantes.

Alcanzaron un profundo dominio del campo donde se realizó la investigación. En la documentación generada se destaca la originalidad y creatividad expuesta por los autores.

La herramienta obtenida como resultado es robusta y los resultados arrojados a raíz de las pruebas hechas fueron satisfactorios.

En el trabajo realizado se puede apreciar una alta calidad científico-técnica. Los resultados obtenidos presentan un elevado nivel de rigurosidad y formalismo, lo cual corrobora que el objetivo propuesto al inicio de la investigación fue alcanzado.

Por todo lo anteriormente expresado se considera que ambas estudiantes están aptas para ejercer como Ingeniero en Ciencias Informáticas, y se propone:

Calificación máxima de 5 puntos a la diplomante Mónica Rodríguez Amador.

Calificación máxima de 5 puntos a la diplomante Yeneisy Santoya Chávez.

---

## AGRADECIMIENTOS

*El desarrollo del presente trabajo de diploma no ha sido solo un esfuerzo propio, sino que ha involucrado de una forma u otra a muchas personas. Gracias a todos por brindarme su apoyo y cooperación. Le agradezco principalmente:*

*A mis padres por hacer de mi la persona que soy hoy, gracias por darme todo y más de lo que he necesitado, gracias por cuidarme, por entenderme y estar ahí apoyándome siempre.*

*A todos mis profesores, que con su quehacer diario han contribuido a formar en mí, valores y a construir conocimientos que me han ayudado durante estos cinco años de carrera para ser una mejor profesional.*

*A todos mis compañeros y amigos, que compartieron conmigo tantos momentos de alegría y tristeza y que me ayudaron a salir adelante con sus excelentes consejos.*

*A todos mis familiares por su dedicación y preocupación.*

*“Gracias a todos, los tendré siempre presente.”*

*Yeneisy Santoya Chávez.*

---

## AGRADECIMIENTOS

*Si comenzara a enumerar a todas las personas que de una manera u otra han contribuido para formarme como futura profesional, necesitaría tres trabajos de diploma para reflejarlo. A todos ellos los recordaré en cualquier etapa de mi vida con mucho cariño.*

*Agradezco sin dudas a todos mis maestros, por su entrega en cada clase, por su dedicación y valores creados en mí. A todos ellos les guardaré profundo respeto.*

*No puedo dejar de mencionar a tres personas muy especiales durante mi carrera, por sus consejos y apoyo incondicional en cada momento. Rosa Elena, Lisandra y Dayron, los mantendré siempre en mi corazón.*

*Pero sobre todas las cosas le agradezco a mi mamá, y a ti pipo, por saber educarme de la mejor manera, por guiarme y comprenderme siempre. Ustedes son mi mayor tesoro, mi fuerza emprendedora.*

*A mi hermanita, por confiar tanto en mí, y por esforzarse en sus estudios teniéndome como ejemplo, eso es una extrema recompensa para mi vida. Te adoro.*

*A ti mi vida, que te puedo decir, si has sido mi compañero y amigo. Me has mostrado amor, cariño, respeto y apoyo en cualquier circunstancia. Te estaré eternamente agradecida por tanta paciencia.*

*Mónica Rodríguez Amador*

---

## DEDICATORIA

*El presente trabajo de diploma va dedicado a las personas más importantes de mi vida, a las que más quiero y respeto por encima de todo, mis padres. Mis padres han sido todo para mí, mi guía, mi sostén, mi inspiración, las personas que menos me gustaría defraudar en este mundo y sé que con este título de Ingeniera en Ciencias Informáticas los estoy haciendo, sin lugar a dudas, las personas más felices del mundo, por eso por ustedes y para ustedes, es todo mi esfuerzo y este título.*

*Yeneisy Santoya Chávez*

*Dedico el presente trabajo de diploma a las personas más importantes para mi vida. Por ustedes, mami, pipo y tata es que me mantengo con fuerzas y me motivo en cada acto que realizo. Les tengo todo el amor y respeto del mundo. Nunca me atrevería a defraudarlos. Sé que con la obtención de este título estoy cumpliendo uno de sus mayores sueños.*

*Mónica Rodríguez Amador*

---

## RESUMEN

Como parte de todas las transformaciones que se han manifestado a lo largo de la evolución humana, la necesidad de comunicación del hombre es uno de los aspectos más relevantes. El hecho de querer expresarse provocó que surgieran las primeras representaciones pictográficas, que más tarde a raíz de la propia evolución del hombre, fueron transformándose dando lugar a los primeros mapas geográficos. En la actualidad dichos mapas pueden ser aplicados en muchas esferas de la sociedad, pues la información que brindan los mismos se torna cada vez más importante, amplia y exacta. Los Sistemas de Información Geográfica (SIG), definidos según Salinas como una integración organizada de hardware, software y datos geográficos diseñados para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada; han logrado posicionarse en los primeros lugares en cuanto a aplicación cartográfica se refiere.

En correspondencia al auge que han tenido estos sistemas, la Universidad de las Ciencias Informáticas (UCI) cuenta con un centro, que en gran medida cubre el desarrollo de sistemas de información geográfica. A fin de contar con varias propuestas documentadas sobre el tema y de profundizar en el estudio y empleo de nuevas tecnologías, se pretende desarrollar un SIG sobre el marco de trabajo Dracones. El mismo hoy no cuenta con la posibilidad de construir mapas temáticos. Partiendo de que los mismos son el área de mayor crecimiento dentro del campo de la cartografía y que en esencia son el producto final de un SIG. El equipo de desarrollo se propone llevar a cabo la implementación de una aplicación SIG utilizando herramientas y tecnologías libres para conseguirlo.

-Palabras claves: mapas temáticos, SIG, Dracones.

---

## ABSTRACT

As part of all the transformations that have been expressed throughout human evolution, the need for communication of man is one of the elements most relevant. The need of expression caused that arise early pictographs, which later became the first maps. At present these maps can be applied in many areas of society, because the information that provide becomes is increasingly important, comprehensive and accurate. The Geographic Information Systems (GIS), defined by Salinas as an organized integration of hardware, software and geographic data designed to capture, store, manipulate, analyze and display all forms the geographically referenced information; have been positioned on top of geographic applications.

Corresponding to the development that has these systems, the University of Informatics Sciences (UCI) has a center, which largely covers the development of geographical information systems. In order to have documented several proposals on the subject and use of new technologies, we will develop a GIS on the Dracon framework. Actually the same does not have the possibility of constructing thematic maps. Given that these are the fastest growing area within the field of cartography and in essence are the end product of a GIS. The development team intends to carry out the implementation of a GIS application using tools and free technologies to get the same

-Key words: thematic map, GIS, Dracones.

---

## TABLA DE CONTENIDOS

INTRODUCCIÓN .....	15
Capítulo 1: Fundamentación teórica.....	20
1.1 INTRODUCCIÓN .....	20
1.2 DEFINICIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG) .....	20
1.3 DEFINICIÓN DE MAPA TEMÁTICO .....	21
1.4 CLASIFICACIÓN DE LOS MAPAS TEMÁTICOS .....	21
1.5 CARACTERIZACIÓN DE ALGUNAS HERRAMIENTAS EXISTENTES PARA LA CREACIÓN DE MAPAS TEMÁTICOS.....	25
1.5.1 DESCRIPCIÓN DE QUANTUMGIS .....	25
1.5.2 DESCRIPCIÓN DE UDIG .....	25
1.5.3 DESCRIPCIÓN DE MAPINFO 8.5 .....	26
1.5.4 DESCRIPCIÓN DE UP2MAPS .....	27
1.6 CARACTERIZACIÓN DEL MARCO DE TRABAJO DRACONES.....	27
1.7 CONCLUSIONES .....	28
Capítulo 2: Soporte tecnológico.....	29
1.1 INTRODUCCIÓN .....	29
1.2 VALORACIÓN DE LAS TECNOLOGÍAS DISPONIBLES PARA EL DESARROLLO DE SIG CON DRACONES. DESCRIPCIÓN DE CADA UNA DE ELLAS .....	29
1.2.1 DESCRIPCIÓN DE PYTHON .....	29
1.2.2 DESCRIPCIÓN DE PHP .....	30
1.2.3 DESCRIPCIÓN DE NETBEANS .....	31
1.2.4 DESCRIPCIÓN DE VISUAL PARADIGM .....	32
1.3 DESCRIPCIÓN DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE EXISTENTES. 33	
1.3.1 METODOLOGÍAS TRADICIONALES O PESADAS.....	33
1.3.1.1 DESCRIPCIÓN DE RUP .....	34
1.3.2 METODOLOGÍAS ÁGILES.....	35
1.3.2.1 DESCRIPCIÓN DE XP .....	36
1.3.2.2 DESCRIPCIÓN DE FDD .....	37
1.4 SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE A UTILIZAR .....	39
1.5 CONCLUSIONES .....	40

Capítulo 3: Análisis de la solución propuesta.....	41
1.1 INTRODUCCIÓN .....	41
1.2 MODELO GLOBAL .....	41
1.3 GLOSARIO DE TÉRMINOS ASOCIADOS AL DOMINIO .....	41
1.4 DIAGRAMA GENERAL .....	42
1.5 CONSTRUCCIÓN DE LA LISTA DE RASGOS .....	43
1.6 PLANEACIÓN POR RASGOS.....	46
1.7 RASGOS NO FUNCIONALES.....	47
1.8 ARQUITECTURA QUE UTILIZARÁ EL SISTEMA.....	50
1.9 DESCRIPCIÓN DE LA ARQUITECTURA A EMPLEAR PARA LA IMPLEMENTACIÓN DEL SISTEMA .....	50
1.10 PATRONES DE DISEÑO.....	51
1.10.1 SOLITARIO.....	51
1.10.2 ACCIÓN .....	51
1.11 PATRONES GRASP.....	51
1.11.1 BAJO ACOPLAMIENTO .....	52
1.11.2 ALTA COHESIÓN .....	52
1.12 DIAGRAMA DE CLASES DEL DISEÑO DE LOS COMPONENTES .....	52
1.13 DESCRIPCIÓN DE LAS CLASES DEL DISEÑO .....	53
1.14 MODELO DE DESPLIEGUE .....	56
1.15 CONCLUSIONES .....	57
Capítulo 4: Validación.....	58
1.1 INTRODUCCIÓN .....	58
1.2 PRUEBAS DEL SISTEMA PROPUESTO .....	58
1.3 PRUEBAS DEL CAJA NEGRA .....	58
1.4 CASOS DE PRUEBAS.....	59
1.4.1 DESCRIPCIÓN GENERAL DEL RASGO TEMATIZAR POR COLORES .....	59
1.4.1.1 SECCIONES A PROBAR.....	59
1.4.1.2 DESCRIPCIÓN DE VARIABLES.....	60
1.4.1.3 MATRIZ DE DATOS .....	60
1.4.2 DESCRIPCIÓN GENERAL DEL RASGO TEMATIZAR POR GRÁFICA.....	65
1.4.2.1 SECCIONES A PROBAR.....	65

---

1.4.2.2 DESCRIPCIÓN DE VARIABLES.....	65
1.4.2.3 MATRIZ DE DATOS .....	66
1.4.3 DESCRIPCIÓN GENERAL DEL RASGO CONTROL DE CAPAS.....	69
1.4.3.1 SECCIONES A PROBAR.....	69
1.4.3.2 DESCRIPCIÓN DE VARIABLES.....	70
1.4.3.3 MATRIZ DE DATOS .....	70
1.4.4 DESCRIPCIÓN GENERAL DEL RASGO REALIZAR NAVEGACIÓN.....	70
1.4.4.1 SECCIONES A PROBAR.....	70
1.4.4.2 DESCRIPCIÓN DE VARIABLES.....	71
1.4.4.3 MATRIZ DE DATOS .....	73
1.4.5 DESCRIPCIÓN GENERAL DEL RASGO IDENTIFICAR OBJETO.....	75
1.4.5.1 SECCIONES A PROBAR.....	75
1.4.5.2 DESCRIPCIÓN DE VARIABLES.....	76
1.4.5.3 MATRIZ DE DATOS .....	76
1.5 RENDIMIENTO DEL SISTEMA.....	77
1.6 CONCLUSIONES .....	78
CONCLUSIONES .....	79
RECOMENDACIONES.....	80
REFERENCIAS BIBLIOGRÁFICAS .....	81
BIBLIGRAFÍA.....	82
ANEXOS .....	83

---

## ÍNDICE DE FIGURAS

Figura 1: Fases de desarrollo de la metodología FDD .....	37
Figura 2: Diagrama general .....	42
Figura 3: Diagrama de clases del diseño .....	53
Figura 4: Diagrama de despliegue .....	57

---

## ÍNDICE DE TABLAS

Tabla 1: Listado de rasgos .....	46
Tabla 2: Planeación de los rasgos para la navegación .....	47
Tabla 3: Planeación de los rasgos para la identificación .....	47
Tabla 4: Planeación de los rasgos para análisis.....	47
Tabla 5: Descripción de la clase Draconesph_do .....	55
Tabla 6: Descripción del fichero Core .....	55
Tabla 7: Descripción de la clase Dmap.....	56
Tabla 8: Secciones a probar en tematizar por colores .....	59
Tabla 9: Descripción de variables en tematizar por colores .....	60
Tabla 10: Matriz de datos en tematizar por colores .....	65
Tabla 11: Secciones a probar en tematizar por gráficas .....	65
Tabla 12: Descripción de variables en tematizar por gráficas.....	66
Tabla 13: Matriz de datos en tematizar por gráficas .....	69
Tabla 14: Secciones a probar en control de capas .....	70
Tabla 15: Descripción de variables en control de capas .....	70
Tabla 16: Matriz de datos en control de capas .....	70
Tabla 17: Secciones a probar en realizar navegación .....	71
Tabla 18: Descripción de variables en realizar navegación.....	73
Tabla 19: Matriz de datos en realizar navegación .....	75
Tabla 20: Secciones a probar en control de capas .....	76
Tabla 21: Descripción de variables en control de capas .....	76
Tabla 22: Matriz de datos en control de capas .....	77
Tabla 23: Rendimiento del sistema .....	77

---

## INTRODUCCIÓN

A través de los años el hombre pudo distinguirse del animal por innumerables rasgos. El mismo supo manifestarse de diferentes formas, entre las cuales resultaron importantes aquellas que motivaban sus necesidades más vitales de alimentación, salud, vestido y cobijo. Al no disponer más que de piedras que se han conservado mediante el paso del tiempo, entre otros elementos naturales a su vista. El mismo se auxilió de representaciones pictóricas para comenzar a comunicarse y más tarde convertir sus dibujos en planos históricos que fueron transformándose dando lugar a los primeros mapas geográficos.

Dichos gráficos inicialmente mostraban diversidad de información, la cual trataba diferentes y necesarias temáticas según los tiempos y la sociedad en que se vivía. En tiempos remotos los grabados en las cuevas evidenciaban su manera de subsistir y la ubicación más o menos inmediata del lugar en donde se encontraban; además de reflejar caminos y posibles itinerarios de los antiguos ejércitos. Luego, a raíz de productivas investigaciones y de la propia evolución del hombre, los temas tratados son un tanto más cualitativos y relevantes. Los mismos se traducen en datos con distintas clasificaciones que logran ser materializados en nuevos lenguajes para una nueva etapa gráfica en la historia.

Los primeros mapas surgidos reflejan desde las diferentes formas en que se concebían el planeta, hasta distancias existentes de un lugar a otro, entre otras clasificaciones. Siendo parte todas ellas de los antecedentes que contribuyeron a la conceptualización de entornos gráficos, así como a la transformación del paisaje al lenguaje computacional.

Gracias al desarrollo de las tecnologías de la información y las comunicaciones muchos de los temas o datos que antes apenas se conocían, actualmente pueden ser tratados. Pues es altamente efectivo destacar que dichas tecnologías han originado, progresado y facilitado al mundo actual una amplia gama de posibilidades. Las tecnologías de la información y las comunicaciones están presentes en nuestras vidas y la han transformado. (10di)

Uno de los grandes avances en la evolución de las tecnologías de la información y las comunicaciones, lo constituyen los Sistemas de Información Geográfica (SIG). Los mismos datan desde los años sesenta donde se necesitaba mecanizar tareas pesadas de producción cartográfica ligada a los sistemas digitales. Donde ya, para los años

---

setenta, una empresa de trenes de Irlanda consigue la organización de un sistema de información geográfica para emplear en la superposición de acetatos. Lográndose posteriormente con marcada eficiencia el SIG-Canadá, orientado al manejo de bosques y estructurado en polígonos.

Como bien se ha podido comprobar lo que antes se utilizaba como parte de medios de comunicación, y que luego se tradujo a diferentes particularidades cartográficas, transformadas en mapas bonitos. Hoy se materializa a través de sistemas de información geográfica. Definiéndose los mismos como una integración organizada de hardware, software y datos geográficos diseñados para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También pueden definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. (Salinas, 2007)

La tecnología de los Sistemas de Información Geográfica puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística por nombrar unos pocos. Por ejemplo, un SIG podría permitir a los grupos de emergencia calcular fácilmente los tiempos de respuesta en caso de un desastre natural, el SIG puede ser usado para encontrar los humedales que necesitan protección contra la contaminación, o pueden ser utilizados por una empresa para ubicar un nuevo negocio y aprovechar las ventajas de una zona de mercado con escasa competencia (Dieguez, 2003).

Los sistemas de información geográfica pueden ser desarrollados sobre disímiles marcos de trabajo. Los cuales también han ido apareciendo y avanzando con el gran impacto de las tecnologías de la información y las comunicaciones. La evolución es un proceso constante y progresivo, donde el hombre cuenta con la suficiente capacidad para querer idear nuevas tecnologías y adaptarlas a sus más frías necesidades. Todo ello ha propiciado que en los tiempos actuales se pueda contar con poderosas herramientas tecnológicas que además de contribuir a satisfacer las distintas necesidades humanas; promueven y perfeccionan los orígenes geográficos que antes no podían ser explicados y ahora forman parte de grandiosos descubrimientos.

---

Dracones es uno de los ejemplos de marcos de trabajo. El mismo fue desarrollado por un grupo de especialistas entre los que se destacan las figuras de David Buckeridge y Cristiano Jauvin. Hoy el mismo no cuenta con ningún módulo que brinde la posibilidad de construir mapas temáticos.

Partiendo de que dichos mapas persiguen objetivos específicos, y que basados en los mapas topográficos, representan cualquier fenómeno de la superficie terrestre que sea cartografiable. Además de considerar que son el área de mayor crecimiento dentro del campo de la cartografía; que hacen referencia a la representación de ciertas características de distribución, relación, densidad o regionalización de objetos reales (vegetación, suelos, geología), o de conceptos abstractos (indicadores de violencia o de calidad de vida).

Se define como **problema científico de la investigación**: ¿Cómo lograr la representación de información estadística de variables según su distribución espacial?

Partiendo del problema científico antes mencionado el **objeto de estudio** estaría encaminado a los sistemas de información geográfica.

El **campo de acción** se centra en la creación de mapas temáticos en Sistemas de información geográfica.

Definiéndose para la investigación como **objetivo general** la idea de: Desarrollar un SIG con software libre que permita crear mapas temáticos dinámicos.

**La idea a defender** plantea que: Con el desarrollo de un Sistema de Información Geográfica que permita crear mapas temáticos, se logra un mejor análisis de la distribución espacial de los fenómenos.

Para hacer cumplir los objetivos que se han definido anteriormente, se han planteado las siguientes **tareas de la investigación**:

1. Caracterizar los mapas temáticos.
2. Caracterizar herramientas existentes para la creación de mapas temáticos.
3. Caracterizar el marco de trabajo Dracones.
4. Valorar las tecnologías disponibles para el desarrollo de SIG con Dracones.
5. Describir las tecnologías a emplear para la implementación del sistema.
6. Seleccionar la metodología de desarrollo de software a utilizar.
7. Implementar aplicación simple con el marco de trabajo Dracones.

- 
8. Establecer las funcionalidades que brindará el sistema así como la arquitectura que utilizará.
  9. Describir la arquitectura a emplear para la implementación del sistema.
  10. Diseñar el sistema.
  11. Implementar los Casos de Usos del sistema.
  12. Probar los Casos de Usos del Sistema Implementados.
  13. Documentar el sistema.

Durante el desarrollo de la investigación se ponen de manifiesto una serie de métodos, tanto teóricos como empíricos. A continuación se identifican cada uno de ellos, mencionando las etapas en las cuáles se llevan a cabo con mayor envergadura.

Entre los **métodos teóricos** se evidencia el **histórico-lógico** a la hora de conocer la evolución de los sistemas de información geográfica, así como los marcos de trabajo, específicamente todo lo relacionado con Dracones. Además de conocer acerca de los mapas temáticos y algunas de las herramientas existentes para la construcción de los mismos.

El método **analítico-sintético** en el momento que se propone analizar por separado toda la información recopilada a partir de las partes que componen el objeto de estudio en cuestión con el fin de sintetizar la documentación obtenida para comprender las relaciones de las partes que se auto complementan en la práctica.

Como parte de los **métodos empíricos** se evidencian las entrevistas realizadas al tutor y otros especialistas de la plataforma GeneSIG, con el fin de contar con conocimientos sólidos acerca de la misma, para poder llevar a cabo la propuesta de nuevas tecnologías en el desarrollo de sistemas de información geográfica.

En la presente investigación se han identificado como posibles **resultados a obtener** un prototipo funcional de aplicación SIG en la web con tecnologías libres utilizando Dracones y la documentación técnica del proceso de desarrollo.

El presente trabajo se divide de la siguiente manera:

### **Capítulo 1. Fundamentación teórica de los Sistemas de Información Geográfica para la construcción de mapas temáticos**

En este capítulo se abordará acerca de la fundamentación teórica que sustenta la actual investigación. Ello incluye el análisis y descripción de algunas de las herramientas que existen para crear mapas temáticos. Donde se precisarán con

---

anterioridad los conceptos asociados con el dominio del problema para el logro de un mejor entendimiento del mismo. Finalmente se reflejarán las principales características y otras particularidades sobre el marco de trabajo Dracones, en cual se apoya la investigación para lograr cumplir con los objetivos planteados.

## **Capítulo 2. Soporte tecnológico**

En este capítulo se mostrará una valoración acerca de las tecnologías presentes, que permiten desarrollar sistemas de información geográfica con el marco de trabajo Dracones. A continuación se describirán las tecnologías que se emplearán para lograr implementar el sistema. Lográndose por último la selección de una metodología de desarrollo de software que guíe el proceso en cuestión. Para ello se realizará un estudio y descripción de los dos importantes grupos de metodologías que existen particularizando en cada una de ellas.

## **Capítulo 3. Construcción de la solución propuesta**

En el tercer capítulo, además de precisarse los detalles de la arquitectura a utilizar para el desarrollo del sistema. Se identificará el dominio del problema, construyéndose para ello un modelo global. Conjuntamente se propondrán los rasgos tanto funcionales como no funcionales, y se generarán las descripciones y artefactos necesarios que propone la metodología de desarrollo de software seleccionada en el capítulo anterior. En conclusión se construirá el análisis y diseño de la solución propuesta.

## **Capítulo 4. Validación**

En este capítulo se llevará a cabo la validación del sistema, ejecutándose para ello las pruebas que se requieren. Entre ellas se adoptaran las pruebas de caja negra, además de incluir algunos elementos acerca del rendimiento del sistema. A fin de garantizar la calidad del software a desarrollar, verificando que los rasgos planeados inicialmente son cumplidos; se ejecutaran dichas pruebas mencionadas.

---

# Capítulo 1: Fundamentación teórica

## 1.1 INTRODUCCIÓN

En este capítulo se presentan todos los conceptos asociados al dominio de problema e incluye además un análisis de las herramientas existentes para la creación de mapas temáticos. También se pretende caracterizar el marco de trabajo Dracones a fin de reflejar los elementos generales y particulares del software en el cual se apoya la investigación para darle solución al problema existente.

## 1.2 DEFINICIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIG)

Un SIG puede definirse como: Un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión.

Otra definición de SIG, es la dada por el Instituto Geográfico Agustín Codazzi que dice: Un SIG es un conjunto de métodos, herramientas y actividades que actúan coordinada y sistemáticamente para recolectar, almacenar, validar, actualizar, manipular, integrar, analizar, extraer, y desplegar información, tanto gráfica como descriptiva de los elementos considerados, con el fin de satisfacer múltiples propósitos.

Estos sistemas son usados en infinidad de áreas como por ejemplo:

- ✚ Investigaciones científicas.
- ✚ Gestión de recursos.
- ✚ Gestión de activos.
- ✚ Arqueología.
- ✚ Evaluación del impacto ambiental.
- ✚ Planificación urbana.
- ✚ Cartografía.
- ✚ Sociología.
- ✚ Geografía histórica.
- ✚ Marketing.
- ✚ Logística.

La manera en que los SIG integran la información es a través de capas o coberturas de datos que se van superponiendo unas con otras, según la información que se

---

requiera. De la integración por superposición de estas informaciones se obtiene un mapa temático.

### 1.3 DEFINICIÓN DE MAPA TEMÁTICO

**Los mapas temáticos:** son una representación gráfica de un tema específico, por ejemplo: mapas de densidad de población, de flujos migratorios, de distribución de escuelas, entre otros, que son en esencia el producto final de un SIG (Gaviño, 1999).

Dentro de la gran variedad de mapas que existen en el mundo, unos de los más estudiados y conocidos son los mapas temáticos. Existen múltiples mapas de este tipo especializados en diversas materias, los cuales están normalmente interrelacionados.

Los mismos son mapas que, basados en los mapas topográficos, representan cualquier fenómeno de la superficie terrestre que sea cartografiable. Persiguen objetivos bien definidos. Hacen referencia a la representación de ciertas características de distribución, relación, densidad o regionalización de objetos reales (vegetación, suelos, geología), o de conceptos abstractos (indicadores de violencia o de calidad de vida) (Gaviño, 1999).

### 1.4 CLASIFICACIÓN DE LOS MAPAS TEMÁTICOS

Los mapas temáticos tienen disímiles clasificaciones. Entre las tantas que se estudian y se adoptan se conocen: (Buzo Sánchez)

❖ **Mapas puntuales:** Localizan la información en puntos concretos del mapa: ciudades, población, industrias. De estos mapas también se hallan variantes:

➤ **Mapa de símbolos proporcionales.** Dentro de ellos se encuentran:

- **Lineales:** Localizan la información de manera lineal. Son útiles para representar conexiones entre puntos, tipos de redes, movimientos entre lugares (flujos) y mapas de isolíneas, o de alturas iguales (aquellos mapas que las líneas unen puntos con la misma altura, presión, precipitaciones). Estos a su vez se encuentran divididos en varias categorías:
  - Mapa de líneas o redes: Son aquellos mapas que representan las relaciones existentes entre diferentes lugares del mapa (red de

---

carreteras, ferrocarril, relaciones comerciales, entre otros.). Cada una de ellas se representa con una línea de un color diferente.

- Mapa de flujos: Son aquellos mapas sobre los que se representan variables en movimiento mediante líneas o flechas que unen los puntos de origen y destino, siendo el tamaño proporcional a la intensidad del fenómeno representado. Mapa de flujos es en el que se representan las rutas de migración de las cigüeñas negras. En este caso en particular al representarse movimientos animales se denomina “mapa fenológico”.
- Mapa de isolíneas o isopletas<sup>1</sup>: Son aquellos mapas que representan las variables mediante líneas imaginarias que unen puntos con el mismo valor. Ejemplo son los mapas de isobaras<sup>2</sup>, isoyetas<sup>3</sup>, isotermas, batimétricos, topográficos, entre otros. En el mapa donde las isolíneas o isopletas unen puntos de igual presión; reciben el nombre de isobaras.
- **De superficies**: Localizan la información que se extiende a lo largo del territorio. La tipología varía según las unidades de superficie, si las definimos antes de la medición (Coropletas y anamórfico) o es la propia medición la que define la superficie a representar (corocromático).
- Mapa de coropletas: Son aquellos que utilizan colores o tramas para representar cualquier fenómeno sobre un mapa dividido en unidades de superficie (comarcas, provincias, países, entre otros).
- Mapa anamórfico<sup>4</sup>: Son aquellos mapas que representan la superficie de un lugar en proporción al tamaño de la variable estudiada. Son mapas esquemáticos, que no guardan ni la forma ni la escala real. Mapa en el que se muestra la densidad de población mundial, haciéndola proporcional al tamaño del continente que representa.

---

<sup>1</sup> Isopletas: Se denomina así a la línea que une los puntos en los que una cierta magnitud es constante, en un mapa o diagrama.

<sup>2</sup> Isobaras: Un isobaras es un isograma, una línea de presión igual o constante en un gráfico, mapa o ruta.

<sup>3</sup> Isoyetas: Es una línea trazada sobre un mapa sinóptico con la que se unen puntos (representación de una estación meteorológica), donde se registra igual cantidad de precipitación.

<sup>4</sup> Anamórfico: Proceso por el cual una imagen panorámica 16:9 se comprime horizontalmente para generar una imagen 4:3, con el objetivo de preservar el máximo de resolución vertical posible.

- 
- Mapa corocromático: Son aquellos mapas que utilizan colores o tramas para expresar cualidades del terreno (tipos de suelo, unidades de relieve, climas, entre otras.). Se diferencia del mapa de coropletas en que en este caso no se representan unidades de superficie. Es un mapa en el que se muestran los diferentes climas del mundo, cada uno representado por un color. En este caso no hay unidades superficiales previas, como en el caso de los mapas de coropletas.

Otra rama dentro de los mapas temáticos es el **mapa Geológico**. El mismo muestra la distribución de los distintos tipos de roca sobre el terreno, su forma y las relaciones entre ellos. Además, muestra otros datos adicionales como pueden ser su edad geológica y su estructura (pliegues y fallas). Los datos geológicos observables representados en el mapa geológico pueden ser tanto de carácter **cuantitativo como cualitativo**. (Buzo Sánchez)

El **mapa geotécnico** es aquel mapa geológico en el que mediante una investigación de la estructura tectónica de la corteza terrestre, composición de las rocas que forman la parte más superficial de la misma, análisis de los fenómenos geológicos actuales y con las experiencias habidas en otras zonas geológicas y geográficas similares; se establece una distribución de los procesos y fenómenos geotécnicos. Descubre los factores que rigen las condiciones geológicas para la construcción y predice los campos que en las condiciones geotécnicas pueden producir esas construcciones.

**El mapa hidrogeológico** es aquel que ofrece información sobre una visión general y sintética de la distribución espacial de los acuíferos subterráneos del territorio, sus condiciones de ubicación y características generales, supuestas o formuladas, del funcionamiento de estos acuíferos así como de sus eventuales interrelaciones. También presentan una evaluación de la potencialidad de los recursos subterráneos del territorio. Suele indicarse en los mismos también los puntos de agua y el quimismo de las mismas.

En los **mapas metalogenéticos** se representan, sobre una doble base topográfica y geológica simplificada, los yacimientos o indicios minerales conocidos. Proporcionan además información sobre el recurso mineral del que se trate en cada caso (minerales metálicos, no metálicos o recursos energéticos), así como sobre su morfología y volumen.

---

El mapa metalogenético también analiza el origen de los indicios, con el objeto de deducir los rasgos geológicos (fracturas, niveles estratigráficos<sup>5</sup>) que determinan su localización.

Los principales recursos nacionales de rocas de interés industrial se representan en los mapas de rocas industriales. Facilitan, por tanto, la localización de calizas, granitos u otras rocas necesarias para la construcción de obras civiles.

El **mapa tectónico** es aquel en el que se diferencian los principales elementos estructurales de la corteza, fundamentalmente fracturas y pliegues, y permite deducir las direcciones de los esfuerzos soportados por las rocas en una zona concreta.

El **mapa de suelos** indica el tipo de suelo existente y predominante en cada lugar. Ha sido muy discutida la clasificación que debe emplearse en este sentido, pues son numerosas las que han sido propuestas por distintos países sin que se haya llegado aún a una decisión definitiva.

El **mapa agronómico** es el que especifica la vegetación existente o la más apta para cada zona.

También pueden citarse otros **mapas temáticos de uso civil**:

1. Mapa de orientación al vertido de residuos sólidos urbanos.
2. Mapa sismotectónico de España.
3. Mapas de riesgos naturales ligados a movimientos del terreno.

En el ámbito militar también se le da un uso especial a los mapas temáticos. Dichos mapas en esta esfera tienen por finalidad proporcionar una información amplia y precisa del terreno y sus características, con expresión de las variaciones que pueda experimentar por la acción de agentes externos, para facilitar al mando el planteamiento y la dirección de las operaciones y a las tropas la ejecución de la maniobra, en las mejores condiciones.

Podría parecer que los mapas temáticos intentan suplantar a los topográficos, pero la idea es que se complementen para obtener mejores resultados. La principal diferencia entre ambos, es que en los primeros existe una gran variedad y profundidad en la

---

<sup>5</sup> Estratigráficos: Acontecimientos que se han sucedido en la superficie de la Tierra en el transcurso de los tiempos geológicos.

---

información proporcionada y en los fenómenos representados, mientras que en los segundos el número de fenómenos presentados es bastante pequeño.

## 1.5 CARACTERIZACIÓN DE ALGUNAS HERRAMIENTAS EXISTENTES PARA LA CREACIÓN DE MAPAS TEMÁTICOS

Entre las herramientas existentes para la creación de mapas temáticos se encuentran: Quantum GIS, MapInfo, Udig, Up2Maps entre otras. A continuación se describen cada una de ellas.

### 1.5.1 DESCRIPCIÓN DE QUANTUMGIS

Quantum GIS (o QGIS) es un Sistema de Información Geográfica (SIG) de código libre para plataformas Linux, Unix, Mac OS y Microsoft Windows. Era uno de los primeros ocho proyectos de la Fundación OSGEO (Open Source Geospatial Foundation)). Permite ver, editar y crear una variedad de formatos vectoriales y ráster, así como bases de datos. Puede crear plugins personalizados y aplicaciones habilitadas para SIG usando Python o C++. Está disponible bajo la licencia GNU (General Public License) Algunas de sus características son: (Tim Sutton, 2007)

- ✚ Soporte para la extensión espacial de PostgreSQL y PostGIS.
- ✚ Manejo de archivos vectoriales Shapefile, ArcInfo coverages, Mapinfo, GRASS GIS, entre otros.
- ✚ Soporte para un importante número de tipos de archivos ráster (GRASS GIS, GeoTIFF, TIFF, JPG, entre otros.).

Una de sus mayores ventajas es la posibilidad de poder usarse como GUI (graphical user interface) del SIG GRASS (Geographic Resources Analysis Support System), utilizando toda la potencia de análisis de este último en un entorno de trabajo más amigable. QGIS está desarrollado en C++, usando la biblioteca Qt para su interfaz gráfica de usuario. Su dificultad radica en que debe estar necesariamente instalado en la máquina donde se esté utilizando.

### 1.5.2 DESCRIPCIÓN DE UDIG

Es un visor/editor de datos espaciales que hace especial énfasis en cumplir los estándares OpenGIS para los servidores WMS (web map service) y WFS (web feature service). UDIG proporciona una plataforma común en Java para desarrollar aplicaciones espaciales utilizando componentes open source. La versión 0.5 incluye el

---

soporte para poder aplicar estilos a las capas y consigue una mayor velocidad en el renderizado. Entre sus características se encuentran: (Martín, 2007)

- ✚ Interfaz de usuario más "profesional"
- ✚ Basado en el framework de construcción de aplicaciones enriquecidas de Eclipse (RCP o Rich Client Framework), seguramente el framework de construcción de aplicaciones de escritorio más extendido en Java.
- ✚ Basado en Geotools, excelente librería de tratamiento de datos espaciales.
- ✚ Estabilidad y cumplimiento de estándares.

### 1.5.3 DESCRIPCIÓN DE MAPINFO 8.5

Con MapInfo Profesional 8.5 usted puede construir sobre su inversión previa en los sistemas desktop de MapInfo y darse cuenta de los beneficios de la "location intelligence" corporativa. Gracias a la integración mejorada, a compartir los datos y a la flexibilidad, los usuarios de MapInfo Profesional pueden ir más allá de su puesto de trabajo fácilmente para acceder a los potentes servicios de "location intelligence" (como geocodificación y análisis de distancias de conducción). (Jarochos, 2007)

MapInfo Profesional le permite conectarse automáticamente a Envinsa, la plataforma corporativa de MapInfo y otros servicios web externos basados en fuentes de datos; o vía XML a cualquier sistema corporativo. Esto permite a los usuarios visualizar, analizar y crear informes de relaciones entre todos los aspectos de su negocio, accediendo a los datos donde quiera que estén. Entre las mejoras claves de la nueva versión se encuentran:

- ✚ Da valor geográfico a los datos de una manera sencilla por medio de la conectividad con Envinsa y 'Envinsa Online Services' (EOLS).
- ✚ Permite soporte de la capacidad de generación de áreas de tiempos de conducción para creación de áreas efectivas de negocio y análisis a través de la conectividad con Envinsa y EOLS.
- ✚ Presenta opciones de presentación mejoradas, por ejemplo la capacidad de re-proyectar mapas que tienen Grid y/ o ráster.
- ✚ Mejoras en la facilidad de manejo y accesibilidad a funciones de uso frecuente por medio de una nueva barra de iconos, nuevos menús en el botón derecho, cuadros de diálogo más grandes y barra de estado.

---

Fundada en 1986, MapInfo es líder de mercado en herramientas de software para la visualización y análisis de datos corporativos a través de mapas. Esta multinacional dispone de una amplia gama de productos abiertos y escalables (que funcionan en PCs, servidores y entornos internet/extranet.), herramientas de desarrollo de aplicaciones, productos de datos demográficos y geográficos, y servicios de consultoría y soporte.

Las diferentes gamas de productos MapInfo han sido diseñadas para que los usuarios visualicen y examinen los datos desde una perspectiva geográfica, superponiéndolos en mapas digitales con diferentes niveles. De esta forma, es posible utilizar la información geográfica de una empresa (por ejemplo, ciudades, nombres de calles, códigos postales...) para descubrir modelos de comportamiento y tendencias que resultan imposibles de identificar de otra manera. Como resultado, se gana en comprensión del negocio, se mejoran las relaciones con el cliente y el rendimiento resulta mucho mayor, sobre todo en organizaciones con grandes infraestructuras físicas que poseen demasiada información y necesitan conocer las tendencias demográficas de un determinado segmento de mercado, gestionar activos y asignar recursos para competir con eficacia. (Jarochos, 2007)

#### 1.5.4 DESCRIPCIÓN DE UP2MAPS

Up2Maps es una interesante herramienta que da la posibilidad de crear mapas temáticos completamente personalizados para poder compartir los conocimientos con toda la comunidad de Internet. (Diego, 2009)

El servicio es muy fácil de utilizar ya que permite tipear la información en una tabla pre-fabricada o subir documentos de archivos de Excel, además ofrecen plantales de Excel para hacer todo el proceso incluso más simple. Una vez finalizado todo puedes exportar el mapa para insertar en tus reportes o presentaciones, donde una copia será almacenada en la librería del sitio. Aunque esta herramienta es en línea, la misma no permite agregar cartografía, por lo que las tematizaciones están sujetas a la cartografía existente en los servidores.

La meta de Up2Maps es construir entre todo un repositorio de conocimiento comunitario en cuanto a demografía social y otra información estadística. (Diego, 2009)

#### 1.6 CARACTERIZACIÓN DEL MARCO DE TRABAJO DRACONES

---

Dracones es un marco de trabajo desarrollado por un grupo de especialistas, entre los que se destacan las figuras de David Buckeridge y Cristiano Jauvin. El objetivo de su creación estuvo dirigido al desarrollo de una aplicación de mapeo web de código abierto destinado a la Salud Pública de Montreal. Su entidad o empresa de origen es McGill University (Surveillance Lab) y posee licencia BSD (Berkeley Software Distribution). Dracones es un software que maneja el idioma inglés, el formato HTML y niveles tanto de usuarios como de desarrolladores.

Como desventajas, se pueden mencionar que no cuenta con el apoyo de OSGEO (Open Source Geospatial Foundation), no incluye componentes de metadatos y no posee listas de correo. Sin embargo es un software incipiente que no requiere plug-ins privativos. El mismo está basado en PHP mapscript de UMN MapServer, tanto en Python, como en PHP. (Carrillo, 2010)

Dicho marco de trabajo posee componentes de servidor y de cliente. Utiliza JQuery del lado del cliente suministrando un control de mapas que trabaja con peticiones AJAX. Está disponible como código abierto y permite trabajar con ciertos atributos como (selección, visibilidad entre otros) de manera más fácil. Entre sus características técnicas se encuentran que Dracones está escrito en java script, php y python. Consume los servicios WMS (web map service) y WFS (web feature service) de la OGC (Open Geospatial Consortium) y su API admite java script, php y python como lenguaje de programación. (Carrillo, 2010)

## 1.7 CONCLUSIONES

En este capítulo se enunciaron los principales conceptos asociados con el problema en cuestión, lo cual permite un mejor entendimiento del mismo. Seguidamente se mostraron las disímiles clasificaciones acerca de los mapas temáticos, que son en esencia, el campo puntual donde se centra la investigación para dar cumplimiento a sus objetivos. Además se realizó un análisis particular de algunas de las herramientas existentes que permiten construir mapas temáticos. Con el propósito de reflejar los elementos específicos y generales del software en el cual se apoya el trabajo investigativo para solucionar el problema que se plantea; se caracterizó el marco de trabajo Dracones.

---

## Capítulo 2: Soporte tecnológico

### 1.1 INTRODUCCIÓN

En este capítulo se pretenden valorar las tecnologías disponibles para el desarrollo de Sistemas de información geográficas con Dracones. Además de describir las tecnologías a emplear para la implementación del sistema. Incluyéndose la descripción de las metodologías de desarrollo de software existentes, llegando a la selección de una de ellas para guiar el proceso de desarrollo de software en la actual investigación.

### 1.2 VALORACIÓN DE LAS TECNOLOGÍAS DISPONIBLES PARA EL DESARROLLO DE SIG CON DRACONES. DESCRIPCIÓN DE CADA UNA DE ELLAS

Para lograr construir un software con calidad, es de vital importancia planificar bien el tiempo para su desarrollo, además de trabajar de forma organizada, donde se controle y documente todo lo relacionado con el sistema en construcción. También se hace necesario la selección de una metodología adaptable a las características propias del software que se esté desarrollando para de este modo eliminar los riesgos que puedan presentarse durante el desarrollo del proyecto.

Precisamente estos elementos demuestran la necesidad de realizar una valoración y descripción de las tecnologías disponibles para el desarrollo de SIG con Dracones, así como de las metodologías de desarrollo de software, a fin de lograr emplear la que mejor se ajuste al desarrollo del sistema en cuestión.

Partiendo de que el marco de trabajo Dracones permite desarrollo tanto en Python como en PHP. A continuación se describen cada uno de estos lenguajes por separado a fin de seleccionar uno de ellos para la implementación del sistema.

#### 1.2.1 DESCRIPCIÓN DE PYTHON

Python es un lenguaje de programación interpretado, dinámico y orientado a objetos, que puede ser usado de muchas maneras en el desarrollo de software. Ofrece gran soporte e integración con otros lenguajes y herramientas, viene con una extensiva cantidad de librerías y puede ser aprendido en pocos días. Python se puede ejecutar en Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, y teléfonos celulares Nokia. También ha sido portado para las máquinas virtuales de Java y .NET.

---

Python es distribuido bajo la licencia Open Source OSI que lo hace libre para ser usado inclusive en el desarrollo de productos comerciales. (Enríquez)

Este lenguaje tiene varias razones por las que se podría usar, entre ellas se encuentran su portabilidad, versatilidad, simplicidad, interactividad, sintaxis clara y legible, productividad, popularidad, facilidad y rapidez de aprendizaje.

### 1.2.2 DESCRIPCIÓN DE PHP

PHP (Hypertext Preprocessor) por su parte, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El mismo inició como una modificación a Perl, escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su curriculum en su web.

Con PHP se puede hacer cualquier cosa que se desee realizar con un script CGI (Common Gateway Interface), como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Es conocido como una tecnología de código abierto que resulta muy útil para diseñar de forma rápida y eficaz aplicaciones Web dirigidas a bases de datos. PHP es un potente lenguaje de secuencia de comandos diseñado específicamente para permitir a los programadores crear aplicaciones Web con distintas prestaciones de forma rápida.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos, como por ejemplo InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. Este lenguaje también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en pdf, hasta analizar código XML. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente.

Entre sus ventajas se encuentran: (Orbes, 2010)

- ✓ La sintaxis es similar a la del C, por esto cualquier programador con experiencia en lenguajes del estilo C podrá entender rápidamente PHP.
- ✓ Es completamente expandible. Está compuesto de un sistema principal (escrito en Zend), un conjunto de módulos y una variedad de extensiones de código.
- ✓ Actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen yTHTTPD.
- ✓ Es rápido, ya que generalmente es utilizado como módulo de apache y al estar escrito en C se ejecuta rápidamente utilizando poca memoria.

- 
- ✓ Tiene manejo de excepciones (desde PHP5).
  - ✓ No requiere definición de tipos de variables, aunque las mismas se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Teniendo en cuenta la rapidez y utilización de poca memoria que emplea PHP, así como las demás ventajas antes mencionadas; además de considerar que el grupo de desarrollo se encuentra mejor documentado y capacitado en dicho lenguaje. Se decide emplear PHP como lenguaje de programación para el desarrollo del sistema de información geográfica.

Como parte de las demás tecnologías para el desarrollo de SIG con Dracones se utilizarán: Netbeans como IDE de desarrollo y Visual Paradigm como herramienta CASE.

### 1.2.3 DESCRIPCIÓN DE NETBEANS

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio del 2000 y continúa siendo el patrocinador principal de los proyectos. Hoy día hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo, o sea una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java. Pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Este es un producto libre y gratuito sin restricciones de uso.

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, y especializadas en desarrollo de software, que proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

---

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.

Netbeans proporciona las herramientas necesarias para crear aplicaciones profesionales para entornos de escritorio, empresa, web y móviles, ya sea en C/C++, Java, PHP, Ruby, Python, JavaScript...Ha sido desarrollado para distintas plataformas como Linux, MacOS X, Solaris y también Windows. Posee un editor de código robusto con ayuda en línea y reconocimiento de sintaxis. Se integra con Sistemas de Control de Versiones, así como con sistemas de gestión de Base de Datos como Mysql.

#### 1.2.4 DESCRIPCIÓN DE VISUAL PARADIGM

Visual Paradigm para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda, aún más a la rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Entre sus características se encuentran:

- Soporte de UML versión 2.1.
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- Ingeniería de ida y vuelta. Ingeniería inversa.
- Generación de código, modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso, entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas, Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio, Dibujo de diagramas UML con plantillas (esténcils) de MS Visio.

- 
- Editor de figuras.

### 1.3 DESCRIPCIÓN DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE EXISTENTES

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Las mismas definen un conjunto de procesos y actividades que guían a los desarrolladores de software durante el ciclo de vida de un proyecto. Especifican además los artefactos a construir y organizan el personal involucrado en roles dentro del equipo de trabajo. Existen dos grandes grupos de metodologías: las ágiles y las robustas o tradicionales.

Piattini define a la metodología de desarrollo de software como “un conjunto de procedimientos, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”. (Piatini, 2007)

También se definen como un conjunto de métodos que se utilizan para desarrollar una actividad con el objetivo de formalizarla y optimizarla. (Martínez, y otros, 2005)

#### 1.3.1 METODOLOGÍAS TRADICIONALES O PESADAS

Las metodologías tradicionales o pesadas son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo, en la cual se establecen estrictamente las actividades involucradas, los roles definidos, los artefactos que se deben producir, las herramientas y notaciones que serán utilizadas, así como el modelado y documentación detallada.

Este esquema tradicional ha demostrado ser efectivo y necesario en proyectos de gran tamaño (según el tiempo y recursos), sin embargo este enfoque no resulta ser el más aceptado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del buen hacer de la ingeniería del software, asumiendo el riesgo que ello conlleva. (Canós, y otros, 2008)

Entre las características de este tipo de metodologías se encuentran:

- 
- Enfocadas al proceso definido por la metodología y no a las características del proyecto.
  - Establecen fases rígidas.
  - Cada fase tiene una serie de documentos sumamente explícitos.
  - Filosofía basada en predicción del proyecto.

RUP (Rational Unificad Proceses) forma parte de estas metodologías pesadas, y es precisamente la más usada actualmente para el desarrollo de software dentro de ellas.

#### 1.3.1.1 DESCRIPCIÓN DE RUP

El Proceso Unificado de Rational (RUP), es un proceso de ingeniería de software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido. Cubre el ciclo de vida y desarrollo de software. (Díaz-Antón, y otros, 2004)

RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

1. Desarrollo de software en forma iterativa.
2. Manejo de requerimientos.
3. Utiliza arquitectura basada en componentes.
4. Modela el software visualmente (modela con UML).
5. Verifica la calidad del software.
6. Controla los cambios.

El Proceso Unificado es un proceso de desarrollo de software cuyo ciclo de vida se caracteriza por:

- Dirigido por casos de Uso.
- Centrado en arquitectura.
- Iterativo e incremental.

RUP es uno de los de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. Un proyecto realizado siguiendo RUP se divide en cuatro fases:

1. Conceptualización (puesta en marcha).

- 
2. Elaboración (definición, análisis, diseño).
  3. Construcción (implementación).
  4. Transición (fin del proyecto y puesta en producción).

Cada fase se caracteriza por tener una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas se seguirá un modelo de cascada para los flujos de trabajo que requieren los flujos de trabajo anteriormente citados. RUP define nueve flujos de trabajo a tener en cuenta en cada fase del proyecto:

- 1) Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- 2) Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- 3) Análisis y Diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- 4) Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- 5) Test o Pruebas: Busca los defectos a lo largo del ciclo de vida.
- 6) Instalación.
- 7) Administración de Configuración y Cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, entre otros.
- 8) Administración del Proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- 9) Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado (artefactos en la jerga de RUP) que se espera de ellos.

### 1.3.2 METODOLOGÍAS ÁGILES

---

Estas metodologías están orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollos pequeños, hacen especial atención en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

Este tipo de metodología se caracteriza por:

- La resistencia a las metodologías burocráticas.
- Hicieron un postulado en lugar de una metodología:
  - Individuos y sus interacciones son más importantes que procesos y herramientas.
  - Software que funcione es más importante que documentación exhaustiva.
  - Colaboración con el cliente en lugar de negociación de contratos.
  - Respuesta ante el cambio en vez de seguir un plan cerrado. (Martínez, y otros, 2005)

Dentro de las metodologías ágiles se encuentran:

- XP (Extreme Programming).
- FDD (Feature Driven Development).
- ASD (Adaptive Software Development), entre otras.

A continuación se describen dos de ellas.

#### 1.3.2.1 DESCRIPCIÓN DE XP

Esta metodología es una nueva disciplina para el desarrollo de software, que ha irrumpido recientemente con gran impacto en la variedad de métodos, técnicas y metodologías existentes. Se trata de un método ligero, en contraposición a los métodos pesados como métrica (Acebal, y otros, 2002).

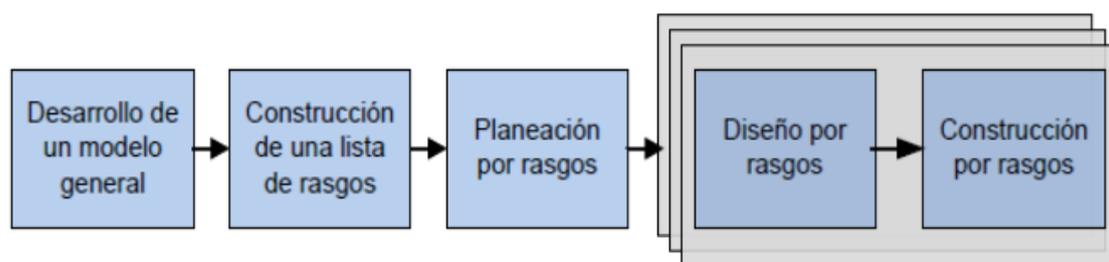
Esta metodología fue desarrollada por Kent Beck. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Esta guiada por una rápida programación y se basa en realimentación continua entre el cliente y el equipo de desarrollo, o sea comunicación fluida entre todos los involucrados en la reutilización del código. Impone un alto nivel de disciplina entre programadores, lo cual permite

mantener un mínimo nivel de documentación que a su vez se traduce en una velocidad de desarrollo. XP se define especialmente para proyecto de corto plazo con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Está formada por cuatro partes fundamentales que encierran sus características principales, como historia de usuarios, roles, procesos y prácticas. Las técnicas de esta metodología se dividen en cuatro ámbitos, que son: planificación, diseño, codificación y pruebas. Es importante aclarar que estas técnicas son aplicables a proyectos que cuentan con un equipo de desarrollo medio grande, encaminadas a solucionar problemas no triviales, por lo que no tiene sentido ser aplicadas en proyectos pequeños.

### 1.3.2.2 DESCRIPCIÓN DE FDD

Esta metodología fue diseñada por Peter Coad, Eric Lefebvre y Jeff DeLuca. Es un proceso ágil para el desarrollo de sistemas. No hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. Se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto. Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado. Propone tener etapas de cierre cada dos semanas. (Molpeceres, 2002)

En dicha metodología se obtienen resultados periódicos y tangibles. Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear. Define claramente entregas tangibles y formas de evaluación del progreso del proyecto. El proceso consiste de cinco pasos secuenciales durante los cuales se diseña y se construye el sistema:



**Figura 1: Fases de desarrollo de la metodología FDD**

 **Desarrollo de un modelo general**

---

La primera fase que plantea esta metodología es el desarrollo de un modelo global, al iniciar esta fase, se tiene una idea del contexto y visión del sistema. Se presenta un ensayo del dominio (walkthrough) en el cual los miembros del equipo son informados a través de una descripción del sistema que se quiere construir. El dominio global es dividido en diferentes áreas y se realiza un ensayo del dominio detallado para cada una de las áreas del dominio. A partir de estas descripciones del contexto y visión del sistema se realizan un modelo de objetos para cada área de dominio y simultáneamente, se construye un modelo global del sistema a partir de los modelos por áreas. (Calabria, 2003)

#### **Construcción de una lista de rasgos.**

Una vez concebido el modelo global se procede a identificar los rasgos o requisitos que resumen las características y comportamiento del sistema que se desea construir. El resultado de esta fase es una lista de rasgos categorizada jerárquicamente. La lista de rasgos se compone por áreas temáticas, actividades del negocio que comprenden estas áreas temáticas y por los rasgos que representan los pasos para dar cumplimiento a cada actividad del negocio. Estos rasgos son pequeñas funcionalidades útiles a los ojos del cliente y los rasgos que requieran de más de diez días se descomponen en otros más pequeños que se puedan cumplir en el tiempo máximo de dos semanas. (Calabria, 2003)

#### **Planeación por rasgos.**

Teniendo en mano los rasgos por los que se guiará el desarrollo, se procede a incluir la creación de un plan de alto nivel, en el que los conjuntos de rasgos se ponen en secuencia conforme a su prioridad y dependencia. Esta planificación permite establecer que rasgos se incluyen en cada iteración de los dos últimos procesos que establece esta metodología. (Calabria, 2003)

#### **Diseño por rasgos y construcción por rasgos.**

Este proceso se realiza de forma iterativa. En cada iteración se selecciona un pequeño conjunto de rasgos. Se identifican las clases involucradas por cada rasgo y se diseñan e implementan estas clases a partir del conjunto de rasgos que están en la iteración. Se procede luego iterativamente hasta que se producen todos los rasgos identificados. Una iteración puede tomar de unos pocos días a un máximo de dos semanas. El proceso iterativo está definido por los hitos: un ensayo del

---

dominio del rasgo a desarrollar, diseño de las clases, inspección del diseño, codificación, pruebas unitarias, inspección de código y por último promoción del rasgo que se construyó. (Calabria, 2003)

#### 1.4 SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE A UTILIZAR

Debido a que todos los procesos se centran en la producción de software, es necesario establecer una comparación, no en su conjunto, sino según los medios que emplean y sus resultados de al menos tres metodologías para lograr adaptar una de ellas al proyecto en cuestión. En este caso se comparará FDD, RUP y XP. A continuación se presentan los diferentes aspectos a comparar:

- Tamaño de los equipos: RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración. FDD y XP se implementan mejor para proyectos cortos y equipos más pequeños, siendo quizás FDD más escalable que XP.
- Obtención de requisitos: RUP y XP crean como base UseCases y UserStories, por lo contrario FDD no define explícitamente esa parte del proyecto sobre la adquisición de requisitos.
- Evaluación del estado del proyecto: FDD es posiblemente el proceso más adecuado para definir métricas que definan el estado del proyecto, puesto que al dividirlos en unidades pequeñas es bastante sencillo hacer un seguimiento de las mismas. XP también define esos componentes pequeños. RUP por su parte, es tan grande y complejo en este sentido como en el resto, por lo que manejar el volumen de información que puede generar requiere mucho tiempo.
- Carga de trabajo: XP es un proceso ligero, esto es, que los creadores del proceso han tenido cuidado de no poner demasiadas tareas organizativas sobre los desarrolladores. RUP es un proceso pesado, basado mucho en la documentación, en la que son deseables todos esos cambios volátiles. FDD es por su parte un proceso intermedio, en el sentido de que genera más documentación que XP pero menos que RUP.
- Relación con el cliente: Con RUP se presentarán al cliente los artefactos del final de una fase, en contrapartida, la aseguración de la calidad en XP y FDD no se basa en formalismos en la documentación, sino en controles propios y una comunicación fluida con el cliente.

- 
- Conocimientos sobre la arquitectura: En RUP se intentará reducir la complejidad del software a producir a través de una planificación intensiva. En XP se conseguirá a través de la programación a pares que ya en la creación del código se puedan evitar errores y malos diseños. En FDD sin embargo se usan las sesiones de trabajo conjuntas en fase de diseño para conseguir una arquitectura sencilla y sin errores. (Molpeceres, 2002)

En conclusión se puede comprobar que FDD es una metodología de desarrollo ágil, que disminuye el riesgo de los proyectos, pues gracias a sus entregas tangibles y al constante monitoreo de su calidad, se asegura el firme avance del mismo. Esta metodología ayuda a los equipos a conseguir resultados periódicos y tangibles. Utiliza pequeños bloques llamados features, los cuales contienen la funcionalidad del sistema. Hace énfasis en la obtención de resultados cada dos semanas.

Dicha metodología asegura la satisfacción de los desarrolladores, gerentes y clientes sin afectar al proyecto. Esto gracias a un buen manejo de las actividades, a la disminución del riesgo del proyecto y al aseguramiento de la calidad del mismo, respectivamente. Es un proceso adaptativo ya que permite realizar cambios de último momento debido a nuevos requerimientos y a las necesidades del negocio.

Por todos los elementos planteados con anterioridad se define el empleo de FDD como metodología de desarrollo de software, pues se cuenta con un equipo pequeño. Además con dicha metodología se lograran establecer métricas que definan el estado del proyecto, así como el avance firme del mismo a través de controles propios sin tantos documentos.

## 1.5 CONCLUSIONES

En este capítulo se llevó a cabo la valoración de las tecnologías disponibles para el desarrollo de sistemas de información geográfica con el marco de trabajo Dracones, así como la descripción de cada una de ellas. También se logró la selección de aquellas tecnologías a emplear para la implementación del sistema. Continuamente se describieron de igual manera los dos grupos de metodologías de desarrollo de software existentes, particularizando en cada una de sus variantes, a fin de lograr la selección de una de ellas para utilizarla como guía en el proceso de desarrollo del sistema a tratar.

---

## Capítulo 3: Análisis de la solución propuesta

### 1.1 INTRODUCCIÓN

En el presente capítulo se analizarán los procesos del negocio que se realizarán para la construcción del sistema de información geográfica. Se desarrollará un modelo general y se identificarán los rasgos funcionales y no funcionales de los servicios que consume el proceso. Además se llevará a cabo una planificación por iteraciones, generándose los correspondientes artefactos del análisis y el diseño.

### 1.2 MODELO GLOBAL

En el presente desarrollo se realizará un modelo global o general, ya que los procesos del negocio no se encuentran bien definidos. Dicho modelo persigue como objetivos: Describir y comprender las clases y objetos más significativos dentro del contexto del problema para lograr definir los procesos y roles más significativos en el sistema a desarrollar. Siendo esto un elemento bien importante que les permite a los usuarios, clientes y desarrolladores, manejar términos semejantes o comunes.

Dicho modelo global consiste específicamente, en la construcción de un diagrama de clases que representa los tipos de objetos más significativos dentro del dominio del problema y las relaciones entre ellos. Dicho diagrama de clases es de carácter estructural y se traduce como el tradicional diagrama entidad/relación de las bases de datos relacionales. El mismo presenta dos grandes diferencias ya que puede incluir relaciones de herencia, generalización y especialización, además de que las operaciones no reflejan conveniencias de programación sino que se describen a través de rasgos, especificando cómo debe comportarse el objeto. (Palmer & J.M.F.A, 2002)

### 1.3 GLOSARIO DE TÉRMINOS ASOCIADOS AL DOMINIO

**Mapa:** Es una representación gráfica de cierta porción de territorio que suele aparecer sobre una superficie plana, aunque, en el caso de los globos terráqueos, también puede adquirir una forma esférica. Los mapas están realizados a escala y con propiedades métricas que permiten tomar medidas de distancias, superficies y ángulos con resultados muy precisos.

**Capa:** Las capas son una forma de organizar la información temática para la elaboración de los SIG. Son transparencias colocadas a criterio del autor para facilitar la manipulación de la información.

**Información socioeconómica:** Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país.

**Modelo ráster:** Se centra en propiedades del espacio, más que en la precisión de la localización. En el mismo se divide el espacio en celdas regulares, donde cada una representa un valor. Importante resaltar que mientras más grande son las dimensiones de cada celda, menor es la precisión.

**Modelo vectorial:** Se centra en la precisión de la localización de los elementos sobre el espacio. Para modelar digitalmente las entidades del mundo real se utilizan tres tipos de objetos: puntos, líneas y polígonos. Dicho modelo es más popular en el mercado.

#### 1.4 DIAGRAMA GENERAL

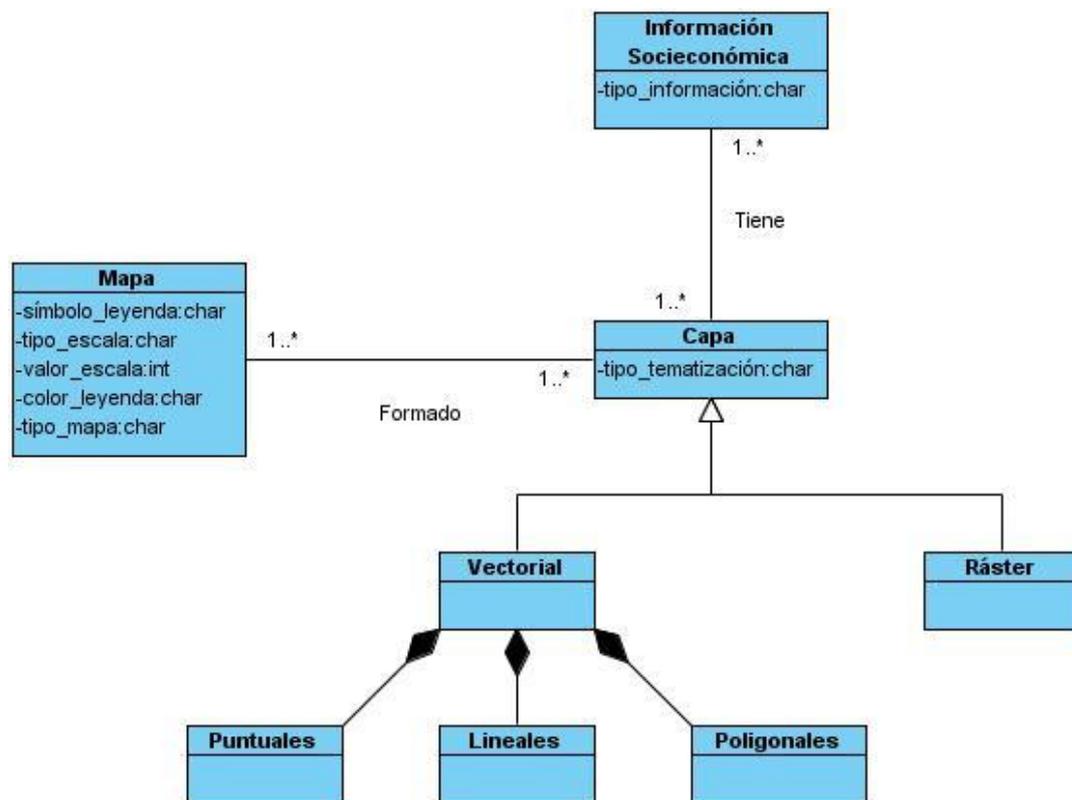


Figura 2: Diagrama general

## 1.5 CONSTRUCCIÓN DE LA LISTA DE RASGOS

A partir del modelo general se pudieron identificar un conjunto de rasgos que representan la base sobre la cual se debe diseñar e implementar el sistema. La lista de rasgos se estructuró de forma jerárquica, donde se definen áreas temáticas, las actividades del negocio comprendidas en cada área temática y los rasgos que permiten cumplimentar la actividad de negocio a la que pertenecen. En la **Tabla 1** se muestran los rasgos identificados con sus correspondientes actividades del negocio dentro del área temática determinada.

Lista de Rasgos		
Área Temática	Actividades del Negocio	Rasgos
Navegación	Mostrar mapa	<b>R1. Mostrar Mapa</b>  1.1 El sistema muestra el mapa al usuario.
	Control de capas	<b>R2. Control de capas</b>  2.1 Se muestra la opción de control de capas, que le permite al usuario seleccionar la capa que desea visualizar.  2.2 El sistema muestra en el mapa la capa seleccionada por el usuario.
	Realizar navegación	<b>R3. Realizar Navegación</b>  3.1 El sistema brinda la posibilidad al usuario de poder realizar diferentes operaciones sobre el mapa, ya sea Acercar, Alejar, Ver todo, Siguiente, Anterior ó Mover el mismo.  3.2 Se visualiza en el mapa la operación ejecutada por el

		usuario.
	Realizar Navegación	<p><b>R3.1 Acercar el mapa</b></p> <p>3.1.1 El sistema brinda la posibilidad de acercar el mapa.</p> <p>3.1.2 El sistema muestra la imagen acercándola y disminuye la escala puntualmente, es decir, coge las coordenadas (x, y) y mueve al centro del mapa el punto donde dio clic disminuyendo al doble la escala.</p>
	Realizar Navegación	<p><b>R3.2 Alejar el mapa</b></p> <p>3.2.1 El sistema brinda la posibilidad de alejar el mapa.</p> <p>3.2.2 El sistema muestra la imagen alejándola y aumenta la escala puntualmente, es decir, coge las coordenadas (x, y) y mueve al centro del mapa el punto donde dio clic aumentando al doble la escala.</p>
	Realizar Navegación	<p><b>R3.3 Ver todo</b></p> <p>3.3.1 Se muestra la opción Ver todo.</p> <p>3.3.2 El sistema muestra la imagen, en este caso el mapa con el tamaño definido en el extend inicial.</p>
	Realizar Navegación	<p><b>R3.4 Historial de navegación</b></p> <p>3.4.1 Se muestran las opciones</p>

		<p>Siguiente y Anterior.</p> <p>3.4.2 El sistema guarda un historial de navegación con las últimas cinco operaciones sobre el mapa. Con la opción anterior retrocede a la posición anterior en que se encontraba el mapa; y con la opción siguiente adelanta una posición.</p>
	Realizar Navegación	<p><b>R3.5 Mover mapa</b></p> <p>3.5.1 El sistema brinda la posibilidad de mover el mapa.</p> <p>3.5.2 El sistema muestra el mapa hacia donde lo traslada el usuario.</p>
Identificación	Identificar Objeto	<p><b>R4. Identificar objeto</b></p> <p>4.1 El sistema permite al usuario poder identificar objetos en el mapa.</p> <p>4.2 El sistema muestra los objetos identificados sobre el mapa.</p>
Análisis	Crear mapa temático por colores	<p><b>R5. Crear mapa temático por colores</b></p> <p>5.1 Se muestra la opción de tematizar por colores.</p> <p>5.2 El sistema actualiza el mapa con la tematización por colores según los valores introducidos por el usuario.</p>

	<p>Crear mapa temático por gráficas</p>	<p><b>R6. Crear mapa temático por gráficas</b></p> <p>6.1 Se muestra la opción de tematizar por gráficas.</p> <p>6.2 El sistema actualiza el mapa con la tematización por gráficas según los valores introducidos por el usuario.</p>
--	---	---

**Tabla 1: Listado de rasgos**

1.6 PLANEACIÓN POR RASGOS

Antes de comenzar a diseñar e implementar el sistema se necesitó planear la lista de rasgos identificada. Esta planeación se realizó teniendo en cuenta todos los rasgos identificados en el proceso. A la hora de planear cada rasgo se tuvo en cuenta que dentro de una iteración no pueden existir rasgos que dependan de otros rasgos que no hayan sido implementados, con la excepción de que puede depender de rasgos no implementados, pero que todos estos se encuentren en la misma iteración. A continuación se muestran en forma de tablas la planeación de cada rasgo.

**Área temática: Navegación**

Realizar Navegación					Rasgos: 3
Rasgos	Modelo general	Diseño	Implementación	Pruebas	Despliegue
	Plan	Plan	Plan	Plan	Plan
Mostrar mapa	10/03/11	11/03/11	15/03/11	22/03/11	25/03/11
Realizar navegación	10/03/11	11/03/11	26/03/11	1/04/11	4/04/11
Control de	10/03/11	11/03/11	5/04/11	11/04/11	14/04/11

capas					
-------	--	--	--	--	--

**Tabla 2: Planeación de los rasgos para la navegación**

Área temática: Identificación

Análisis					Rasgos: 1
Rasgos	Modelo general	Diseño	Implementación	Pruebas	Despliegue
	Plan	Plan	Plan	Plan	Plan
Identificar objeto	10/03/11	11/03/11	15/04/11	21/04/11	24/04/11

**Tabla 3: Planeación de los rasgos para la identificación**

Área temática: Análisis

Análisis					Rasgos: 2
Rasgos	Modelo general	Diseño	Implementación	Pruebas	Despliegue
	Plan	Plan	Plan	Plan	Plan
Crear mapa temático por colores	10/03/11	11/03/11	25/04/11	1/05/11	4/05/11
Crear mapa temático por gráfica	10/03/11	11/03/11	5/05/11	11/05/11	15/05/11

**Tabla 4: Planeación de los rasgos para análisis**

#### 1.7 RASGOS NO FUNCIONALES

---

Los rasgos no funcionales son exigencias de cualidades que el producto debe tener. Un rasgo no funcional es una característica ya sea del sistema, del proyecto o del servicio de soporte, que se requiere junto con la especificación del sistema pero, no se satisface añadiendo código, sino cumpliendo con esta como si de una restricción se tratase.

Los rasgos no funcionales son de igual manera que los funcionales de gran importancia, debido que si alguno de ellos falla el sistema pudiera no tener utilidad alguna. (Tecnología y Synergix. Visión de los Sistemas de Información y la Ingeniería del Software)

### **Usabilidad**

- ✚ El sistema debe ser de fácil uso para los usuarios con escasos conocimientos respecto al tema.
- ✚ Todos los miembros del equipo de desarrollo tendrán la posibilidad de observar y modificar cualquier aspecto deseado del sistema.
- ✚ Toda la documentación generada durante el desarrollo del sistema estará escrita clara y explícitamente para futuras actualizaciones del sistema.

### **Disponibilidad**

- ✚ El sistema debe estar 100% disponible en todas las horas del día.

### **Eficiencia**

- ✚ El tiempo de respuesta del sistema será el menor posible, tratando siempre de lograr la eficiencia y exactitud en los resultados de las peticiones de los usuarios.
- ✚ Ante un fallo del sistema el tiempo de recuperación será en la mayor brevedad posible.

### **Portabilidad**

- ✚ El sistema podrá ser ejecutado tanto en Windows como en Linux.

### **Soporte**

- ✚ El sistema recibirá mantenimiento en el período de tiempo determinado por el

---

equipo de desarrollo y los clientes.

- ✚ El equipo de desarrollo será el encargado de configurar e instalar las tecnologías requeridas en las PCs para la utilización del sistema.

### **Restricciones en el Diseño y la Implementación**

- ✚ El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.
- ✚ Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación) y patrones de diseño.

### **Interfaz de usuario**

- ✚ El sistema debe contar con una interfaz amigable, atractiva y fácil de entender.
- ✚ Los colores deben ser de tonos fuertes y contrastantes para que el usuario logre observar claramente lo que solicitó del sistema.
- ✚ Los íconos correspondientes a las funcionalidades deben ser comprensibles, de tal manera que especifiquen la funcionalidad que encierra cada uno de estos elementos
- ✚ Debe predominar el uso de los íconos para la interacción con el usuario.

### **Software**

#### **Para las PCs clientes:**

- ✚ Un Navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares W3C (World Wide Web Consortium)
- ✚ Sistema operativo: GNU/Linux o Windows.

#### **Para los servidores:**

- ✚ Sistemas operativos GNU/Linux o Windows Server 2008 o superior.
- ✚ Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- ✚ PostgreSQL como Sistema Gestor de Base de Datos.
- ✚ PostGis como extensión de PostgreSQL como soporte de datos espaciales.
- ✚ MapServer 5.6 o superior, con extensión PHP mapscript.

---

## Hardware

### Para las PCs clientes:

- ✚ Se requiere tengan tarjeta de red.
- ✚ Al menos 256 MB de memoria RAM.
- ✚ Se requiere al menos 40 GB de disco duro.
- ✚ Procesador 512 MHz como mínimo.

### Para los servidores:

- ✚ Se requiere tarjeta de red.
- ✚ El Servidor de Mapas tenga como mínimo 2GB de RAM y 40GB de disco duro.
- ✚ El Servidor de BD tenga como mínimo 2GB de RAM y 80GB de disco duro.
- ✚ Procesador 3 GHz como mínimo.

## Requisitos Legales, de Derecho de Autor y otros

- ✚ El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

## 1.8 ARQUITECTURA QUE UTILIZARÁ EL SISTEMA

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, la confiabilidad, escalabilidad, portabilidad entre otros. (Kruchten & Philippe, 1995)

La misma establece las bases y fundamentos para que los desarrolladores trabajen en el sistema y alcancen en mayor medida los objetivos trazados. La selección de una arquitectura se realiza según las características propias del sistema a implementar y según las metas que se pretendan lograr con el avance del mismo.

## 1.9 DESCRIPCIÓN DE LA ARQUITECTURA A EMPLEAR PARA LA IMPLEMENTACIÓN DEL SISTEMA

---

El marco de trabajo Dracones, sobre el cual se estará desarrollando la aplicación, se define principalmente como una comunicación e interacción entre clases y objetos del sistema. El mismo propone una arquitectura Orientada a Objetos, la cual se centra fundamentalmente en la herencia y el encapsulamiento.

Dicha arquitectura se define como una estructura de clases y paquetes que permiten mantener una aplicación de software flexible, robusto y desarrollable. Los componentes del sistema encapsulan datos y operaciones que deben utilizarse para manipular dichos datos. La comunicación y coordinación entre componentes se realiza mediante envío de mensajes. En esencia es un sistema donde se enfatiza el empaquetamiento entre datos y operaciones que permiten manipular y acceder a dichos datos. (Martín R. C., 2009)

#### 1.10 PATRONES DE DISEÑO

Los patrones de diseño se definen como soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. Es evidente que a lo largo de multitud de diseños de aplicaciones hay problemas que se repiten o que son análogos, es decir, que responden a un cierto patrón. Sería deseable tener una colección de dichos patrones con las soluciones óptimas para cada caso. Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. Los mismos han revolucionado el diseño orientado a objetos. (García, 2005) A continuación se especifican los patrones de diseño empleados en nuestro sistema.

##### 1.10.1 SOLITARIO

En el diseño de clases es necesario aplicar la solución del patrón Solitario que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. El objetivo del mismo es crear el objeto mapa, para que no se cree cada vez que se hace un cambio en la aplicación.

##### 1.10.2 ACCIÓN

Se utiliza en el proceso de petición mediante la Interfaz Gráfica de Usuario (GUI) al sistema, de una información cualquiera por un cliente.

#### 1.11 PATRONES GRASP

---

Los patrones generales de software para asignación de responsabilidades, también constituyen una serie de buenas prácticas de aplicación recomendable en el diseño de software. Dichos patrones se dividen en 6, los mismos son: experto, creador, controlador, fachada, alta cohesión y bajo acoplamiento. En el caso particular de la presente investigación se emplean dos de ellos. A continuación se describen los mismos.

#### 1.11.1 BAJO ACOPLAMIENTO

El bajo acoplamiento es un principio que se debe tener siempre en cuenta durante las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar.

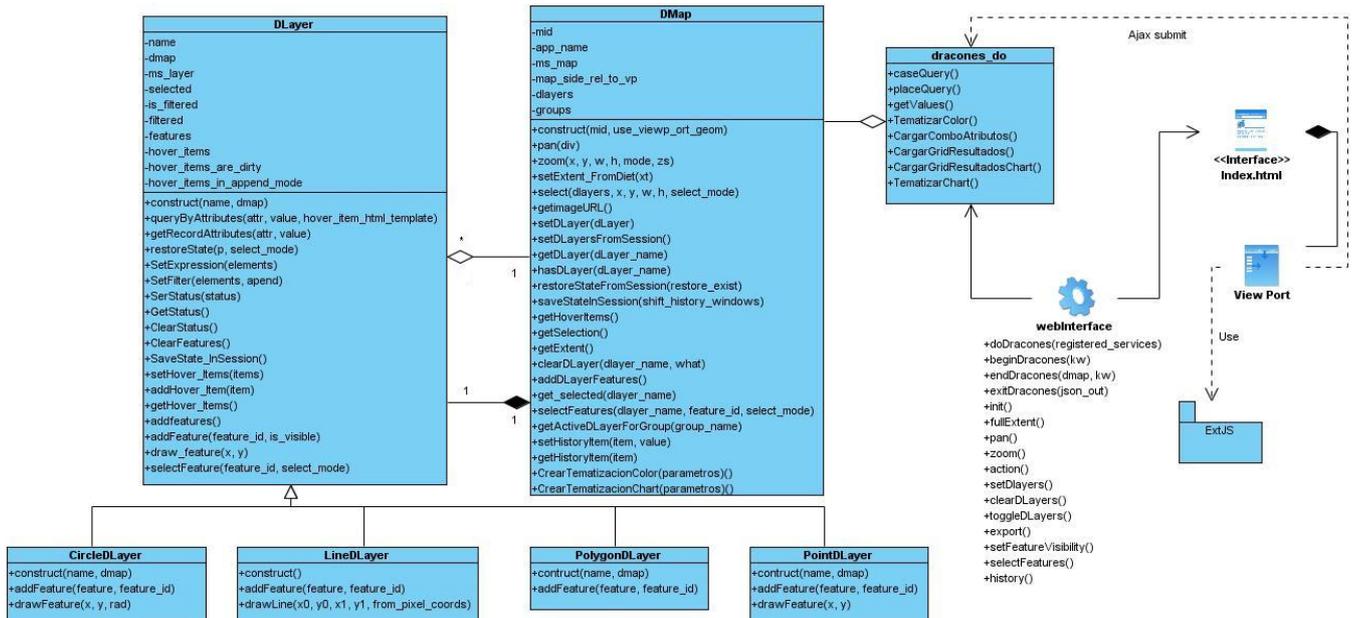
#### 1.11.2 ALTA COHESIÓN

El patrón alta cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. Con el uso de este patrón mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifican el mantenimiento y las mejoras en funcionalidad. A menudo se genera un bajo acoplamiento. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

#### 1.12 DIAGRAMA DE CLASES DEL DISEÑO DE LOS COMPONENTES

Un diagrama de clases del diseño muestra un conjunto de interfaces, colaboraciones y sus relaciones. Los mismos se utilizan para modelar

fundamentalmente la vista de diseño estática de un sistema. En esta actividad se incluye modelar el vocabulario del sistema y las colaboraciones o esquemas. El hecho de que los diagramas de clases permiten visualizar, especificar y documentar modelos estructurales, así como la construcción de sistemas ejecutables aplicando tanto ingeniería directa como inversa, es precisamente lo que hace de dichos diagramas un elemento importante. A continuación se muestra el diagrama de clases del diseño del sistema.



**Figura 3: Diagrama de clases del diseño**

### 1.13 DESCRIPCIÓN DE LAS CLASES DEL DISEÑO

En esta sección se describen las clases del diseño modificadas, o sea en las cuales se produjo algún aporte. Mostrándose para cada una de ellas su nombre, tipo, así como una breve explicación de su responsabilidad o función. Además se especifica el nombre y tipo de cada uno de sus atributos. Por otra parte se expone el objetivo y forma de funcionamiento de las operaciones fundamentales contenidas en la clase.

#### Draconesph\_do

Esta clase presenta un grupo de funcionalidades para darle respuesta a las necesidades de reportes del cliente de la aplicación. Las funcionalidades mencionadas se describen a continuación.

**Nombre:** Draconesph\_do

<b>Atributo</b>	<b>Tipo</b>
No aplicable	No aplicable
<b>Para cada responsabilidad:</b>	
Nombre:	caseQuery()
Descripción:	Función que crea la capa con los datos retornados según la selección del usuario, específicamente con datos de los casos de enfermedades.
Nombre:	placeQuery()
Descripción:	Función que crea la capa con los datos retornados según la selección del usuario, específicamente con los datos de los lugares (Colegios, Tiendas de comidas y Restaurantes-Bares).
Nombre:	getValues()
Descripción:	Función que retorna los valores de los elementos seleccionados por el usuario.
Nombre:	tematizarColor()
Descripción:	Función para recepcionar los datos provenientes de la interfaz, se construye el arreglo de parámetros que posteriormente son pasados al método crear tematización por colores de la clase Dmap.
Nombre:	cargarComboAtributos()
Descripción:	Función para buscar los datos correspondientes a la capa seleccionada y enviarlos a la interfaz para ser mostrados.
Nombre:	cargarGridResultados()
Descripción:	Función para obtener los datos resultantes que se van a tematizar por colores.
Nombre:	cargarGridResultadosChart()
Descripción:	Función para obtener los datos resultantes que se van a tematizar por gráficas.
Nombre:	tematizarChart()
Descripción:	Función para recepcionar los datos provenientes de la interfaz,

	se construye el arreglo de parámetros que posteriormente son pasados al método crear tematización por chart de la clase Dmap.
--	---

**Tabla 5: Descripción de la clase Draconesph\_do**

### Fichero\_Core

Dentro del fichero Core se encuentran las clases Dmap, Dlayer, CircleDlayer, PointDlayer, LineDlayer y PolygonDlayer. El fichero como tal presenta una serie de métodos auxiliares que se encuentran descritos en la documentación del framework. Por lo que a continuación solo se describen los que han sido incluidos.

<b>Nombre:</b> Fichero_Core (métodos auxiliares)	
<b>Atributo</b>	<b>Tipo</b>
No aplicable	No aplicable
<b>Para cada responsabilidad:</b>	
Nombre:	convertHexToRGB( array Color_Hex)
Descripción:	Función para convertir los valores hexadecimales de los colores que provienen de la interfaz en RGB, para que MapServer los pueda comprender.
Nombre:	generarColores( cantidad_elementos)
Descripción:	Función para generar colores de forma aleatoria, cuya cantidad debe ser igual a la especificada por parámetros.
Nombre:	parserQuery(consulta)
Descripción:	Función creada para picar la consulta SQL escrita en el campo data de la capa en el .map.
Nombre:	eliminarEspaciosDobles(cadena)
Descripción:	Función para eliminar los espacios dobles en una cadena.
Nombre:	obtenerCamposConsulta(consulta)
Descripción:	Función para obtener los campos de una consulta.

**Tabla 6: Descripción del fichero Core**

## Dmap

Dmap es una de las clases propias del framework y es la que controla el manejo con el mapa y sus capas, entre otras funcionalidades. En la misma se incluyeron dos operaciones relacionadas con la tematización, las cuales se describen a continuación.

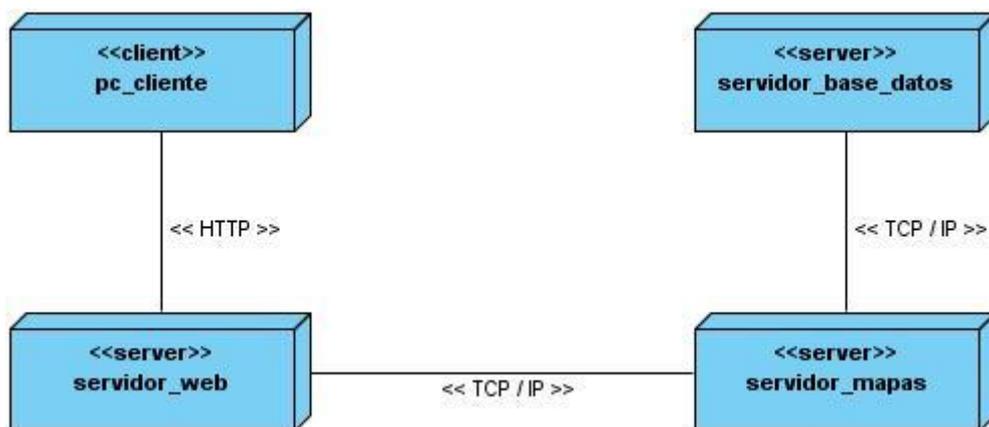
Nombre: Dmap	
Atributo	Tipo
No aplicable	No aplicable
<b>Para cada responsabilidad:</b>	
Nombre:	crearTematizacionColor(parámetros)
Descripción:	Función para tematizar por colores.
Nombre:	crearTematizacionChart(parámetros)
Descripción:	Función para tematizar por gráficas.

### Tabla7: Descripción de la clase Dmap

Las clases CircleDlayer, PointDlayer, LineDlayer y PolygonDlayer heredan de la clase Dlayer. Como bien se mencionó anteriormente, las mismas son propias del framework Dracones y se encuentran descritas en su propia documentación, por lo que no fue necesario incluirlas en el presente documento.

#### 1.14 MODELO DE DESPLIEGUE

El modelo de despliegue muestra la correlación de los elementos de proceso del sistema en tiempo de ejecución, los enlaces de comunicación entre ellos y posiblemente otros dispositivos, representados a través de nodos físicos permitiendo la distribución de comportamiento entre los nodos que se deben representar. Este modelo posibilita comprender y representar la relación entre la arquitectura de software y la arquitectura de hardware del sistema que en este caso debe disponer de computadoras clientes, así como de un servidor web, el servidor de mapas, así como el servidor de base de datos. A continuación en la **Figura 4**, se muestra el diagrama de despliegue para el Sistema de Información Geográfica sobre Dracones:



**Figura 4: Diagrama de despliegue**

### 1.15 CONCLUSIONES

En el presente capítulo se construyó la solución propuesta. Para ello se realizó el modelo general que supone la metodología seleccionada, el cual permitió la construcción de la lista de rasgos que constituyen la base sobre la cual diseñar e implementar el sistema. Además se reflejaron y explicaron los términos asociados con el dominio del problema. Continuamente se identificaron los rasgos no funcionales de los servicios que consume el proceso, así como la arquitectura y patrones a utilizar. Finalmente se realizó la planificación por iteraciones que también plantea la metodología de desarrollo de software utilizada, generándose posteriormente los correspondientes artefactos del análisis y el diseño.

---

## Capítulo 4: Validación

### 1.1 INTRODUCCIÓN

En el actual capítulo se llevarán a cabo las pruebas necesarias que permiten la validación del sistema desarrollado. Para ello se ejecutarán los casos de pruebas pertinentes. Específicamente las pruebas a emplear son las de caja negra o funcionales. Las mismas más bien se centran en lo que se espera como resultado de la aplicación, y serán de notada importancia para lograr verificar que el software implementado funciona con calidad. También se especificarán algunos detalles del rendimiento del sistema.

### 1.2 PRUEBAS DEL SISTEMA PROPUESTO

Durante el desarrollo del software es inevitable encontrarse con diferentes tipos de fallas en el sistema. Dichos errores pueden manifestarse desde el comienzo del proceso de desarrollo. Es por ello que el hecho de ejecutar las pruebas en el programa puede irse desarrollando desde la fase de inicio, aunque es preciso mencionar que el momento cumbre de su realización se materializa en la etapa de construcción. El hecho es que más allá del momento en que se realicen las pruebas, la importancia radica en realizarlas siempre de la mejor manera.

Las mismas son el complemento en cualquier aplicación a desarrollar, ya que permiten realizar las revisiones finales de las especificaciones del diseño y la codificación, además de verificar el cumplimiento de los rasgos planteados en una primera fase. Todos los elementos planteados con anterioridad colaboran para garantizar la calidad del cualquier producto a desarrollar.

### 1.3 PRUEBAS DEL CAJA NEGRA

Dentro de la variedad de pruebas que se le pueden realizar a un software, se pueden mencionar las pruebas de caja blanca y las pruebas de caja negra. En las primeras mencionadas se cuenta con el código, o sea se conoce el mismo y se tratan de ejecutar cada uno de los elementos del mismo. Por su parte en las pruebas de caja negra se cuenta con la interfaz y se maneja con cada uno de los elementos que la componen para llevar a cabo la validación.

El empleo de pruebas de caja negra fija su atención en el modo de interacción del sistema con el medio exterior, entendiendo que es lo que hace sin enfocar relevancia

al como lo hace. En realidad su pretensión es demostrar o verificar que las funcionalidades del programa son prácticas y reales. O sea el probador se limita a suministrar datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Ellas intentan encontrar fallas tales como: (Mañas, 1994)

- ✚ Funciones incorrectas o ausentes.
- ✚ Errores de interfaz.
- ✚ Errores en estructuras de datos o accesos a BD externas.
- ✚ Errores de rendimiento.
- ✚ Errores de inicialización y de terminación.

A continuación se muestran los casos de pruebas realizados al sistema desarrollado. Donde para un total de nueve rasgos funcionales, se probaron cada uno de ellos y se obtuvo como resultados un número de cinco no conformidades o pruebas insatisfactorias, así como un total de once pruebas operativas. En resumen se logró más del 60% de resultados satisfactorios.

#### 1.4 CASOS DE PRUEBAS

##### 1.4.1 DESCRIPCIÓN GENERAL DEL RASGO TEMATIZAR POR COLORES

El rasgo se inicia cuando el usuario desea realizar una tematización por colores atendiendo a una serie de criterios, y termina cuando este obtiene su tematización.

##### 1.4.1.1 SECCIONES A PROBAR

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Crear Mapa Temático por colores.	EC1.1: Crear un mapa temático por colores.	Se crea un mapa temático por colores a partir de los valores introducidos por el usuario.
	EC 1.2: Tratamiento de errores	En caso de entrada errónea de los valores, el sistema muestra una ventana donde se especifica el error cometido.

**Tabla 8: Secciones a probar en tematizar por colores**

#### 1.4.1.2 DESCRIPCIÓN DE VARIABLES

No	Nombre de campo	Clasificación	Requerido	Descripción
1.	Capas	campo de selección	Sí	Se tiene que seleccionar la capa sobre la cual tematizar.
2.	Atributos	campo de selección	Sí	Se tiene que seleccionar un atributo.
3.	Operación	campo de selección	No	Se selecciona una operación.
4.	Valor	campo de entrada	No	Se introduce un valor para desarrollar la operación.
5.	Clases	campo de entrada	No	Se introduce el valor de la clase definida por el usuario.

**Tabla 9: Descripción de variables en tematizar por colores**

#### 1.4.1.3 MATRIZ DE DATOS

Escenario	V	V	V	V	V	Respuesta del sistema	Resultados de la prueba	Flujo central
	1	2	3	4	5			
1.1 Crear mapa temático por colores.	V	V	V	V	V	Se crea un mapa temático por colores a partir de los valores introducidos por el usuario.	Satisfactoria	1. Se muestra la opción de tematizar por colores. 2. Seleccionar la capa sobre la cual se realizará la tematización. 3. Introducir y

							<p>seleccionar los criterios de análisis.</p> <p>4. Definir las clases si el usuario así lo desea.</p> <p>5. Seleccionar la opción Generar.</p> <p>6. Se visualizan los resultados a tematizar.</p> <p>7. Cambiar de ser deseado por el usuario, el color en la pestaña estilo del editor de resultados.</p> <p>8. Seleccionar la opción Tematizar.</p> <p>9. Se representa en el mapa la tematización por colores.</p>
1.1.2	V	V	V	V	V	<p>En caso que se seleccione la capa montreal_ct y se presione directamente en el botón tematizar.</p> <p>Ocurre un error debido a que alguno de los</p>	<p>Insatisfactoria</p> <p>1. Se muestra la opción de tematizar por colores.</p> <p>2. Seleccionar la capa sobre la cual se realizará la tematización.</p> <p>3. Introducir y seleccionar los criterios de</p>

						<p>parámetros contiene un valor inválido.</p>		<p>análisis.</p> <p>4. Definir las clases si el usuario así lo desea.</p> <p>5. Seleccionar la opción Generar.</p> <p>6. Se visualizan los resultados a tematizar.</p> <p>7. Cambiar de ser deseado por el usuario, el color en la pestaña estilo del editor de resultados.</p> <p>8. Seleccionar la opción Tematizar.</p> <p>9. Se representa en el mapa la tematización por colores.</p>
1.1.3	V	V	V	V	V	<p>En caso de seleccionar la capa montreal_ct y luego el atributo gid, definiendo que el mismo sea mayor que 555, ocurre un error debido a que no existen datos para la</p>	<p>Insatisfactoria</p>	<p>1. Se muestra la opción de tematizar por colores.</p> <p>2. Seleccionar la capa sobre la cual se realizará la tematización.</p> <p>3. Introducir y seleccionar los criterios de análisis.</p> <p>4. Definir las</p>

						solicitud requerida.		<p>clases si el usuario así lo desea.</p> <p>5. Seleccionar la opción Generar.</p> <p>6. Se visualizan los resultados a tematizar.</p> <p>7. Cambiar de ser deseado por el usuario, el color en la pestaña estilo del editor de resultados.</p> <p>8. Seleccionar la opción Tematizar.</p> <p>9. Se representa en el mapa la tematización por colores.</p>
1.1.4	V	V	V	V	V	En caso de seleccionar la capa sobre la cual tematizar, así como los criterios de análisis. Si se introduce alguna letra u carácter extraño en el campo clases; ocurre un error debido a que solo pueden	Insatisfactoria	<p>1. Se muestra la opción de tematizar por colores.</p> <p>2. Seleccionar la capa sobre la cual se realizará la tematización.</p> <p>3. Introducir y seleccionar los criterios de análisis.</p> <p>4. Definir las clases si el usuario así lo</p>

					introducirse datos numéricos.		desea. 5. Seleccionar la opción Generar. 6. Se visualizan los resultados a tematizar. 7. Cambiar de ser deseado por el usuario, el color en la pestaña estilo del editor de resultados. 8. Seleccionar la opción Tematizar. 9. Se representa en el mapa la tematización por colores.
1.2 Tratamiento de errores	V	V	V	V	V	En caso de entrada errónea de los valores, el sistema muestra una ventana donde se especifica el error cometido.	Satisfactoria  1. Se muestra la opción de tematizar por colores.  2. Seleccionar la capa sobre la cual se realizará la tematización. 3. Introducir y seleccionar los criterios de análisis. 4. Definir las clases si el usuario así lo desea. 5. Seleccionar la



No	Nombre de campo	Clasificación	Requerido	Descripción
1.	Capas	campo de selección	Sí	Se tiene que seleccionar la capa sobre la cual tematizar.
2.	Campos	campos de selección	Sí	Se tiene que seleccionar como mínimo dos campos en los checkbox de la opción marcar.
3.	Tipo de gráfica	campo de selección	Sí	Se selecciona un tipo de gráfica (barras ó pastel).

**Tabla 12: Descripción de variables en tematizar por gráficas**

#### 1.4.2.3 MATRIZ DE DATOS

Escenario	V 1	V 2	V 3	Respuesta del sistema	Resultados de la prueba	Flujo central
1.1. Crear Mapa Temático por gráficas.	V	V	V	Se crea un mapa temático por gráficas a partir de los valores introducidos por el usuario.	Satisfactoria	1. Se muestra la opción de tematizar por gráficas. 2. Seleccionar la capa sobre la cual se realizará la tematización. 3. Seleccionar como mínimo dos campos en los checkbox de la pestaña marcar. 4. Seleccionar el

						<p>tipo de gráfica a tematizar.</p> <p>5. Cambiar de ser deseado por el usuario, el color en la pestaña estilo.</p> <p>6. Seleccionar la opción Tematizar.</p> <p>7. Se representa en el mapa la tematización por gráficas.</p>
1.1.2	V	V	V	<p>En caso de presionar la opción tematizar sin previa selección de la capa sobre la cual tematizar, ocurre un error debido a que al menos un parámetro contiene un valor inválido.</p>	Insatisfactoria	<p>1. Se muestra la opción de tematizar por gráficas.</p> <p>2. Seleccionar la capa sobre la cual se realizará la tematización.</p> <p>3. Seleccionar como mínimo dos campos en los checkbox de la pestaña marcar.</p> <p>4. Seleccionar el tipo de gráfica a tematizar.</p> <p>5. Cambiar de ser deseado por el usuario, el color en la pestaña estilo.</p> <p>6. Seleccionar la opción Tematizar.</p> <p>7. Se representa</p>

						en el mapa la tematización por gráficas.
1.1.3	V	V	V	En caso de seleccionar la capa montreal_clsc y marcar solo un campo, al presionar la opción tematizar se produce un error debido a que deben existir como mínimo dos campos chequeados.	Insatisfactoria	<ol style="list-style-type: none"> <li>1. Se muestra la opción de tematizar por gráficas.</li> <li>2. Seleccionar la capa sobre la cual se realizará la tematización.</li> <li>3. Seleccionar como mínimo dos campos en los checkbox de la pestaña marcar.</li> <li>4. Seleccionar el tipo de gráfica a tematizar.</li> <li>5. Cambiar de ser deseado por el usuario, el color en la pestaña estilo.</li> <li>6. Seleccionar la opción Tematizar.</li> <li>7. Se representa en el mapa la tematización por gráficas.</li> </ol>
1.2. Tratamiento de errores	V	V	V	En caso de entrada errónea de los valores el sistema muestra una ventana donde se especifica el	Satisfactoria	<ol style="list-style-type: none"> <li>1. Se muestra la opción de tematizar por gráficas.</li> <li>2. Seleccionar la capa sobre la cual</li> </ol>

				error cometido.		<p>se realizará la tematización.</p> <p>3. Seleccionar como mínimo dos campos en los checkbox de la pestaña marcar.</p> <p>4. Seleccionar el tipo de gráfica a tematizar.</p> <p>5. Cambiar de ser deseado por el usuario, el color en la pestaña estilo.</p> <p>6. Seleccionar la opción Tematizar.</p> <p>7. Se representa en el mapa la tematización por gráficas.</p>
--	--	--	--	-----------------	--	---

**Tabla 13: Matriz de datos en tematizar por gráficas**

### 1.4.3 DESCRIPCIÓN GENERAL DEL RASGO CONTROL DE CAPAS

El rasgo se inicia cuando el usuario desea seleccionar una capa sobre la cual realizar algún tipo de operación, y termina cuando este obtiene actualizada la capa seleccionada sobre el mapa.

#### 1.4.3.1 SECCIONES A PROBAR

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Control de capas.	EC1.1: Seleccionar una capa determinada por el usuario.	Según la selección de la capa realizada por el usuario, el sistema muestra en el mapa la actualización de la misma.

**Tabla 14: Secciones a probar en control de capas**

1.4.3.2 DESCRIPCIÓN DE VARIABLES

No	Nombre de campo	Clasificación	Requerido	Descripción
1.	Capas	campo de selección	Sí	Se tiene que seleccionar la capa sobre la cual tematizar activándose la misma.

**Tabla 15: Descripción de variables en control de capas**

1.4.3.3 MATRIZ DE DATOS

Escenario	V 1	Respuesta del sistema	Resultados de la prueba	Flujo central
1.1. Seleccionar una capa determinada por el usuario.	V	Se activa la capa seleccionada por el usuario y se actualiza sobre el mapa.	Satisfactoria	1. Se muestra la opción de control de capas.  2. Seleccionar la capa sobre la cual el usuario desea operar.  3. Se visualiza la capa seleccionada por el usuario sobre el mapa.

**Tabla 16: Matriz de datos en control de capas**

1.4.4 DESCRIPCIÓN GENERAL DEL RASGO REALIZAR NAVEGACIÓN

El rasgo se inicia cuando el usuario desea acercar, alejar, ver todo, ver siguiente, o anterior así como mover el mapa, y termina cuando este visualiza los resultados de la operación realizada sobre el mapa.

1.4.4.1 SECCIONES A PROBAR

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Realizar navegación.	EC1.1: Acercar el mapa según la captura de las coordenadas del centro del mapa realizada por el usuario a través del desplazamiento hacia arriba ejecutado con el mouse.	Según el desplazamiento hacia arriba a través del mouse, realizado por el usuario sobre el mapa, el mismo se acerca disminuyendo puntualmente la escala.
	EC1.2: Alejar el mapa según la captura de las coordenadas del centro del mapa realizada por el usuario a través del desplazamiento hacia abajo ejecutado con el mouse.	Según el desplazamiento hacia abajo a través del mouse realizado por el usuario sobre el mapa, el mismo se aleja aumentando puntualmente la escala.
	EC1.3: Seleccionar la opción ver todo sobre el mapa.	Según la selección del usuario en el campo Ver todo, el sistema muestra el mapa con el tamaño definido en el extend inicial.
	EC1.4: Historial de navegación.	Según la selección del usuario en los campos Siguiente o Anterior, el sistema muestra la posición anterior en que se encontraba el mapa, si la opción fue Anterior, sino adelanta una posición.
	EC1.5: Mover el mapa según la captura de las coordenadas de la posición del puntero, realizada por el usuario a través del clic con el mouse.	Según el desplazamiento realizado por el usuario a través del clic con el mouse, el sistema muestra el mapa en la posición trasladada.

**Tabla 17: Secciones a probar en realizar navegación**

#### 1.4.4.2 DESCRIPCIÓN DE VARIABLES

No	Nombre de campo	Clasificación	Requerido	Descripción
1.	Coordenadas	Campo de captura.	Sí	Se tienen que capturar las coordenadas del centro del mapa a través del desplazamiento hacia arriba ejecutado con el mouse, para poder visualizar la imagen del mapa acercado.
2.	Coordenadas	Campo de captura.	Sí	Se tienen que capturar las coordenadas del centro del mapa a través del desplazamiento hacia abajo ejecutado con el mouse, para poder visualizar la imagen del mapa alejado.
3.	Ver todo	Campo de selección.	Sí	Se tiene que seleccionar la opción Ver todo para poder visualizar la imagen del mapa, según el tamaño definido en el extend inicial.
4.	Siguiente	Campo de selección.	Sí	Se tiene que seleccionar la opción Siguiente, para poder visualizar

				la imagen del mapa en una posición siguiente.
5.	Anterior	Campo de selección.	Sí	Se tiene que seleccionar la opción Anterior, para poder visualizar la imagen de la posición anterior en que se encontraba el mapa.
6.	Coordenadas	Campo de captura.	Sí	Se tienen que capturar las coordenadas de la posición del puntero a través del clic ejecutado con el mouse, para poder visualizar la imagen del mapa trasladado.

**Tabla 18: Descripción de variables en realizar navegación**

1.4.4.3 MATRIZ DE DATOS

Escenario	V 1	Respuesta del sistema	Resultados de la prueba	Flujo central
1.1. Acercar el mapa según la captura de las coordenadas del centro del mapa realizada por el usuario a través del	V	Se visualiza la imagen del mapa acercado.	Satisfactoria	1. Se capturan las coordenadas del centro del mapa a través del desplazamiento hacia arriba ejecutado con el

desplazamiento hacia arriba ejecutado con el mouse.				mouse. 2. Se visualiza la imagen del mapa acercado.
1.1.2 Alejar el mapa según la captura de las coordenadas del centro del mapa realizada por el usuario a través del desplazamiento hacia abajo ejecutado con el mouse.	V	Se visualiza la imagen del mapa alejado.	Satisfactoria	1. Se capturan las coordenadas del centro del mapa a través del desplazamiento hacia abajo ejecutado con el mouse. 2. Se visualiza la imagen del mapa alejado.
1.1.3 Seleccionar la opción ver todo sobre el mapa.	V	Se visualiza la imagen del mapa con el tamaño definido en el extend inicial.	Satisfactoria	1. Se muestra la opción Ver todo. 2. Seleccionar la opción Ver todo. 3. Se visualiza la imagen del mapa con el tamaño definido en el extend inicial.
1.1.4 Historial de navegación.	V	Se muestra la imagen en una posición siguiente o anterior a la que se encontraba.	Satisfactoria	1. Se muestra la opción Siguiente y la opción anterior. 2. Se selecciona la opción Siguiente o Anterior. 3. Se visualiza la

				imagen del mapa en una posición siguiente a la que se encontraba, si se seleccionó la opción Siguiente, sino muestra la imagen del mapa en una posición anterior a la que se encontraba.
1.1.5 Mover el mapa según la captura de las coordenadas de la posición del puntero, realizada por el usuario a través del clic con el mouse.	V	Se muestra la imagen del mapa trasladado.	Satisfactoria	<ol style="list-style-type: none"> <li>1. Se capturan las coordenadas de la posición del puntero a través del clic ejecutado con el mouse.</li> <li>2. Se visualiza la imagen del mapa trasladado.</li> </ol>

**Tabla 19: Matriz de datos en realizar navegación**

#### 1.4.5 DESCRIPCIÓN GENERAL DEL RASGO IDENTIFICAR OBJETO

El rasgo se inicia cuando el usuario desea identificar un objeto sobre el mapa, y termina cuando este obtiene la visualización del objeto deseado.

##### 1.4.5.1 SECCIONES A PROBAR

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC1: Identificar objeto.	EC1.1: Identificar algún objeto siempre y cuando existan elementos localizados en el mapa.	Luego de existir elementos ubicados sobre el mapa. Según el desplazamiento del usuario a través de los mismos, utilizando el mouse para ello. El sistema

		muestra información por cada objeto sobre el cual el usuario se posiciona.
--	--	--

**Tabla 20: Secciones a probar en control de capas**

1.4.5.2 DESCRIPCIÓN DE VARIABLES

No	Nombre de campo	Clasificación	Requerido	Descripción
1.	Coordenadas	campo captura	Sí	Se tienen que capturar las coordenadas de la posición del puntero, a través del posicionamiento del mismo sobre los elementos ubicados en el mapa; para poder identificar algún objeto.

**Tabla 21: Descripción de variables en control de capas**

1.4.5.3 MATRIZ DE DATOS

Escenario	V	Respuesta del sistema	Resultados de la prueba	Flujo central
1.1. Identificar algún objeto siempre y cuando existan elementos localizados	V	Se muestra información de los elementos ubicados sobre el mapa, o sea se identifican los objetos existentes	Satisfactoria	1. Se capturan las coordenadas de la posición del puntero a través del posicionamiento con el mouse. 2. Se identifican

en el mapa.		sobre el plano.		los objetos sobre el plano.
-------------	--	-----------------	--	-----------------------------

**Tabla 22: Matriz de datos en control de capas**

### 1.5 RENDIMIENTO DEL SISTEMA

Las pruebas que se llevan a cabo desde la perspectiva de determinar lo rápido que un sistema, en ciertas condiciones particulares de trabajo puede realizar una tarea; son denominadas pruebas de rendimiento. Las mismas forman parte de una práctica informática que puede servir para múltiples propósitos. Pueden demostrar que el sistema cumple los criterios de rendimiento. Pueden medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal, entre otras variantes. En el presente sistema dichas pruebas de rendimiento se materializarán a través de la comparación entre dos sistemas a fin de encontrar cuál de ellos funciona mejor.

GeneSIG y el Sistema de información geográfica sobre Dracones son los dos sistemas a comparar. GeneSIG se encuentra instalado en una máquina con microprocesador Dual Core a 2.5 GHz y cuenta con 1 GB de memoria RAM. El sistema sobre Dracones por su parte trabaja sobre una maquina con microprocesador Core Duo a 1.6 GHz, con 1 GB de RAM. Una vez comparados ambos sistemas se comprobó que el sistema sobre Dracones posee mejor rendimiento que el de GeneSIG. Los datos que manifiestan tales resultados de la comparación se muestran seguidamente:

Sistemas	GeneSIG	SIG sobre Dracones
<b>Carga inicial de la aplicación</b>	10.37 segundos	2.59 segundos
<b>Proceso de acercar el mapa</b>	4.79 segundos	1.05 segundos
<b>Proceso de ver todo</b>	1.11 segundos	0.34 segundos
<b>Proceso tematizar por colores</b>	2.58 segundos	1.69 segundos
<b>Proceso tematizar por gráficas</b>	2.15 segundos	1.41 segundos

**Tabla 23: Rendimiento del sistema**

---

## 1.6 CONCLUSIONES

En el presente capítulo se llevaron a cabo las pruebas necesarias para lograr validar el sistema de información geográfica desarrollado. Inicialmente se analizaron algunos de los elementos fundamentales de las llamadas pruebas de caja blanca y pruebas de caja negra. Particularmente las pruebas de caja negra fueron las seleccionadas para lograr comprobar la validez del software implementado. Con las mismas básicamente se centró la atención en demostrar o verificar que las funcionalidades del programa son operativas. Donde para un total de 16 pruebas realizadas, 11 de ellas resultaron conformes a lo esperado, dando paso a obtener más del 60 % de resultados satisfactorios. Además se comprobó a través de la comparación entre dos sistemas informáticos que el SIG sobre Dracones posee el mayor rendimiento entre ellos.

---

## CONCLUSIONES

El sistema de información geográfica que se obtuvo con el presente trabajo de diploma, permite la creación de mapas temáticos dinámicos, que son en esencia el producto final de un SIG. Dicho sistema fue construido utilizando tecnologías libres, por lo que resulta de gran utilidad para ser propuesto al grupo de desarrollo del proyecto Aplicativos SIG de la Facultad #6, perteneciente a la Universidad de las Ciencias informáticas. Con el desarrollo de este sistema se logra cumplir con el objetivo trazado en el inicio de la actual investigación, por lo que se puede concluir:

- ✓ Con el desarrollo del sistema de información geográfica se logra un mejor análisis de la distribución espacial de los fenómenos, ya que permite tematizar por colores y gráficas.
- ✓ Para futuros trabajos con el marco de trabajo Dracones, ya se cuenta con la posibilidad de tematizar.
- ✓ El presente trabajo forma parte de una excelente propuesta documentada que le permitirá a la universidad seguir ampliando su desarrollo en aplicaciones web, específicamente vinculadas al tema de la información geográfica.

---

## RECOMENDACIONES

Con el desarrollo del Sistema de Información geográfica sobre Dracones, se logra contar con un nuevo prototipo de aplicación SIG en la web utilizando tecnologías libres, además de la obtención de la documentación técnica del proceso de desarrollo. Todo ello permite contar con una excelente propuesta para el grupo de desarrollo del proyecto Aplicativos SIG. La presente solución informática resulta de gran utilidad pues logra la tematización de mapas, siendo ello el producto final de un SIG. Pero también es válido aclarar que el marco de trabajo sobre el cual se desarrolló es prácticamente un proyecto incipiente, por lo que se recomienda:

- Continuar estudiando el marco de trabajo Dracones a fin de incluir nuevas funcionalidades para lograr del mismo un entorno de trabajo más robusto.
- Utilizar el framework Dracones para aplicaciones reales no muy complejas, ya que permite un ágil desarrollo.

---

## REFERENCIAS BIBLIOGRÁFICAS

*Diseño de software orientado a objetos*

*Diseño de una metodología ágil de desarrollo de software*2004

*El lenguaje de programación Python*

*El proceso unificado de desarrollo de software*2000

*Extreme Programming (XP): un nuevo método de desarrollo de software*2002

*Guía de usuario*2007

*Impacto de los SIG en la sociedad* 2003

*Información Geográfica, Software Libre e Infraestructura de datos espaciales*2007

*Informe sobre metodología FDD*2003

*Practical Guide to Feature Driven Development*2002

Jarochos2007

KruchtenPhilippe 1995*Architectural Blueprints The View Model os Software Architecture..*

*Mappinginteractivo.com*

*Mapwindow.org*Comparación de clientes web para SIG v.5

*Metodologías ágiles en el desarrollo de software*2008

*Personales.ya.com*Extremadura Montijo (Badajoz)

Piatini 2007 *Análisis y diseño detallado de aplicaciones informáticas de gestión*

*Políticas ambientales*1999

*Procesos de desarrollo: RUP, XP y FDD*

*ProgramaciónSIG*

*Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica*2004

*Pruebas de programas*1994

*Up2Maps*2009

*ventajas y desventajas del PHP*2010

---

## BIBLIGRAFÍA

Retrieved diciembre from definicion-de-tic.html 2010

Diego, J. (2009). *Up2Maps*. Retrieved from <http://www.hatsnew.com>

*Dracones A Higher-Level component framework for MapServer*. (n.d.). Retrieved 2010, from <http://surveillance.mcgill.ca/dracones/>

*elwebmaster.com*. (2008, diciembre). Retrieved from <http://www.elwebmaster.com/articulos/up2maps-herramienta-para-la-creacion-de-mapas-tematicos>

*Google Earth Blog*. (2008, junio). Retrieved from [http://es.gearthblog.com/blog/archives/2008/06/thematic\\_mapping\\_engine\\_plugin.html](http://es.gearthblog.com/blog/archives/2008/06/thematic_mapping_engine_plugin.html)

IGAC. (1995). *Conceptos Básicos sobre Sistemas de Información Geográfica y aplicaciones en Latinoamérica*.

*Mapa Educativo*. (2008). Retrieved Diciembre 2010, from <http://www.mapaeducativo.edu.ar/encontrar/?a=qsig>

NCGIA, N. C. (1990). *Introducción a los SIG*.

*NETBEANS*. (2011). Retrieved from [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html)

*OSGeo*. (2010). Retrieved from <http://www.osgeo.org/qgis>

S, C. V. (n.d.). *¿Qué es el php?* Retrieved from <http://www.maestrosdelweb.com>

*Tecnología y Synergix. Visión de los Sistemas de Información y la Ingeniería del Software*. (n.d.). Retrieved from <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>

---

## ANEXOS

### GLOSARIO DE TÉRMINOS

**Cartografía:** Es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás. Por extensión, también se denomina cartografía a un conjunto de documentos territoriales referidos a un ámbito concreto de estudio.

**Framework:** Se define como un marco de trabajo, o sea como un diseño reutilizable de todo o parte de un sistema, representado por un conjunto de componentes abstractos, y la forma en la que dichos componentes interactúan.

**XML:** Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium.

**IDE:** Integrated Development Environment (Entorno de Desarrollo Integrado). Es un programa compuesto por un conjunto de herramientas para un programador.

**CASE:** Ingeniería de Software Asistida por Computadora, por sus siglas en inglés (Computer Aided Software Engineering), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

**API:** Interfaz de Programación de Aplicaciones, por sus siglas en inglés (Application Programming Interface).

**HTML:** HiperText Markup Language (Lenguaje de Marcación de Hipertexto) es un lenguaje que se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes.