

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 6



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

Título: Análisis de un Sistema de Información Geográfica para la
representación de los centros universitarios del país.

Autor: Yanet Pedroso Caraballo

Tutor: Ing. Yordany Piñeiro Gómez

La Habana, Junio de 2011

“Año 53 de la Revolución”



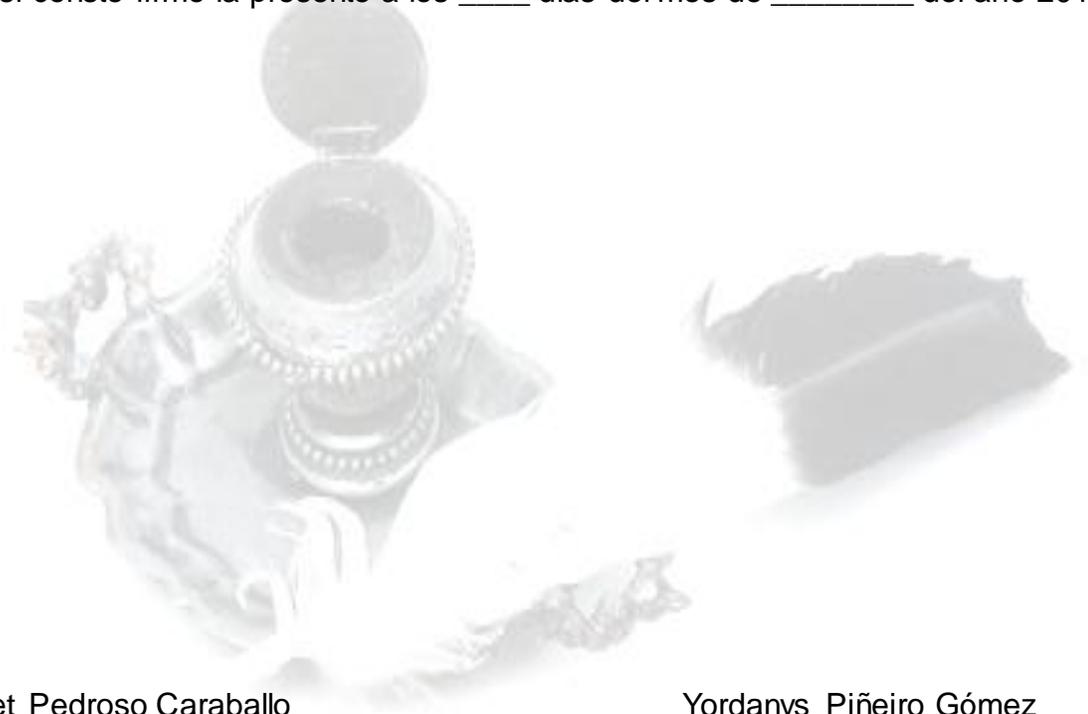
*"La tarea que los graduados de la UCI tienen por delante es grandiosa.
Espero que la cumplan y la cumplirán."*

*Fidel Castro Ruz
17 de junio de 2007*

DECLARACIÓN DE AUTORÍA

Declaro ser autora del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas (UCI) y al Departamento de GEySED a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2011.



Yanet Pedroso Caraballo

Yordanys Piñeiro Gómez

Firma del Autor.

Firma del Tutor.

DEDICATORIA

Quiero dedicarle esta tesis a la persona más importante de mi vida, a la persona que me ha dedicado todo de sí, a mi guía, mi estrella, mi mejor amiga, mi razón de ser y de existir,

Por ser mi ejemplo a seguir

El más grande de los amores de mi vida: A mi mamita linda que tanto amo y que siempre amaré mas allá de la eternidad (Te adoro y siempre estaré muy orgullosa de ti).

A mi papá que lo quiero tanto y además por sentirse siempre tan orgulloso de mí.

AGRADECIMIENTOS

Quiero agradecer a todas las personas que han estado a mi lado en este recorrido y que de una forma u otra han tenido que ver en mi desempeño.

Agradecer a la Revolución Cubana y a Fidel Castro por haberme permitido estudiar en esta universidad de excelencia.

A Dios.

A mi mamita linda por estar siempre ahí para mí en mis buenos y malos momentos, por haberme sabido guiar siempre y hacer de mí lo que soy hoy, por darme siempre esperanzas y aliento para seguir adelante y ser mi ejemplo a seguir y sentirme tan orgullosa de ser su hija.

A mi papá por quererme tanto y sentirse siempre tan orgulloso de mí.

A mi abuela Bertha por darme tanto amor en mis años de niñez y ayudarme tanto, por siempre estar orgullosa de mí.

A mi hermanito Noslen por ser siempre mi guardián y estar siempre ahí para mí cuando lo necesitaba.

A mi corazón pequeño mi sobrina Yaneidys por darme tantos momentos de felicidad.

A mis demás abuelos por darme siempre tanto amor y cariño.

*A mi hermano Yusniel porque a pesar de nuestro encuentro tardío se ha convertido
en mi confidente y mi mejor amigo.*

*A mi novio lindo Alioski por haberme acompañado durante estos cinco años,
haberme dado tanto amor y fuerzas para seguir adelante y darme cada día razones
para amarlo tanto.*

A mi ahijado Brayan por darme tantos momentos alegras.

*Al niño (Henry) por ser un hermano más y siempre haberme elegido para que este a
sus lado*

*A mi amiga Eliadis por estar conmigo desde la secundaria y ser mi mayor
confidente.*

*A la jungla por darme tantos momentos de alegría aún cuando parecía que no se
podía más (Noli, Yaima, Nora y Mara).*

*A las niñas de mi primer apartamento (Lusmey, Elizabetha, Maiden, las Yanet,
Grechin y Leidi).*

*A Dairon por ser siempre mi mejor amigo y estar presente siempre y ayudarme
tanto con sus consejos y su infinito cariño.*

A mis amigos del pre (Migue, Alien, Robe, Adrian y Zaila).

A mi tutora por ser la mejor de las tutoras del mundo, por haberme enseñado y ayudarme tanto en esta etapa, le estaré agradecida toda mi vida.

A los secuaces (Norqui, Yimi, Pedrito, Ismael y Wilian) por darme tantos y tantos momentos de alegría.

A mis vecinos por apoyarme tanto, especialmente a Bone, Marisol, Maribel, Mireya y.

A mis suegros por apoyarme y sentirse siempre orgullosos de mí

A mis primos especialmente a Eliecer, Pope y Diarei.

A mi maestra Rosa por ser la mejor de todas las maestras y quererme tanto.

A mi profesor Lázaro por proporcionarme mi mejor etapa de preuniversitario.

DATOS DEL CONTACTO.

TUTOR: Ing. Yordanys Piñeiro Gómez.

Graduado de Ingeniería en Ciencias Informáticas en Julio de 2007.

Profesor de Ingeniería de Software I y II.

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 27, Apto: 201.

Teléfono Oficina: +53 – 7 – 837 2559 Teléfono Apto: +53 – 7 – 835 8867.

E-mail: ypineirog@uci.cu

Síntesis del Tutor:

Profesión: Profesor de Ingeniería de Software de la Facultad 6 de la Universidad de las Ciencias Informáticas.

Categoría docente: Instructor

Años de graduado: 3

RESUMEN

Hoy, cuando inicia el siglo XXI, Internet ha logrado que se alcance lo que nunca antes fue posible, producir mapas instantáneamente de cada ciudad, cada pueblo y cada calle de todas las regiones del mundo conformándose así los SIG, los cuales se han convertido en una importante herramienta para la toma de decisiones en cualquier rama de la sociedad entre las que se encuentra el sector de la Educación Superior.

El presente trabajo de diploma tiene como objetivo realizar la documentación técnica correspondiente al Análisis de un Sistema de Información Geográfica (SIG) para la representación de los centros universitarios del país así como de su información socioeconómica. Para ello se estarán utilizando métodos científicos como el de análisis y síntesis, el histórico-lógico, el causal, el de modelación así como las entrevistas. Para la documentación técnica que se desprende de la realización de este trabajo de diploma se estará utilizando como metodología de desarrollo de software a RUP, como lenguaje de modelado a UML y como herramienta CASE Visual Paradigm, lo que permitirá un correcto entendimiento entre el cliente y el equipo de desarrollo. El desarrollo de esta investigación tiene un aporte muy importante para Cuba ya que a través de la misma se prevé proporcionarle al país un sistema con el cual no cuenta en estos momentos y resulta de tanta importancia para la búsqueda de información en el sector de la Educación Superior.

Palabras Claves.

Información, información socioeconómica, centros universitarios, Sistemas de Información Geográfica

ÍNDICE

INTRODUCCIÓN.....	13
CAPÍTULO 1	18
Introducción.....	18
1.1 Conceptos asociados al dominio del problema.	19
1.2 Objeto de estudio.....	22
1.3 Situación Problemática.	28
1.4 Análisis de las soluciones existentes.	29
1.5 Conclusiones Parciales.	36
CAPÍTULO 2	36
Tendencias y tecnologías actuales.....	37
2.1 Metodologías de Desarrollo de Software.	37
2.1.2 Metodologías Tradicionales de Desarrollo.....	38
2.1.2.1 Rational Unified Process (RUP).....	38
2.1.3 Metodologías Ágiles de Desarrollo.....	42
2.1.3.1 Programación Extrema (XP).....	42
2.2 Lenguaje Unificado de Modelado (UML).....	43
2.3 Herramientas CASE.....	45
2.3.1 Rational Rose	46
2.3.2 Visual Paradigm.....	48
2.4 Observaciones del estudio de las Tendencias y Tecnologías.....	49
2.4.1 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.	49
2.4.2 Visual Paradigm como base en el desarrollo de la solución.	50
CAPÍTULO 3	51
Análisis del Sistema.....	51
Introducción.....	51
3.1 Modelo de Dominio en el Desarrollo de Software.....	51
3.1.1 Definición de las clases del modelo del dominio	52
3.1.2 Breve descripción del diagrama	54
3.2 Requisitos	55
3.2.1 Requisitos Funcionales.	55
3.2.2 Requisitos no Funcionales.	57

3.3 Modelo de Casos de Uso del Sistema	60
3.4 Descripción textual de Casos de Uso del Sistema	61
Conclusiones Parciales.	65
CAPÍTULO 4	66
Introducción.....	66
4.1 Métodos de estimación:.....	67
Análisis Costo-Beneficio del proyecto.	74
4.2 Validación de Requisitos.....	75
Conclusiones Parciales.	79
REFERENCIAS BIBLIOGRÁFICAS	81
BIBLIOGRAFÍA	84
ANEXOS	¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1. Modelo de datos raster y vectorial.....	23
Figura 2. GeneSIG	32
Figura 3. SIG de la Universidad de las Ciencias Informáticas.....	33
Figura 4. Disciplinas, Fases e Iteraciones de la Metodología RUP	39
Figura 5. Diagrama de Clases del Dominio	52
Figura 6. Modelo de Casos de Uso del Sistema	61
Figura 7. Matriz de trazabilidad.....	78

INTRODUCCIÓN

Dibujar el mundo y obtener información del territorio donde vivimos ha sido desde siempre una tarea auto impuesta de los seres humanos, en un comprensible afán de entender el lugar en el que habitan, su más cercano entorno y la más alejada geografía. Primero hubo que descifrar el terreno y conocer sus accidentes; después medir las distancias hasta los pueblos cercanos para ayudar al comercio; más tarde se empezó a viajar y a conocer lo que había mucho más allá, incluso de la imaginación.

Y todo se fue plasmando en mapas, que poco a poco fueron integrando las aguas y las tierras del planeta. Hoy, cuando inicia el siglo XXI, Internet ha logrado que se alcance lo que nunca antes fue posible, producir mapas instantáneamente de cada ciudad, cada pueblo y cada calle de todas las regiones del mundo. La revolución digital de finales del siglo pasado permitió que la información geográfica se volviera más accesible para la mayoría de las personas.

Con la presencia de nuevas tecnologías y la necesidad del uso de estas en el manejo de la información geográfica en el mundo se fue desarrollando el ámbito perfecto para la creación y aparición de los Sistemas de Información Geográfica, los cuales tuvieron dos factores importantes para el surgimiento: la necesidad del uso y el manejo de la información geográfica, y la aparición de las primeras computadoras en el mundo. Hoy en día se plantean problemas a diario que solamente pueden ser resueltos por la geografía o nuevas tecnologías que permiten la obtención y uso de la información en diferentes campos de la sociedad.

A medida que avanza el desarrollo del conocimiento humano y las crecientes necesidades de la sociedad actual las Tecnologías de la Información y las Comunicaciones (TIC) tienden a tener un lugar creciente en la vida humana y funcionamiento de las sociedades, por lo que la presencia de modelos, datos geográficos y la capacidad de manipular y almacenar estos datos se hacen imprescindibles para el entendimiento de una nueva sociedad inmersa en el desarrollo tecnológico. Su uso extendido en la aplicación de nuevos SIG hace que se facilite la ubicación de todos los elementos dentro del mundo actual.

Los primeros SIG aparecieron en los años 1950 con el surgimiento de los primeros elementos cartográficos, los que marcarían el inicio del largo camino de un amplio desarrollo en los aspectos

tecnológicos, así como técnicas y datos relacionados con la ubicación geográfica. “Los SIG son una tecnología que permite gestionar y analizar la información espacial, los usuarios pueden combinar datos e información real con datos locales, visualizarlos además de hacer consultas” (RODRÍGUEZ, y otros, 1998).

El tratamiento geográfico de la información ha cobrado un auge vertiginoso a escala mundial, cada día con mayores posibilidades de aplicación, y Cuba no ha estado ajena a esta realidad. El país cuenta con instituciones que desarrollan software de este tipo, dentro de las que se encuentran: La Universidad de las Ciencias Informáticas (UCI), la cual combina el estudio, la investigación y la producción. Esta universidad se ha convertido en uno de los centros de altos estudios más notables del país, no sólo por la calidad de su enseñanza, sino que desde sus inicios al calor de la Batalla de Ideas, ha contemplado entre sus principales actividades la producción de software, convirtiéndose en una fuente de ingreso importante para la economía del país y ha jugado un papel decisivo en el proceso de informatización que se lleva a cabo.

La universidad cuenta con diversos Centros de Desarrollo que se encargan de potenciar el trabajo en diferentes líneas de desarrollo, uno de ellos es precisamente GEySED, perteneciente a la Facultad 6, el cual hace pocos años viene trabajando en la implementación de Sistemas de Información Geográfica. En una de las visitas realizadas a la universidad, el Ministro de Educación Superior se acercó a este Centro de Desarrollo y planteó la necesidad que tiene el país de contar con un sistema que permita representar todas las universidades que forman parte del mismo y que contribuya a la búsqueda rápida y eficiente de todos los elementos, aspectos y características de las mismas.

A partir de la problemática identificada con anterioridad surge el siguiente problema científico, que guiará el desarrollo de esta investigación: ¿Cómo contribuir al desarrollo de un sistema que permita representar y consultar la información socioeconómica correspondiente a los centros universitarios del país?

Para darle solución a este problema se tiene como **objetivo general**: Elaborar la documentación técnica del análisis del Sistema de Información Geográfica (SIG) para la representación y consulta de la información socioeconómica de los centros universitarios del país, utilizando las técnicas y herramientas de la Ingeniería de Software seleccionadas.

Para darle solución al objetivo la investigación se centra en: El proceso de gestión de la información geográfica, lo que se toma como **objeto de estudio**, delimitando el **campo de acción** hacia: El proceso de gestión de la información geográfica de un Sistema de Información Geográfica para la representación de los centros universitarios del país.

Para el cumplimiento a este objetivo se han trazado las siguientes tareas:

1. Caracterizar el proceso de desarrollo de los Sistemas de Información Geográfica.
2. Argumentar el uso de la metodología de desarrollo a utilizar, lenguaje de modelado y herramientas CASE.
3. Elaborar la documentación técnica correspondiente al Modelo de Dominio para lograr familiarizarse con el entorno de trabajo que actualmente existe.
4. Describir los requisitos funcionales y no funcionales del SIG.
5. Elaborar la Matriz de Trazabilidad.
6. Elaborar la documentación técnica correspondiente al Modelo de Casos de Uso del Sistema del SIG.
7. Validar requisitos.

La **idea a defender** que se plantea es: Con la elaboración adecuada de la documentación técnica correspondiente al análisis del Sistema de Información Geográfica para la representación de los centros universitarios del país se garantizará un correcto diseño e implementación del mismo.

Para el desarrollo de esta investigación fue necesario recurrir a métodos científicos, tanto teóricos como empíricos, entre los teóricos se utilizaron el Histórico-Lógico, el de Análisis y Síntesis, el método Causal y el de Modelación. De los métodos empíricos solamente se utilizaron las entrevistas.

El método Histórico – Lógico se utiliza para investigar la existencia de otros SIG desarrollados para universidades a nivel nacional e internacional con el objetivo de ver qué herramientas fueron utilizadas en su realización, así como funcionalidades comunes entre estos SIG y de esta forma definir cuáles de ellas deberían formar parte de la solución.

El método de Análisis y Síntesis se define con el objetivo de realizar un análisis y síntesis de la documentación existente relacionada con el proceso de desarrollo de los Sistemas de Información Geográfico así como de las metodologías existentes.

El método Causal se emplea para el análisis de los factores que provocan la necesidad del desarrollo de un Sistema de Información Geográfica para las universidades del país.

Las Entrevistas se realizan a los especialistas interesados en el desarrollo de este sistema que se reunieron directamente con el Ministro de Educación Superior quien es el cliente de la aplicación a desarrollar en un futuro, así como entrevistas a otros especialistas que han venido trabajando en la temática de los SIG hace algunos años, con el objetivo de determinar qué funcionalidades debe tener la aplicación.

La investigación está estructurada de la siguiente forma:

Capítulo 1: Sistema de Información Geográfica. Fundamentación teórica. En este capítulo serán expuestos los principales conceptos asociados al problema en cuestión para lograr un mayor entendimiento del mismo. Se explican además todos los argumentos del objeto de estudio así como una síntesis de algunas soluciones existentes para el problema identificado.

Capítulo 2: Tendencias y Tecnologías. Este capítulo tratará acerca de las tendencias y las tecnologías que se utilizan en la actualidad a nivel internacional para el desarrollo de Sistemas de Información Geográfica. Se estudia aquí un conjunto de metodologías (ágiles o pesadas) que se manejan en el mundo para el desarrollo de software de forma general, con el objetivo de seleccionar de ellas, la más adecuada para guiar el proceso de desarrollo del software. También se expone qué lenguaje de modelado y herramientas CASE son las más adecuadas para la construcción de los artefactos correspondientes a cada uno de los flujos de trabajo.

Capítulo 3: Descripción de la solución propuesta. Se presenta la solución propuesta a partir de la descripción del modelo de dominio y la identificación de requisitos funcionales y no funcionales del

sistema, se identifican los casos de uso del sistema, se realiza una descripción textual de cada uno de ellos y se conforma el Modelo de Casos de Uso del Sistema.

Capítulo 4: Validación de requisitos. En este capítulo se realiza la validación de requisitos mediante las técnicas correspondientes y se realiza el estudio de la factibilidad del proyecto.

CAPÍTULO **1**

Fundamentación Teórica

Introducción

En la actualidad el progresivo avance de las tecnologías computacionales y el cada vez más grande crecimiento de la información conllevan a la necesidad de un mayor uso de los Sistemas de Información Geográfica, de ahí que se hayan convertido en herramientas muy populares y su impacto no tenga antecedentes, su utilidad no tiene límites y puede manifestarse en diversas actividades relacionadas con la tecnología de computadores para integrar, manipular y visualizar una gran variedad de datos capaces de crear una imagen de la geografía, medio ambiente y características socioeconómicas de una zona determinada estudiada. Hoy en día es común utilizar los SIG en una gran diversidad de actividades y su impacto es tal que abarcan desde el trazado básico de mapas hasta la exploración y desarrollo de recursos, desde la ordenación del medio ambiente hasta la planificación y administración de importantes sistemas e infraestructuras que forman parte de las distintas esferas de la vida, sin dejar de mencionar uno de sus objetivos más importantes, la toma de mejores decisiones.

Precisamente, en este capítulo se abordan conceptos y se caracterizan de manera general a los Sistemas de Información Geográfica así como los aspectos relacionados con el objeto de estudio de la investigación. Se explica además el problema existente y se realiza un análisis detallado de las soluciones existentes.

1.1 Conceptos asociados al dominio del problema.

Con el objetivo de que el lector pueda tener una comprensión mayor de los temas que se relacionan en el capítulo, se describen detalladamente a continuación un grupo de conceptos asociados al dominio del problema, entre los que se destacan: Información, Sistemas de Información, Información Geográfica, Sistema de Información Geográfica y Datos espaciales.

Información

La información tiene muchos y variados conceptos según la ciencia o el punto de vista donde se esté viendo, como una definición general se tiene que la información es un conjunto organizado de datos, que constituyan un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje. Desde el punto de vista de la computación, la información es un conocimiento explícito extraído por seres vivos o sistemas expertos como resultado de la interacción con el entorno. Es el resultado de procesar, recabar, manipular y organizar datos. La información es un conjunto de mecanismos que se le suministran al individuo para que pueda entender e interpretar su entorno y actuar sobre él con mayor seguridad. La información es un factor fundamental que se representa a través de símbolos, específicamente en forma de datos binarios. (Babylon)

Información geográfica.

Se denomina Información Geográfica (IG) a aquellos datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos datos espaciales suelen llevar una información alfanumérica asociada. Se estima que el 80% de los datos corporativos existentes en todo el mundo poseen esta componente geográfica". (YAGÜEZ, y otros, 2002)

Sistema de Información.

Es el conjunto de elementos orientados a la administración y tratamiento de información y datos, organizados y listos para su posterior uso, con el fin de cubrir una necesidad.

Un sistema de información cuenta con varios elementos entre los que se encuentran.

- Personas.
- Datos.
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos proporcionando información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización. El sistema de información basado en computadoras es el campo de estudio de las tecnologías de información, con un conjunto de componentes relacionados que recolectan, procesan, almacenan y distribuyen información para apuntalar la toma de decisiones y el control de una organización.

Hay tres actividades fundamentales dentro de un sistema de información que producen la información necesaria para controlar operaciones, analizar problemas y crear nuevos productos o servicios. Estas actividades son:

- Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.
- Procesamiento: convierte esa entrada de datos en una forma más significativa.
- Salida: transfiere la información procesada a la persona que la usará o a las actividades para las que se utilizará.
- Los sistemas de información también requieren retroalimentación, que no es más que la salida que se devuelve al personal adecuado de la organización para ayudarle a evaluar y corregir la etapa de entrada.

Sistema de información Geográfica (SIG)

“Un Sistema de Información Geográfica (SIG) o GIS, en su acrónimo en inglés [Geographic Information System]) es una integración de hardware, software y datos geográficos diseñada 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. (NCGIA, 1990) También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre, y construido para satisfacer unas necesidades concretas de información.

Se puede decir que “en un sentido más estricto es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada y en un sentido más genérico los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y representar los resultados de todas estas operaciones” (Bosque, 1992). Es por eso que hoy en día los SIG son de vital importancia para la vida del hombre ya que hace mucha más rápida la búsqueda de cualquier información dentro de un mapa.

Datos espaciales

Los datos geográficos representan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen. **Un dato espacial** es una variable asociada a una localización del espacio, que ofrece información sobre la localización y las formas de un objeto geográfico y las relaciones entre ellos, normalmente con coordenadas y topología.

“Los datos espaciales refieren a entidades o fenómenos que cumplen los principios básicos:

- Tienen posición absoluta: sobre un sistema de coordenadas (x, y, z).
- Tienen una posición relativa: frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado).
- Tienen una figura geométrica que las representan (punto, línea, polígono).

Tienen atributos que lo describen (características del elemento o fenómeno)”. (Guerra, 2009)

1.2 Objeto de estudio.

Sistema de Información Geográfica (SIG)

Los Sistemas de Información Geográfica desde sus inicios presentan gran relación con la tecnología medio – ambiental, y es muy probable que esto se deba a que en este campo fue su primera aplicación ya que el primer sistema de este tipo fue desarrollado en Canadá a partir del año 1964 y estaba orientado principalmente al inventario de recursos naturales. En la actualidad existen pocos ámbitos en la vida humana en los que no se estén aplicando o puedan aplicarse estos sistemas, han pasado del total desconocimiento a la práctica cotidiana en el mundo de los negocios, en las universidades, etc., usándose para resolver problemas diversos. “Los SIG son una tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato”. (RODRÍGUEZ y otros, 1998)

Para poder tener una definición estándar de lo que es un SIG la NCGIA¹ lo definió como:

“Una integración organizada de hardware, software, datos geográficos y personal, diseñada para capturar, almacenar, manejar, analizar, modelar y representar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión”. (NCGIA, 1990)

En la práctica, un SIG es un sistema informático que tiene la capacidad de realizar una gestión completa de datos referenciados. Por referenciados los mapas tienen coordenadas reales asociadas, lo que permite manejar y realizar análisis con datos reales como áreas, longitudes o perímetros. Todos estos datos asociados a los mapas más los que los usuarios quieran añadirles los gestiona una base de datos que viene integrada al SIG.

¹ **NCGIA:** Acrónimo en inglés del Centro Nacional de Análisis de Sistemas de Información Geográfica en EE.UU.

Características de un SIG

“Según el modelo de datos implementados en cada sistema, se distinguen tres grandes grupos de Sistemas de Información Geográfica: SIG **Vectoriales**, SIG **Raster** y SIG con modelo de datos **Orientados a Objetos**. Fundamentalmente la mayoría de los sistemas existentes en la actualidad pertenecen a los dos primeros grupos (vectoriales y raster) (Ver Figura 1).

SIG Vectorial: Son aquellos Sistemas de Información Geográfica que utilizan pares de coordenadas relativas a algún sistema cartográfico para la descripción de los objetos geográficos. Emplea segmentos de líneas discretos o puntos para identificar lugares. Los objetos vectoriales no necesariamente ocupan un área y no todos los lugares necesitan referenciarse en el modelo.

SIG Raster: Los Sistemas de Información Raster basan su funcionalidad en una concepción implícita de las relaciones de vecindad entre los objetos geográficos. El modelo raster divide toda el área de estudio en una red regular de celdas con una secuencia específica. La secuencia convencional es fila por fila desde la esquina izquierda superior. El modelo vectorial solo indica dónde ocurre todo”. (Ortiz, 2007)

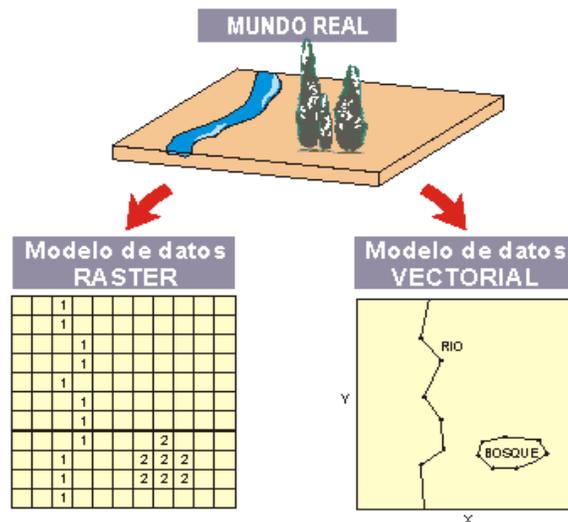


Figura 1. Modelo de datos raster y vectorial.

Componentes de un SIG

Un SIG es un sistema de hardware, software y procedimientos diseñados para soportar la captura, administración, manipulación, análisis y graficación de datos u objetos referenciados espacialmente, para resolver problemas complejos de planeación y dirección. Dando una definición más resumida y sencilla es: Un sistema de computador capaz de mantener y usar datos con localizaciones exactas en una superficie terrestre.

La composición de los SIG está dividida en cinco partes que son fundamentales: el hardware, el software, la información, el personal capacitado y los métodos de desarrollo.

Los SIG requieren de varios tipos de computadoras, desde equipos centralizados hasta configuraciones individuales o de red, su organización ha de ser de un hardware suficientemente específico y fuerte para cumplir con las necesidades de aplicación. Los componentes principales del software SIG son:

- Sistema de manejo de base de datos.
- Una interface gráfica de usuarios (IGU) para el fácil acceso a las herramientas.
- Herramientas para captura y manejo de información geográfica.
- Herramientas para soporte de consultas, análisis y visualización de datos geográficos.

El componente más importante para un SIG es la información. Se requiere de apropiados datos de soporte para que el SIG pueda resolver los problemas y contestar a preguntas de la forma más acertada. La recolección de los datos es un proceso largo que frecuentemente demora el desarrollo de productos que son de utilidad. Los datos geográficos y alfanuméricos pueden obtenerse por recursos propios u obtenerse a través de proveedores de datos. Mantener, organizar y manejar los datos debe ser política de la organización.

Si no se cuenta con los especialistas en manejar el sistema y desarrollar planes de implementación del mismo, sin el personal experto en su desarrollo, si la información es desactualizada y se maneja erróneamente, el hardware y el software no se manipulan en todo su potencial, las tecnologías SIG son de valor limitado. Además, hay que agregar que para que un SIG tenga una implementación

triunfante debe basarse en un correcto diseño y deben de estar bien definidas las reglas de actividad, las cuales son los modelos y prácticas operativas exclusivas en cada organización.

Importancia de los SIG

La tecnología SIG permite almacenar y manipular información usando geografía, analizar patrones, relaciones, y tendencias en la información, todo con el interés de contribuir a la toma de mejores decisiones.

Debido a la disminución en el costo de los Sistemas Informáticos, están materializándose importantes beneficios económicos en las empresas y entidades que implementan esta tecnología SIG. Entre estos beneficios se destacan:

- Realizar un gran número de manipulaciones, sobresaliendo las superposiciones de mapas en corto tiempo, transformaciones de escala, la representación gráfica y la gestión de bases de datos, así como su administración y mantenimiento.
- Consultar rápidamente las bases de datos, tanto espacial como alfanumérica, almacenadas en el sistema, con información exacta, actualizada y centralizada.
- Realizar pruebas analíticas complejas rápidas y repetir modelos conceptuales en despliegue espacial, sin la necesidad de repetir actividades redundantes o tediosas.
- Minimizar los costos operacionales e incrementar la productividad.
- Ayudar en la toma de decisiones con el fin de focalizar esfuerzos y realizar inversiones más efectivas.
- Almacenar información espacial como series temporales para visualizar la evolución de parámetros físicos y socio-económicos.
- Comparar eficazmente los datos espaciales a través del tiempo (análisis temporal).
- Efectuar algunos análisis, de forma rápida que hechos manualmente resultarían largos y molestos.
- Integrar en el futuro, otro tipo de información complementaria que se considere relevante y que esté relacionada con la base de datos nativa u original.

Aplicación de los SIG

La principal aplicación que tiene un Sistema de Información radica fundamentalmente en su capacidad de construir modelos y la representación del mundo real para más tarde poder utilizar esos modelos en la simulación de los efectos que un proceso de la naturaleza produce sobre un determinado escenario en un momento dado. La construcción de modelos constituye un instrumento muy eficiente para analizar las tendencias y determinar los factores que las influyen así como para evaluar las posibles consecuencias de las decisiones de planificación sobre los recursos existentes en el área de interés.

Funcionalidades de los SIG

Dentro de las funciones básicas de un sistema de información se puede describir la captura de la información la cual se logra mediante procesos de digitalización, procesamiento de imágenes de satélite, fotografías, videos, procesos aerofotogramétricos, entre otros.

Otra función básica de procesamiento de un SIG hace referencia a la parte del análisis que se puede realizar con los datos gráficos y no gráficos, se puede especificar la función de contigüidad de objetos sobre una área determinada, del mismo modo, se puede especificar la función de coincidencia que se refiere a la superposición de objetos dispuestos sobre un mapa. Pero también los SIG tienen funcionalidades comunes, a las cuales se hace referencia a continuación:

“Visualizar coordenadas: Permite que el usuario pueda visualizar las coordenadas del punto donde se encuentra situado el cursor en el mapa.

Visualizar escala: Permite que el usuario pueda visualizar a través de una representación gráfica y valores numéricos la escala en la cual está representado el mapa.

Realizar zoom gradual: Permite que el usuario aumente o reduzca el mapa según sus necesidades de definición del mismo o de determinada información reflejada en este.

Visualizar scroll direccional: Permite que el usuario realice movimientos sobre el mapa hacia el norte, sur, este u oeste.

Visualizar mapa guía temático: Permite que el usuario visualice el mapa general en una imagen pequeña en la cual se visualiza el espacio geométrico donde se está realizando determinada acción

sobre el mapa.

Realizar Zoom: Permite que el usuario pueda realizar las diferentes funciones de zoom sobre el mapa.

- **Realizar Zoom In:** Permite que el usuario realice un zoom hacia dentro de la página centrada en la posición puntual o área definida por el usuario, permitiendo así que aumente el nivel del zoom al hacer clic sobre la vista.
- **Realizar Zoom Out:** Permite que el usuario realice un zoom hacia fuera de la página centrada en la posición puntual o área definida por el usuario, permitiendo así que disminuya el nivel del zoom al hacer clic sobre la vista.
- **Realizar Zoom Extends:** Permite que el usuario pueda obtener una vista del mapa completo en su versión inicial. Muestra la vista en su extensión gráfica completa.

Medir distancia: Permite que se puedan realizar trazos por el mapa y visualizar la distancia total del primer al último vértice trazado.

Calcular Área: Permite que se puedan realizar trazados formando una región determinada, para poder visualizar el cálculo del área de la misma.

Calcular Perímetro: Permite que se puedan realizar trazados formando una región determinada, para poder visualizar el perímetro de la misma.

Imprimir mapas

Permite que el usuario imprima la foto del mapa sobre el cual se encuentra trabajando.

Mostrar Leyenda: Permite que se pueda visualizar la leyenda correspondiente al mapa en que se encuentra trabajando.” (Guerra, 2009)

Ciclo de desarrollo de un SIG

“Las aplicaciones SIG presentan la particularidad del tipo de información con que trabajan, esta los diferencia totalmente de cualquier otro tipo de sistema de información, porque la información geográfica está determinada por su posición, es decir, un sistema de coordenadas que la ubican respecto a la superficie terrestre y un conjunto de datos adicionales que no pueden obviarse en la

implementación de los SIG. Por tanto, el desarrollo de estas aplicaciones tampoco se comporta de igual forma que el de otros sistemas de información, el ciclo de desarrollo de un SIG comprende”. (Comas, y otros, 1993)

- “Evaluar las funciones del SIG que se necesita implementar y la información geográfica que será representada.
- Diseñar conceptualmente el SIG.
- Realizar el estudio de la información disponible.
- Realizar el estudio del hardware y software SIG existente en el mercado.
- Planificar y diseñar detalladamente la base de datos.
- Construir la base de datos.
- Realizar las pruebas pilotos de referencia.
- Adquirir el hardware y el software necesario para implementar el SIG.
- Integrar todos los subsistemas que conforman el SIG.
- Desarrollar la aplicación SIG
- Realizar el mantenimiento del SIG.” (Fornaris, 2009)

1.3 Situación Problemática.

En los últimos tiempos el tratamiento geográfico de la información ha cobrado un auge vertiginoso a escala mundial, cada día tiene mayores posibilidades de aplicación gracias al desarrollo de la aparición de los Sistemas de Información Geográfica.

La creación de un SIG viene dada por la necesidad de contar con una herramienta de ayuda para representar y manipular la información en relación con su posición geográfica. Teniendo en cuenta la facilidad que brinda tener la información georeferenciada, un SIG es la solución correcta para aquellos que desean obtener cualquier tipo de datos de un mapa, pudiendo realizar búsquedas temáticas, filtros

por capas, cálculos de distancia, acercamientos, navegación en el terreno, localización estructural, entre otras muchas funcionalidades.

En los últimos tiempos se ha incrementado exponencialmente el uso de los SIG debido a sus diversas aplicaciones en diferentes sectores de la sociedad, lo cual está íntimamente relacionado con el incremento de la efectividad de un sistema de este tipo.

En este contexto surge en la Universidad de Ciencias Informáticas, específicamente en la facultad 6 el proyecto GeneSIG, cuyo objetivo fundamental ha sido desarrollar una plataforma que sirva de base para la construcción de diversos SIG. La personalización de esta plataforma a diferentes negocios que requieran un SIG se encuentra a cargo del proyecto SIG-Aplicativos, perteneciente al centro GEySED de la facultad 6. En el mismo se han trazado líneas de investigación para abordar las diferentes áreas y temáticas del mundo de los SIG, dentro de las cuales se encuentra el desarrollo de SIG para todas las esferas y ramas de la sociedad actual, entre las que se encuentra la Educación, donde actualmente se desconoce el soporte y las facilidades que pueden brindar el desarrollo de los mismos.

En una de las visitas realizadas por parte del Ministro de Educación Superior a la universidad, se estuvieron exponiendo un conjunto de productos que se han sido desarrollados para la informatización de la misma y como parte de estos se encontraba el Sistema de Información Geográfica-UCI (SIG-UCI), desarrollado por el proyecto Aplicativos-SIG, y conociendo las potencialidades que brinda la utilización de un sistema de este tipo planteó la necesidad que tiene el país de contar con un SIG para la representación y consulta de la información socioeconómica correspondiente a los centros universitarios del país.

1.4 Análisis de las soluciones existentes.

La tecnología SIG constituye un impacto social muy importante en casi todos los sectores de la sociedad moderna. Gran parte de la información tratada por instituciones y empresas tienen en alguna medida relación con datos espaciales, lo que demuestra que la toma de decisiones depende en gran parte de la calidad, exactitud y actualidad de esta información espacial.

La puesta en marcha de una solución SIG en cualquier negocio social o gubernamental, posibilita el seguimiento y control de la información socioeconómica de sus objetivos sobre una ubicación geográfica, de manera que se garantice una elevada eficiencia en la toma de decisiones a partir de su representación y análisis espacial. Un sistema que integre la gestión de sus recursos sobre mapas, no

solo brinda resultados desde el punto de vista de la ubicación espacial, sino que amplía el escenario de evaluación de variables económicas mediante nuevos criterios de medida y factores geográficos. Uno de los sectores donde son utilizados los SIG es en la Educación Superior. A nivel internacional existen muchos sistemas de este tipo destinados a representar la información socioeconómica de cada universidad, sin embargo en la actualidad no existe un sistema de este tipo implementado en el país, y solo la Universidad de las Ciencias Informáticas cuenta con uno que sirve de base para conocer qué tipo de información le interesarían a las universidades representar de manera general. Existe además una plataforma que constituye la base para el desarrollo de aplicaciones SIG destinadas a cualquier negocio social o interinstitucional, lo cual es muy factible para conceptualizar soluciones integrales en un período corto de tiempo asegurando buena calidad, posibilitando a los clientes la modelización de las variables de su accionar a través de su localización geográfica y asegurar un exitoso proceso de toma de decisiones.

A continuación se realiza una descripción sintetizada de algunas de las soluciones mencionadas con anterioridad.

GeneSIG 1.0

La plataforma GeneSIG 1.0 surge como necesidad de contar con un producto soberano que sirva como soporte al desarrollo de aplicaciones de Sistemas de Información Geográfica (SIG) en entornos libres. En los últimos tiempos se ha incrementado exponencialmente el uso de los SIG para la toma de decisiones, relacionado íntimamente con el incremento de la efectividad y la disminución de los costos que se obtienen con la puesta en marcha de un sistema de este tipo. GeneSIG es una plataforma para el desarrollo de SIG implementada con herramientas y tecnologías libres, cumpliendo además con las especificaciones OpenGis que establece el Open Geospatial Consortium (OGC) y en consecuencia con la política de migración al software libre y de soberanía tecnológica que impulsa el país. La plataforma tiene como objetivo fundamental realizar la representación geoespacial de la información asociada a negocios específicos, permitiendo además realizar análisis sobre dicha información. Presenta una arquitectura distribuida, que cuenta con una base cartográfica, y sobre ella un conjunto de objetos representados geoespacialmente que contienen información asociada, identificados a partir de las necesidades particulares de los clientes finales. Constituye un producto de gran actualidad y con grandes expectativas en el mercado internacional.

La plataforma cuenta con 9 módulos, que se explican brevemente a continuación:

Módulo de Navegación: En este módulo se gestiona toda la interacción del usuario con la interfaz visual donde se encuentra el mapa, garantizando que este pueda realizar las operaciones de movimiento, acercamiento y alejamiento en sus diferentes variantes.

Módulo de Selección: En este módulo se realizan las operaciones de selección de objetos geográficos dentro de las capas del mapa que sean seleccionables. Además, se ofrece la opción de desmarcar dichas selecciones y de realizar operaciones de consulta o persistencia de selección.

Módulo de Consulta: En este módulo se realizan consultas espaciales a objetos puntuales o los determinados por un área que defina el usuario.

Módulo de Análisis: En este módulo se realizan operaciones de análisis de la información espacial, como el cálculo de distancias, áreas, acimut, entre otras funciones de mayor complejidad.

Módulo de Configuración: En este módulo se realizan operaciones de configuración generales de la aplicación.

Módulo de Elementos Complementarios: En este módulo se visualizan u ocultan la escala, navegador virtual y mapa de referencia.

Módulo de Servicios: En este módulo se realizan operaciones relacionadas con la gestión de las capas, como selección y agregado de nuevas capas en diferentes formatos.

Módulo de Impresión: En este módulo se definen las características necesarias para exportar el mapa a formato .pdf para su posterior impresión.

Módulo de Edición: En este módulo se realizan operaciones de edición que permiten a los usuarios insertar nuevos objetos geográficos a capas temporales, además de otras funciones como la activación de puntos.

Servicios o ramas donde pueden implantarse personalizaciones de GeneSIG:

- Servicio informativo de transporte urbano, del correo postal, de turismo de ciudad y mejores ofertas de ventas.
- Control Epidemiológico.
- Defensa Civil.

- Gestión de recursos minerales e hidráulicos.
- Planificación Física.
- Planificación y control de flotas de Transporte.
- Gestión de redes (Datos, Telefónica, Eléctricas).
- Agricultura de precisión.
- Educación, etc.

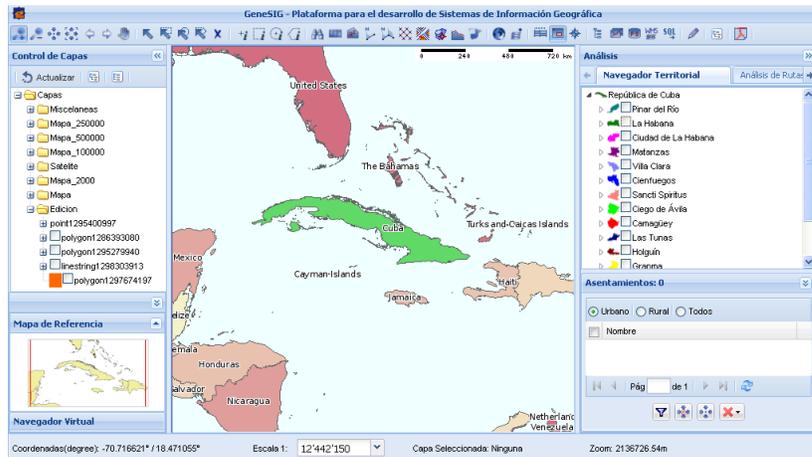


Figura 2. GeneSIG.

SIG UCI

La mayoría de las universidades importantes a nivel mundial cuentan con Sistemas de Información Geográfica ya que se considera que entre el 80 y el 90 por ciento de la toma de decisiones tiene involucrada un componente geo-espacial y no solo en las universidades sino de empresas, instituciones y del resto de las esferas de la sociedad. Uno de los objetivos de la Universidad de las Ciencias Informáticas es la automatización de los procesos involucrados en el quehacer cotidiano, de ahí que se haya incorporado el desarrollo de un SIG para la misma, que a medida que pasa el tiempo va creciendo en funcionalidades.

Actualmente cuenta con dos módulos: el módulo de administración del sistema que lleva consigo la tarea de integrar el sistema a las otras aplicaciones existentes y el módulo para la visualización de

mapas. También tiene definido dos roles principales para el uso del sistema y sus funcionalidades, ellos son: el usuario que puede solo visualizar los mapas y realizar operaciones básicas sobre dicho mapa y el administrador del sistema que puede realizar cualquier acción sobre el sistema, dígase creación de nuevos roles, inserción de funcionalidades o mapas, etc.

La página principal de la aplicación, se concibe como un portal, donde la funcionalidad principal ocupa la mayor parte de la página. La página cliente para el usuario común muestra en la parte superior izquierda un mapa de referencia, en la parte inferior izquierda aparece un treeview con las capas que contiene el mapa que se muestra en la parte superior aparecen combobox para la búsqueda y edits para localizar puntos. En el centro de la página aparece el navegador donde se encuentra el mapa y en la parte superior derecha aparece la leyenda. Ver Figura 3.

Para el diseño de las principales opciones se utilizaron imágenes acompañadas de textos que aparecen al desplazar el mouse sobre dichas imágenes y que describen la funcionalidad de estas. Las opciones se encuentran debajo de los combobox para realizar las búsquedas.

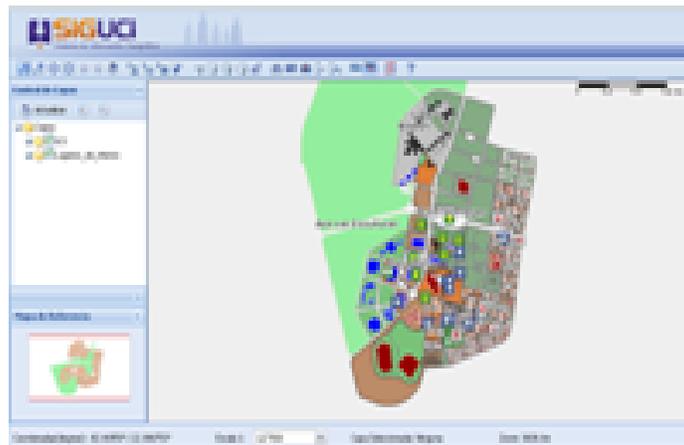


Figura 3. SIG de la Universidad de las Ciencias Informáticas

El SIGUCI está constituido por varias funcionalidades, algunas de estas son:

- Navegación: se encarga de gestionar toda la interacción del usuario con la interfaz visual donde se encuentra el mapa y garantiza que este pueda realizar las operaciones de movimiento, acercamiento (Zoom in), alejamiento (Zoom out).
- Buscar persona: ofrece la posibilidad de localizar geográficamente a personas, usando como

criterio de búsqueda el usuario, el número del solapín o el número del carné de identidad.

- Búsqueda temática: ofrece la posibilidad de localizar geográficamente los lugares, usando como criterio de búsqueda las estructuras.
- Búsqueda de edificios: ofrece la posibilidad de localizar geográficamente los edificios, usando como criterio de búsqueda el número de estos.
- Medir distancias: ofrece la posibilidad de medir las distancias entre puntos.
- Calcular área: ofrece la posibilidad de calcular áreas utilizando puntos para graficar las mismas.

SIG Universidad de California en Davis

Luego del estudio e investigación de diferentes SIG a nivel internacional destinados a la representación de la información socioeconómica de universidades específicas, se encontraron sistemas como el de la Universidad del Norte de Rusia, la Universidad de Colombia, Universidad del Norte de Inglaterra y la Universidad de California en Davis, esta última seleccionada para realizársele un análisis más profundo por poseer el sistema completitud en las funcionalidades que lo integran.

La Universidad de California en Davis (también conocida como UC Davis, UCD) es una universidad pública co-educacional localizada en Davis, California, al oeste de Sacramento la capital del estado de California. La universidad es parte del sistema de la Universidad de California. UC Davis no es sólo considerada una líder en investigación, sino también una de las mejores universidades del mundo. Cuenta con un SIG en el cual se representa toda la información ya sea geográfica o socioeconómica del centro. Entre las principales funcionalidades con que cuenta este sistema se encuentran: aplicar zoom, editar detalles, mostrar cantidad de estudiantes extranjeros existentes, obtener cursos de posgrado, obtener localización de la residencia, aulas etc.

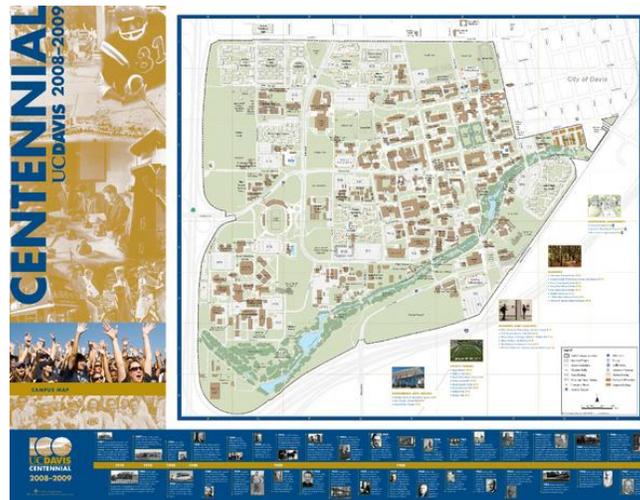


Figura 4: SIG de la Universidad de California en Davis.

1.4.1 Observación de las soluciones existentes

El estudio de un conjunto de Sistemas de Información Geográfica correspondientes a universidades a nivel internacional, así como del SIG de la Universidad de las Ciencias Informáticas y a su vez de la plataforma GENESIG permitió identificar el conjunto de funcionalidades básicas que debe poseer la aplicación que se desea construir. Se debe puntualizar que los SIG correspondientes a la universidades nacionales e internacionales solo podrían servir de base para la identificación del objetivo antes mencionado, no para su utilización, ya que son aplicaciones específicas de una u otra universidad en particular, presentando características muy específicas de información socioeconómica de interés para una u otra entidad educacional, de acuerdo a las características propias de cada país o región. Por su parte la plataforma GENESIG sí constituye el punto de partida para el desarrollo de la aplicación ya que proporciona una estructura altamente escalable, de manera tal que se le pueden seguir añadiendo, quitando o modificando funcionalidades, en pos de realizar personalizaciones para un negocio específico que lo requiera, que es el caso del sistema que se quiere desarrollar; esta plataforma desarrollada con software libre permite quitar todas las funcionalidades que no sean necesarias e incorporar las que el negocio estudiado requiera.

1.5 Conclusiones Parciales.

Los Sistemas de Información Geográfica son herramientas que permiten el análisis, manejo, consulta y representación de los datos geográficamente referenciados. A lo largo del presente capítulo quedaron caracterizados los SIG en cuanto a su definición, importancia, construcción, funcionamiento, así como su aplicación, lo que permitió profundizar en los elementos más representativos, que se deben tener en cuenta para desarrollar un sistema de este tipo, que sirva de base para la conceptualización y posterior desarrollo del SIG que se desea construir.

El análisis de los principales conceptos asociados al dominio del problema permitió tener un mayor entendimiento y comprensión de las temáticas que conforman la investigación así como de los factores que influyen en la situación problemática actual.

El estudio de las soluciones existentes contribuyó a concluir que no existe un Sistema de Información Geográfica a nivel nacional ni internacional, para la representación de forma integrada de la información socioeconómica de las universidades de un país. A su vez, sirvió de guía para conocer los intereses particulares de representación de determinada información que tiene cada universidad, lo que contribuirá en la correcta definición de las funcionalidades que debe poseer la aplicación a desarrollar.

CAPÍTULO 2

Tendencias y tecnologías actuales.

En la actualidad el crecimiento alcanzado en el desarrollo de software es inmenso, todo el personal involucrado en este proceso se ha visto en la obligación de estandarizar el trabajo de las aplicaciones que se crean. Se han desplegado una serie de metodologías para el desarrollo de software lo que posibilita que todos los miembros del proyecto puedan entenderse utilizando el mismo estándar. Además, se han desarrollado diferentes lenguajes de modelación orientado a objetos como es el UML. Las herramientas CASE (Computer Aided/Assisted Software/System Engineering) o (Ingeniería de Software Asistida por Computadora) no se han quedado atrás y al igual que las metodologías y lenguajes de modelado han tenido un amplio desarrollo; son un conjunto de herramientas que brindan soporte a un enfoque de ingeniería en el desarrollo de software en alguna o en todas las fases de este proceso.

En este capítulo se hace un estudio de metodologías de desarrollo de software, de un lenguaje de modelado y de herramientas para la construcción de software para realizar una comparación y proponer las más adecuadas para la elaboración del sistema propuesto.

2.1 Metodologías de Desarrollo de Software.

“Una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc.” (VALENCIA) Las metodologías y estándares utilizados en el desarrollo de software proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado. Cada una de las metodologías que existen actualmente tiene características específicas que las hacen singulares, por lo que clasificarlas sería una tarea engorrosa; a partir de su filosofía de desarrollo se pueden separar en pesadas o tradicionales y ligeras o ágiles que a continuación se describen. Entre las metodologías

más utilizadas en los últimos tiempos se encuentran Rational Unified Process (RUP) y Programación Extrema (XP), de las que se abordan algunas características en este epígrafe para poder argumentar así la selección realizada.

2.1.2 Metodologías Tradicionales de Desarrollo

Las metodologías tradicionales o pesadas del desarrollo de software se guían expresamente a través de un plan durante el proceso de desarrollo, en el cual se generan gran cantidad de artefactos y roles mediante un modelado y documentación detallada, apoyándose en poderosas herramientas de trabajo y notaciones definidas en la arquitectura del software. Las metodologías tradicionales se caracterizan por estar enfocadas principalmente al proceso definido por la metodología y no a las características del proyecto que se desarrolla.

A continuación se exponen algunas de las características fundamentales de una de las metodologías tradicionales que cuenta con un gran prestigio en su clasificación.

2.1.2.1 Rational Unified Process (RUP)

“La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, es una metodología cuyo fin es entregar un producto de software. Permite estructurar todos los procesos y medir la eficiencia de la organización. Es un proceso de desarrollo de software que utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Proporciona además disciplinas en las cuales se encuentran artefactos con los que se puede contar como guías para poder documentar e implementar de una manera fácil y eficiente para un buen desarrollo”. (Molpeceres, 2002)

Según RUP, el desarrollo del software se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de las cuales es desarrollada mediante ciclos de iteraciones.

- Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la línea base de la arquitectura.
- Construcción: En esta etapa el objetivo es obtener la capacidad operacional inicial.
- Transición: En esta etapa el objetivo es llegar a obtener el release del producto.

Cada una de estas etapas es desarrollada mediante ciclos de iteraciones, lo cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería o proceso y los tres últimos como de apoyo o soporte.

Estos se observan en la figura que se muestra a continuación:

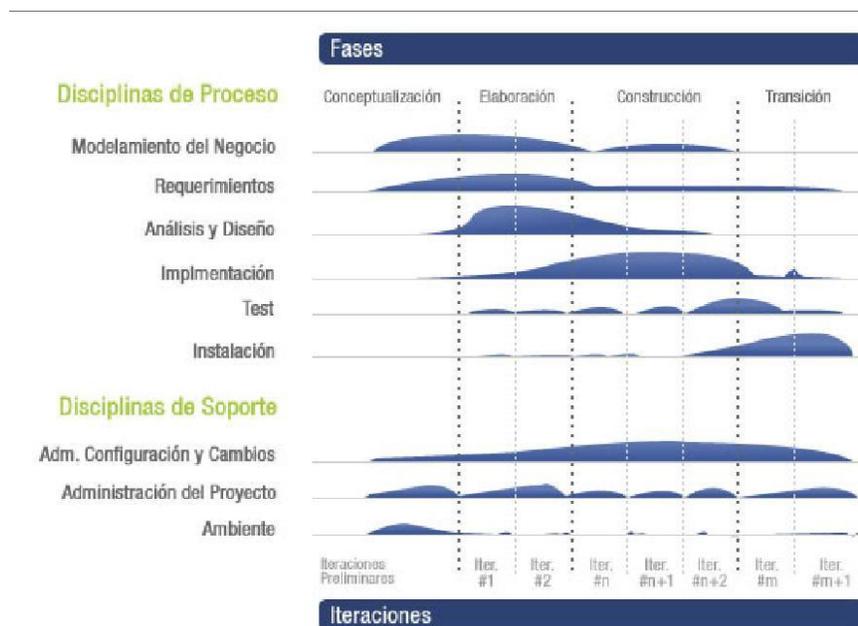


Figura 5. Disciplinas, Fases e Iteraciones de la Metodología RUP

“RUP es un proceso que está definido por tres características esenciales: dirigido por los casos de uso, iterativo e incremental y centrado en la arquitectura.

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requisitos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Centrado en la arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada, entre otros, por plataformas software, sistemas operativos, gestores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requisitos no funcionales. La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. Todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, recibe este nombre porque lo forman las vistas lógica, de implementación, proceso y despliegue, más la de casos de uso que es la que da cohesión a todas.

Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requisitos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de

trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada, es por eso que se dice que son miniproyectos.” (Pressman, 2005)

El RUP está basado en 5 principios:

- Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico, aunque se debe tener en cuenta el alcance del proyecto.
- Balancear prioridades: Debe encontrarse un balance que satisfaga los deseos de todos.
- Demostrar valor iterativamente: Los proyectos se entregan en etapas iteradas. En cada iteración se analiza la opinión, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- Elevar el nivel de abstracción: Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes de cuarta generación (SQL, lenguajes de consulta), o esquemas (frameworks). Esto previene a los ingenieros de software dirigirse directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.
- Enfocarse en la calidad: El control de calidad debe estar presente en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Las actividades de RUP se centran en crear y mantener modelos utilizando UML. Como no existe un único proceso que sea apropiado para todos los desarrollos, RUP es un proceso configurable. Se adapta tanto a grupos pequeños de desarrollo como a grandes organizaciones. Basándose en lo que se consideran las mejores prácticas de desarrollo de software, RUP resulta apropiado para una amplia gama de proyectos y organizaciones.

“RUP aplica 6 prácticas principales para el desarrollo de sistemas. Estas prácticas son:

- Desarrollo de software en forma iterativa: permite ir creciendo en el entendimiento del problema a través de refinamientos sucesivos. Esto también permite introducir cambios tácticos en los

requisitos, características del sistema o en los tiempos.

- Gestión de requisitos: es necesario para garantizar que al final el software tenga la calidad requerida.
- Uso de arquitecturas basadas en componentes: el uso de una arquitectura basada en componentes hace que el sistema pueda reutilizar componentes desarrollados con antelación y facilitar el trabajo que se realiza.
- Modelización visual del software: El proceso le demuestra cómo modelar visualmente software para capturar la estructura y el comportamiento de arquitecturas y de componentes.
- Verificación de calidad del software: RUP le asiste en el planeamiento, el diseño, la puesta en marcha, la ejecución, y la evaluación de las pruebas de confiabilidad, funcionalidad, performance de la aplicación y el sistema.
- Control de cambios: La capacidad de manejar los cambios, asegurándose que cada cambio sea aceptable, y pudiendo continuar con los mismos, es esencial en un ambiente en el cual el cambio es inevitable." (JACOBSON *et al.* 2000)

2.1.3 Metodologías Ágiles de Desarrollo

“Las metodologías ágiles se enfocan generalmente en intentar evadir las burocráticas guías de las metodologías tradicionales” (MARTÍNEZ, 2005). Centran sus procesos de desarrollo a los individuos que interactúan con el proyecto así como con el equipo de trabajo, obteniendo buenos resultados, es decir, se desarrolla un buen software en cuanto a su funcionamiento pero no genera una documentación consistente. Las metodologías ágiles están basadas en heurísticas provenientes de prácticas de producción de código y el cliente es parte del equipo de desarrollo, generan pocos artefactos, poseen pocos roles, además de no hacer énfasis en la arquitectura del software. A continuación se exponen algunas de las características de una de estas metodologías.

2.1.3.1 Programación Extrema (XP)

XP (Extreme Programming, por sus siglas en inglés) pertenece al grupo de metodologías ágiles, donde lo principal es la comunicación entre el personal implicado y la obtención de un resultado deseado en el menor tiempo posible.

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para

proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

En esta metodología se realizan pruebas unitarias, que sirven como método para adelantarse a obtener los posibles errores, se utiliza la refabricación, lo cual se basa en la reutilización de código y la programación en pares, es decir, dos desarrolladores trabajando en una misma estación de trabajo, lo cual constituye una peculiaridad de XP. Esta metodología le da el derecho al usuario de decidir qué se implementa, saber el estado real y el progreso del proyecto, añadir, cambiar o quitar requisitos en cualquier momento, y a los desarrolladores les permite decidir cómo se implementan los procesos, pedir al cliente en cualquier momento aclaraciones de los requisitos, estimar el esfuerzo para implementar el sistema, y muy importante, cambiar los requisitos en base a nuevos descubrimientos.

“Las principales características de esta metodología son:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Esta metodología empieza en pequeño y añade funcionalidad con retroalimentación continua, donde el manejo del cambio se convierte en parte sustantiva del proceso y el costo del cambio no depende de la fase o etapa. No introduce funcionalidades antes que sean necesarias además que el cliente o el usuario se convierte en miembro del equipo”. (Molpeceres,2002)

Lo fundamental en XP es que la comunicación, entre los usuarios y los desarrolladores siempre es íntegra, se logra la simplicidad al desarrollar y codificar los módulos del sistema y la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

2.2 Lenguaje Unificado de Modelado (UML)

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos. Muchas organizaciones

los usan en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo.

Hoy en día es considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo, mediante este es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. Se ha convertido en la actualidad en uno de los más utilizados por lo expresivo, claro y uniforme que resulta para el diseño orientado a objetos y además por permitir una fuerte integración entre las herramientas, los procesos y los dominios. Desde finales de 1997 es un lenguaje de modelado orientado a objetos estándar. Posibilita el intercambio de modelos entre las distintas herramientas CASE y mejora el tiempo total de desarrollo, además presenta una alta reutilización y minimización de costos.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Sus principales funciones son las de Visualizar, Especificar, Construir y Documentar. A través de este lenguaje se puede modelar lo mismo sistemas de software, hardware u organizaciones del mundo real. Un modelo UML está compuesto por tres clases de bloques de construcción: Elementos, Relaciones y Diagramas.

UML no puede compararse con la programación estructurada, pues UML significa (Lengua de Modelación Unificada), no es programación, solo se diagrama la realidad de una utilización en un requisito. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la orientación a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML para lenguajes orientados a objetos.

“UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

- Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado como son los Diagrama de clases, Diagrama de componentes, Diagrama de objetos,

Diagrama de estructura compuesta (UML 2.0), Diagrama de despliegue, Diagrama de paquetes.

- Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado, por ejemplo: Diagrama de actividades, Diagrama de casos de uso, Diagrama de estados.
- Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado como es el Diagrama de secuencia, Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x), Diagrama de tiempos (UML 2.0), Diagrama de vista de interacción (UML 2.0).” (Zuluaga,2008)

UML permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. También documenta todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.). Este lenguaje de modelado es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

2.3 Herramientas CASE

“Para superar las dificultades existentes en el uso de las diversas tecnologías, la industria de computadoras ha desarrollado un soporte automatizado para el desarrollo y mantenimiento de software. Este es llamado Computer Aided Software Engineering (CASE).” (Guerra, 2009)

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de un software. Los estados en el ciclo de vida de desarrollo de un software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.

Una herramienta CASE se utiliza para ayudar a las actividades del proceso de software que es utilizado para diseñar e implementar otro software. Facilitan mejoras en la calidad y productividad del diseño y desarrollo. Ayudan a automatizar metodologías de software y desarrollo de aplicaciones.

Estas herramientas proporcionan a los desarrolladores de software grandes ventajas: apoyan a las metodologías y métodos, mejoran la comunicación entre las personas que interactúan con el sistema

permitiéndoles compartir mejor su trabajo, establecen métodos eficientes para almacenar y utilizar los datos, hacen más eficaz el mantenimiento y automatizan fragmentos del análisis y diseño pesados y vulnerables a errores.

A continuación se abordan las características de dos de estas herramientas para así seleccionar una de ellas que garantice una mayor eficiencia en el desarrollo del software.

2.3.1 Rational Rose

Es una herramienta software para el modelado visual mediante UML de sistemas software. Permite Especificar, Analizar, Diseñar el sistema antes de Codificarlo. Brinda soporte al Proceso Unificado de Rational (RUP). Rational es un agrupamiento de metodologías y herramientas que abarca todos los aspectos del desarrollo de software, desde su concepción hasta la elaboración del producto. La metodología de RATIONAL propone el desarrollo de software basado en las mejores prácticas recopiladas en innumerables proyectos exitosos. Esta metodología esta compilada en el proceso de desarrollo Rational Unified Process.

“Entre sus principales características se encuentran:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Soporte Enterprise Java Beans 2.0.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requisitos de aplicación a través de diseños lógicos y físicos.

- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo".(Murillo, 1999)

Para ayudar en la implementación del proceso, Rational ha desarrollado un conjunto de herramientas integradas que permiten desarrollar las actividades del proceso y obtener una sinergia al integrarse entre las distintas herramientas, permitiendo a todo el equipo de desarrollo compartir y utilizar la información necesaria en el momento adecuado.

Rational provee herramientas para las siguientes fases del desarrollo de software:

1. Gestión de requisitos.
2. Modelado visual de sistemas basado en UML.
3. Desarrollo de aplicaciones Web y Java.
4. Pruebas de software.
5. Gerenciamiento de la configuración y el cambio.

Rational Rose Enterprise Edition es una herramienta que se puede encuadrar dentro del grupo de herramientas más técnicas debido a que se encarga de llevar a cabo tanto la automatización de los sistemas para la posterior generación de código (esto es, realización de los distintos diagramas y generación del código posterior), como para labores de ingeniería inversa, es decir, realización de los diagramas una vez conocido el código. Sin duda, en esto Rational Rose Enterprise Edition es una forma de ayuda para la comprensión del sistema y de sus distintos componentes, y lo mejor es que se puede aplicar ingeniería inversa a una multitud de códigos distintos, siempre que estén orientados a objetos. La clave está en la creación de componentes, los cuales van a contener una serie de archivos dentro de los cuales se encuentran las distintas clases que pertenecen a dicho componente. Mediante la especificación de la sintaxis que presentan dichos ficheros, se realiza de forma automática la ingeniería inversa.

Rational Rose presenta una pequeña desventaja, y es que necesita de mucha memoria para poder de

alguna forma ser manejado de forma rápida y eficiente.

2.3.2 Visual Paradigm

Visual Paradigm es una herramienta con un diseño centrado en casos de uso y enfocado al negocio que genera un software de calidad, tiene la particularidad de ser un lenguaje estándar común a todo el equipo de desarrollo y eso facilita por supuesto la comunicación. Posee capacidades de ingeniería directa e inversa, su modelo y el código se encuentran sincronizados en todo el ciclo de desarrollo, y está disponible en múltiples versiones y plataformas. Es muy personalizable, tiene un generador de mapeo de objetos-relacionales para los lenguajes de programación Java, .NET y PHP y se integra de manera sencilla con NetBeans. La versión gratuita no permite realizar ingeniería inversa, pero permite crear diagramas y generar código a partir de ellos.

“Visual Paradigm para UML es una herramienta UML 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 a una más 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.

Visual Paradigm Studio cuenta con las siguientes características.

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.
- Documento Modelado colaborativo con CVS y Subversión.
- Interoperabilidad con modelos UML 2.1.
- Código a modelo.
- Código a diagrama.
- Ingeniería inversa Java, C++.
- Generación de código.
- Modelo a código y diagrama a código.
- Editor de Detalles de Casos de Uso.

- Diagramas de flujo de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de bases de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML". (Paradigm, 2010)

2.4 Observaciones del estudio de las Tendencias y Tecnologías.

Luego de estudiar y plasmar las características particulares de metodologías de desarrollo ágiles y pesadas, del lenguaje de modelado y herramientas CASE, se hace necesario concluir cuáles de estas serán la base para el desarrollo de la solución.

2.4.1 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.

Este proceso de desarrollo de software junto al lenguaje unificado de modelado constituye la metodología estándar más utilizada para el análisis, implementación y documentación de los sistemas orientados a objetos. Esta metodología aplica varias de las mejores prácticas en el desarrollo moderno de software y cuenta además con diferentes elementos de planificación para el control del desarrollo de software y gestión de riesgos permitiendo prevenir y corregir problemas y fallos conocidos. Otra de las razones por la que ha sido escogida es que, provee un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Por ser abarcadora en la guía de procesos de desarrollo de software y además por ser completa en la determinación de la documentación ingenieril ha sido RUP seleccionada para la realización de este trabajo. Es significativo destacar que esta investigación solo comprende las disciplinas de Modelamiento del negocio y Requisitos, lo cual trae consigo que los flujos de trabajo de Análisis y

Diseño, Implementación y Prueba tienen que seguir desarrollándose y para ello se necesita que el equipo de desarrollo que continúe trabajando en el desarrollo de esta aplicación tenga una documentación exhaustiva del negocio y sistema de la aplicación en cuestión y precisamente esta metodología provee toda la documentación completa y exacta para la realización del sistema.

2.4.2 Visual Paradigm como base en el desarrollo de la solución.

Visual Paradigm es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Además de permitir dibujar todos los tipos de diagramas que se necesitan en esta investigación permite generar la documentación necesaria para el posterior diseño e implementación del sistema. Tiene también un diseño centrado en casos de uso y enfocado al negocio, pero uno de los aspectos más importantes de la selección de esta herramienta es que en la Universidad de Ciencias Informáticas se está trabajando en función de utilizar tecnologías libres y Visual Paradigm es una herramienta multiplataforma.

2.5 Conclusiones Parciales.

Tras realizar el análisis de las metodologías de desarrollo, herramientas CASE y lenguaje de modelado usados frecuentemente para la implementación de sistemas de forma general, así como de las características y particularidades de cada una de ellas, se puede concluir que la metodología a utilizar será el RUP, Visual Paradigm será la herramienta Case y UML el lenguaje de modelado. La integración de estas tecnologías garantizará que se puedan construir cada uno de los artefactos que servirán de base para el posterior diseño e implementación del producto que se desea construir. Las ventajas que cada una ofrece, garantizará que el equipo de desarrollo que continuará trabajando en el desarrollo de la aplicación pueda tener una documentación detallada de todo el proceso a seguir, para a partir de esta, exista un entendimiento entre los desarrolladores sobre lo que se quiere hacer. Además, se trabajará con una herramienta que al ser multiplataforma permitirá que no exista divergencia alguna con las políticas de utilización de software libre que sigue el país.

CAPÍTULO 3

Análisis del Sistema

Introducción

En el presente capítulo se enumeran y especifican los Requisitos Funcionales (RF) y los Requisitos No Funcionales (RNF) que debe cumplir el sistema, se identifican los actores y casos de uso del sistema para conformar el Diagrama de Casos de Uso del Sistema (DCUS). Unido a esto se realiza una descripción detallada de los casos de uso que forman parte de la solución.

3.1 Modelo de Dominio en el Desarrollo de Software

El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, el modelo de dominio es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El objetivo fundamental que se persigue al realizar el modelo de dominio es comprender y describir las clases más importantes dentro del contexto del sistema. Además, éste ayuda a los usuarios a mantener una terminología común, lo que les permite compartir el conocimiento de forma más consistente. El Modelo de Dominio es empleado fundamentalmente cuando los flujos de información son difusos, es decir, que tengan múltiples orígenes y cuando son solo eventos o sucesos. También la imposibilidad de realizar subsistemas en el proceso de desarrollo del software dado por las múltiples interconexiones, es uno de los factores que influyen en la decisión de realizar un modelo de este tipo. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo cual no significa que siempre que haya un Modelo de Dominio tenga que existir obligatoriamente un Modelo de Negocio. Teniendo en cuenta que no se tienen bien definidos los procesos del negocio se realizará una modelación del dominio y se procederá a explicar cada uno de

los conceptos que forman parte del mismo. Todo ello para tener una mejor comprensión de la estructura y dinámica de la organización, los problemas actuales dentro e identificar las mejoras potenciales.

Modelo del Dominio

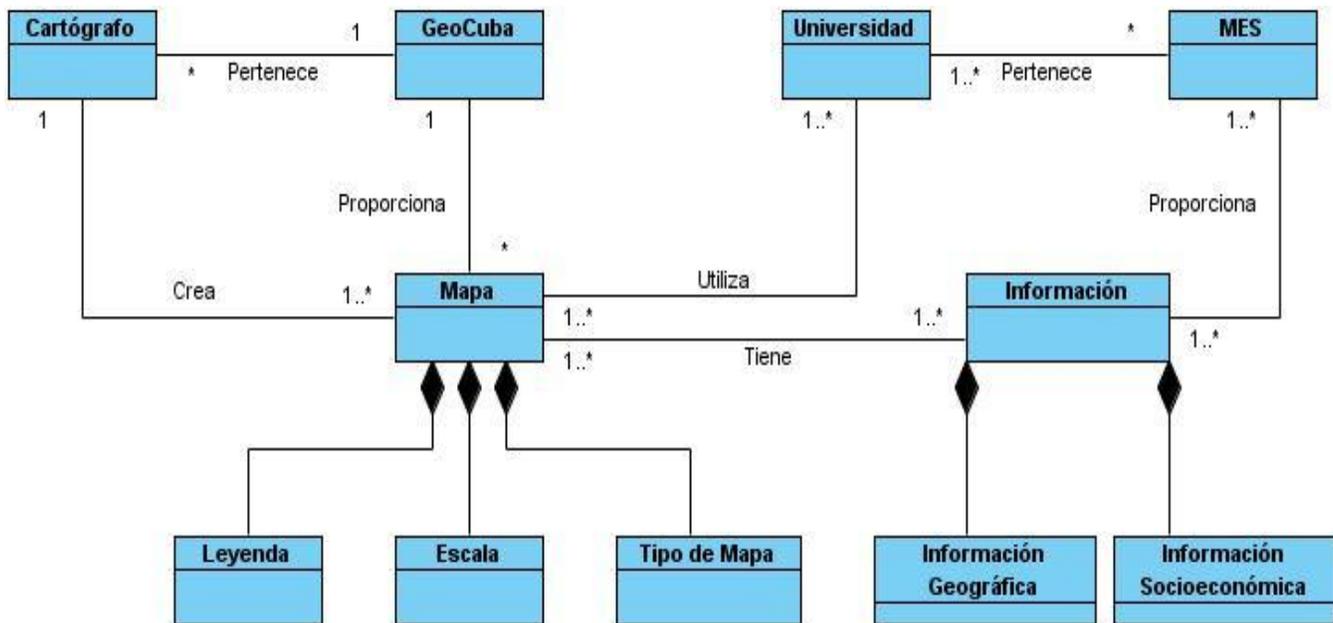


Figura 6. Diagrama de Clases del Dominio

3.1.1 Definición de las clases del modelo del dominio

Cartógrafo

Persona que se dedica profesionalmente a la realización de cartas geográficas, al estudio y elaboración de mapas.

GeoCuba

Es un grupo empresarial que se dedica a la elaboración, producción y venta de planos, mapas y cartas náuticas con diversos fines, así como a la realización de estudios geográficos, de impacto ambiental, e investigaciones científicas en ramas del campo de las geociencias, entregando a sus clientes, productos informativos terminados con una alta calidad y fiabilidad.

Usuario

Cualquier persona que tenga acceso a la aplicación y que a su vez pueda utilizar o consultar algún tipo de información referente a las universidades del país.

MES (Ministerio de Educación Superior)

Garantiza, con un claustro altamente comprometido con la Revolución, la formación integral y la superación continua de los profesionales revolucionarios que demanda la sociedad, con alta calidad y pertinencia. Desarrolla y promueve la ciencia, la cultura y la innovación tecnológica, como respuesta a las necesidades del desarrollo sostenible, en correspondencia con la universalización de la universidad, en los marcos de los programas de la Revolución. Este Ministerio está conformado por un grupo numeroso de universidades de todo el país.

Entidad

Empresa, organismo u organización que solicita un servicio determinado y que proporciona la información socioeconómica referente al mismo.

Mapa

Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El hecho de que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.

Escala

Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.

La relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa. Por extensión puede referirse a la mayor o menor profundidad del enfoque en un tema geográfico.

Leyenda

Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.

Tipo de Mapa

Clasificación que se le da a los mapas de acuerdo con su especificación.

Información Socioeconómica

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

Información Geográfica

Datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales.

3.1.2 Breve descripción del diagrama

GeoCuba es la empresa que determina todos los procesos geológicos desarrollados en Cuba, a la que pertenecen varios cartógrafos que son los encargados de crear o construir los mapas, que se le proporcionan a otras entidades y distinguen a dicha empresa.

Los mapas están compuestos por varias escalas representativas, leyendas que permiten un mejor entendimiento de los mismos y la tipografía, que está referida a la variedad de mapas existentes de acuerdo con su especificación, que pueden ser de División Político-Administrativa, Vegetación, Yacimientos Geológicos, entre otros. También estos mapas tienen en su haber toda la información socioeconómica, tratada en los distintos tipos de datos referentes a todos los sectores de la economía y la sociedad.

Los mapas son utilizados, además, por los diferentes usuarios que deseen consultar cualquier tipo de información geográfica de una o varias universidades, pertenecientes al MES, este último unido a las universidades del país proporcionan toda la información socioeconómica que se le agregará posteriormente a los mapas.

3.2 Requisitos

“Esta es la etapa más crucial del desarrollo de un proyecto de software. Los requisitos son el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software, es la Ingeniería de Requisitos, donde se lleva a cabo el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones, del sistema de software que se desea producir; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requisitos del software. Los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar.” (Martínez, 2004)

Al lograr identificar los requisitos iniciales del software y realizar el análisis de los mismos mediante la técnica de verificación, donde se limaron las ambigüedades y la falta de consistencia, se arrojaron los requisitos funcionales y no funcionales que se presentan a continuación.

3.2.1 Requisitos Funcionales.

“Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, incluyendo las acciones.”(Pressman, 2005) que podrán ser ejecutadas por el usuario, así como las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el mismo.

Los requisitos que debe cumplir el SIG son los siguientes:

RF1: Realizar Zoom.

Se quiere que el usuario pueda realizar las diferentes funciones de zoom sobre el mapa.

RF 1.1 Realizar zoom +.

RF 1.2 Realizar zoom -.

RF 1.3 Realizar zoom extenso.

RF 1.4 Realizar zoom previo.

RF 1.5 Realizar zoom siguiente.

RF2: Recentrar mapa.

Se quiere que el usuario pueda recentrar el mapa según la posición puntual que realizó sin modificar la escala del mapa.

RF3: Mover mapa.

Con este requisito se quiere que el usuario pueda recentrar el mapa variando con el puntero del ratón la posición de la vista que se presenta.

RF4: Redimensionar mapa.

Con este requisito se quiere que el usuario una vez introducidos los valores (x, y) de las coordenadas visualice la región deseada en el centro del mapa.

RF5: Localizar universidad.

Con este requisito se quiere que el usuario pueda localizar en el mapa la universidad seleccionada por él.

RF6: Localizar Estructura.

Con este requisito se quiere que el usuario pueda localizar estructuras específicas que forman parte de una universidad determinada.

RF7: Localizar persona.

Con este requisito se quiere que el usuario pueda localizar una persona en una universidad determinada utilizando varios criterios.

RF8: Localizar Carrera.

Con este requisito se quiere que el usuario pueda localizar en el mapa en qué universidades se encuentra la carrera seleccionada por él.

RF9: Visualizar plano de universidad.

Con este requisito se quiere que el usuario pueda visualizar el plano de la universidad seleccionada.

RF10: Obtener Ranking de puntuación de los estudiantes.

Con este requisito se quiere que el usuario pueda obtener el ranking de puntos de los estudiantes de cualquier facultad de una universidad determinada.

RF11: Obtener ranking de promoción de universidades.

Con este requisito se quiere que el usuario pueda visualizar el ranking de un tema determinado en una universidad seleccionada.

RF12: Mostrar ayuda.

Con este requisito se quiere que el usuario pueda visualizar la ayuda que muestra el sistema con todos

sus íconos y barras para una mayor información de las universidades o de una universidad en específico.

RF13: Utilizar mapa de referencia.

Permite que el usuario pueda moverse por el mapa, en un rango de escala determinado a partir de un rectángulo equivalente al extend visualizado.

RF14: Utilizar navegador virtual.

Permite que el usuario pueda mover el mapa en las ocho direcciones principales: norte, sur, este, oeste, noreste, sureste, noroeste, suroeste.

RF15: Visualizar escala gráfica.

Con este requerimiento se quiere que el usuario pueda visualizar y ocultar la escala gráfica sobre el mapa.

RF16: Medir distancia.

Con este requerimiento se quiere que el usuario pueda realizar trazos por el mapa y que visualice la distancia total acumulada así como la existente entre los dos últimos vértices trazados.

RF17: Calcular área.

Con este requerimiento se quiere que el usuario pueda a partir del trazado de una región poder visualizar el perímetro de la misma.

RF18: Calcular perímetro de una región.

Con este requerimiento se quiere que el usuario pueda a partir del trazado de una región poder visualizar el área de la misma.

RF19: Exportar mapa

Este caso de uso se realiza con el objetivo de exportar un mapa o vista de éste a un fichero en formato pdf. Incluye la configuración de la página y demás elementos para garantizar la posterior impresión

RF20: Realizar selección.

Con este requerimiento se quiere que el usuario pueda seleccionar información de las capas del mapa que sean seleccionables; esta selección puede ser: puntual, radial, rectangular o poligonal

3.2.2 Requisitos no Funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido y confiable. Por lo tanto la aplicación debe cumplir con las siguientes propiedades:

Usabilidad

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.
- El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

Fiabilidad

- La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores. La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas.
- El tiempo medio de reparación, en caso de un fallo es de 7 días.

Eficiencia

- El tiempo de respuesta estará dado por la cantidad de información a procesar, oscilando entre 5 y 7 segundos.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación y oscilará entre 10 y 20 segundos.

Soporte

- La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

Restricciones de diseño

- Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

- 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).
- Se debe lograr un producto altamente configurable, teniendo en cuenta que se desarrollará sobre la plataforma GENESIG, que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

Interfaz

Interfaces de usuario

- El sistema debe tener una apariencia profesional y un diseño gráfico sencillo; posibilitarle al usuario la configuración del entorno de trabajo y ser intuitivo.

Interfaces de hardware

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 256 MB de memoria RAM.
- Se requiere al menos 1 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 2GB de disco duro.
- El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

Interfaces de software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software:

Para las PCs clientes:

- Un Navegador como Mozilla Firefox, Zafari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows y Mac OS.

Para los Servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 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.
- PgRouting como extensión de PostgreSQL para análisis de rutas.
- MapServer 5.2.2 o superior, con extensión PHP mapscript.

3.3 Modelo de Casos de Uso del Sistema

Representa gráficamente a los procesos y su interacción con los actores. Después de la identificación de los actores que interactúan con la aplicación, y la agrupación en casos de uso del conjunto de requisitos identificados con anterioridad, se conforma el Modelo de Casos de Uso que se presenta a continuación:

Actor	Descripción
Usuario	Cualquier persona que tenga acceso a la aplicación y que a su vez pueda utilizar o consultar algún tipo de información referente a las universidades del país.

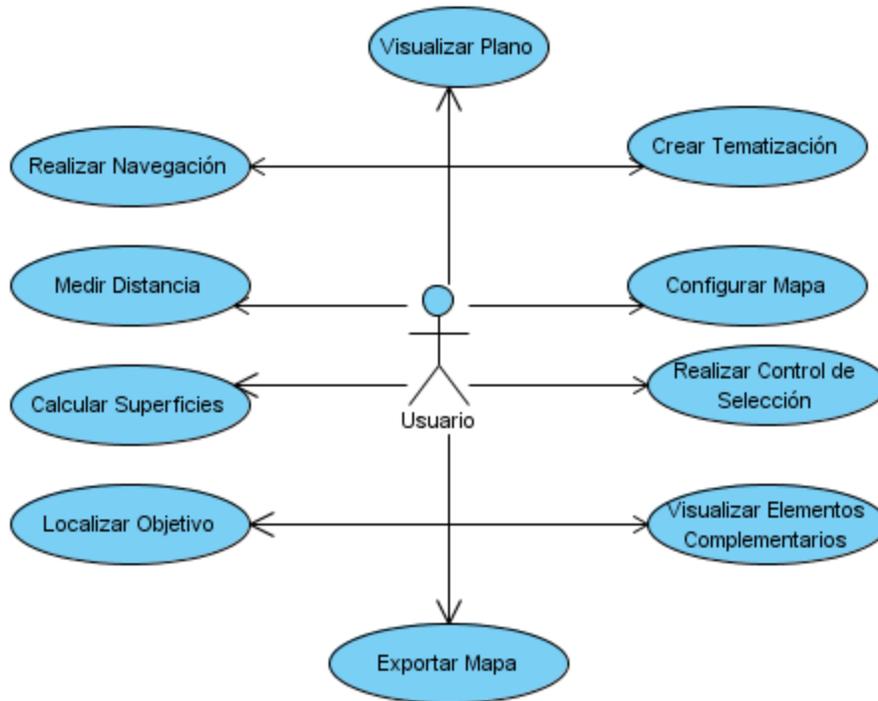


Figura 7. Modelo de Casos de Uso del Sistema

3.4 Descripción textual de Casos de Uso del Sistema.

En el presente epígrafe se realiza la descripción textual del caso de uso del sistema **Localizar objetivo**, las demás descripciones textuales se encuentran en los Anexos (Ver Anexo 4).

Especificación de los casos de uso:

Localizar objetivo.

Caso de Uso:	Localizar objetivo.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con la finalidad de localizar objetivos (personas, universidades, estructuras) en el mapa.
Resumen:	El caso de uso se inicia cuando el usuario selecciona el objetivo que desea localizar y termina cuando el sistema muestra en el mapa el resultado de la búsqueda.
Precondiciones:	El usuario tiene que estar autenticado.
Referencias	RF 5, RF 6, RF 7, RF 8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el usuario selecciona el objetivo que desea localizar:</p> <ul style="list-style-type: none"> • Localizar estructura. • Localizar persona. • Localizar universidad. • Localizar carrera. 	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> • Si selecciona “Localizar estructura”, ver sección “Localizar estructura”. • Si selecciona “Localizar persona”, ver sección “Localizar persona”. • Si selecciona “Localizar universidad”, ver sección “Localizar universidad”. • Si selecciona “Localizar carrera”, ver sección “Localizar carrera”.
	<p>3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario y actualiza la visualización del mapa.</p>

Prototipo de Interfaz



Sección “Localizar Estructura”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de Localizar Estructura.	2. El sistema muestra un menú en el cual se encuentran todas las estructuras definidas para visualizar en el mapa.
3. El usuario selecciona la estructura que desea visualizar.	4. El caso de uso termina cuando el sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa mostrando en este la ubicación de la estructura seleccionada.

Prototipo de Interfaz



Sección “Localizar Persona”

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

<p>1. El usuario selecciona la opción Localizar persona.</p>	<p>2. El sistema muestra una ventana en la cual se visualizan diferentes criterios de búsqueda que le permita al usuario localizar a una persona. Estos criterios pueden ser: Carné de Identidad, Facultad y Provincia.</p>
<p>3. El usuario selecciona el o los criterios de búsqueda que desee emplear.</p>	<p>3. El caso de uso termina cuando el sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa mostrando en él la ubicación de la persona en el mapa, además de mostrar una breve información del mismo como nombre, edad, facultad a la que pertenece, provincia, municipio etc.</p>

Prototipo de Interfaz

Sección “Localizar carrera”

Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona la opción Localizar carrera.</p>	<p>2. El sistema muestra un combobox con todas las carreras que se estudian en el país.</p>
<p>3. El usuario selecciona la carrera que desea visualizar en el Mapa.</p>	<p>4. El caso de uso termina cuando el sistema procesa la información a partir de la acción realizada por el usuario y actualiza la</p>

	visualización del mapa mostrando la localización de la carrera seleccionada a lo largo del país, visualizando además una breve información de la misma: universidad a la que pertenece, años de estudio, matrícula de estudiantes, etc.
Sección “Localizar Universidad”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Localizar universidad.	2. El sistema muestra combobox con todas las universidades del país.
3. El usuario selecciona la(s) universidad(es) de la que quiere conocer la ubicación.	4. El caso de uso termina cuando el sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa mostrando la localización de la(s) universidad(es) con una breve información de su creación, matrícula de estudiantes, facultades, etc.

Conclusiones Parciales.

El desarrollo de este capítulo permitió que a partir de la construcción del Modelo de Dominio se pudiera tener un entendimiento del negocio que se desea automatizar, lo cual serviría como punto de partida para identificar las funcionalidades que el sistema debe tener, que satisfaga las necesidades del cliente, las cuales fueron agrupadas en casos de uso para conformar el Modelo de Casos de Uso, lo cual permitió tener una vista global del sistema a desarrollar. Además, garantizó que a partir de las descripciones textuales de los casos de uso identificados se tenga la idea exacta de qué se quiere llevar a cabo con cada funcionalidad, a partir de la acción del actor y la respuesta que debe dar el sistema. Por su parte se reflejaron los requisitos no funcionales garantizando que se establecieran las bases mínimas de las propiedades o cualidades que debe tener el producto que se quiere construir.

CAPÍTULO 4

Estudio de factibilidad y validación del sistema

Introducción

La comprensión de los requisitos en un problema cualquiera es una de las más difíciles tareas que enfrenta un ingeniero de software. Preguntas como ¿el cliente no sabe lo que quiere?, ¿los usuarios finales no deberían entender bien las características y funciones que les proporcionarán un beneficio? se encuentran entre las incógnitas que en muchas ocasiones arrojan respuestas negativas, ya que estos requisitos pueden cambiar durante el transcurso y desarrollo del proyecto por lo cual es necesario un correcto proceso de validación de los mismos para corregir errores.

Por otra parte, tener un Plan de Proyecto garantiza la recogida de la información requerida para dirigir el proyecto. Un Plan de Proyecto es una guía para la administración del proyecto y sus actividades, que se encarga además de establecer planes razonables para desarrollar la Ingeniería de Software y manejar los cambios de los proyectos de Software. Para que un plan de proyecto se cumpla exitosamente debe haber entre otras cosas una correcta planificación, la cual pueda manejar los cambios de los proyectos de Software y además estimar los resultados del proyecto y los valores de costo, tiempo y recursos requeridos mediante el proceso de la información para llegar a estimaciones razonables.

En el presente capítulo se lleva a cabo el estudio de la factibilidad del proyecto mediante los métodos existentes para esto y se hará además la validación de requisitos utilizando para ello las técnicas de validación de requisitos más apropiadas.

4.1 Métodos de estimación:

Como se mencionaba anteriormente, uno de los objetivos de la planificación es lograr estimaciones razonables. Existen técnicas para la estimación de costo, recurso y tiempo, relacionadas al desarrollo del software, que se muestran a continuación:

1. Análisis de Puntos de Casos de Uso: Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso. La estimación mediante el Análisis de Puntos de Casos de Uso, es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

2. COCOMO II. El SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Este método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

Después de un estudio de los métodos de estimación se escogió el método de Análisis de Puntos de Casos de Uso ya que aplicando inicialmente el Análisis de Puntos de Función de Casos de Uso, se podrá obtener una estimación bastante imprecisa debida principalmente a la escasa información que se tiene sobre el software al principio de un proyecto, pero permitirá obtener una idea del esfuerzo necesario para llevar adelante el mismo, y podrá ser refinada a medida que se obtenga más información. Pero si se aplica nuevamente el Análisis de Puntos de Función sobre una descripción más detallada de los Casos de Uso, la estimación del tamaño y esfuerzo será más precisa que la anterior. Por lo que cada vez que se aplique se obtendrá más perfecta y acertada la estimación para la factibilidad del proyecto. A continuación se muestra como se aplicó esta técnica para el cálculo de la estimación para la factibilidad de la investigación en curso.

Paso 1. Identificar los Puntos de Casos de Uso Desajustados

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Para calcular UAW

$$\text{UAW} = \sum \text{Cantidad de actores} * \text{peso}$$

$$\text{UAW} = 1 * 3$$

$$\text{UAW} = 3$$

Tipo	Descripción	Peso	Cantidad de actores * peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1*3
Total			3

Para calcular UUCW

Tipo	Descripción	Peso	Cantidad de casos de uso de ese tipo * peso
------	-------------	------	---

Capítulo 4: Estudio de factibilidad y Validación del sistema

Simple	El caso de uso contiene de 1 a 3 transacciones.	5	$5*5 = 25$
Medio	El caso de uso contiene de 4 a 7 transacciones.	10	$3*10 = 30$
Complejo	El caso de uso contiene más de 8 transacciones.	15	$2*15 = 30$
Total			85

Luego: $UUCP = 3 + 85 = 88$

Paso 2. Ajustar los Puntos de Casos de Uso

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Para calcular TCF

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor}_i \text{ es un número del 0 al 5)}$$

Significado de los valores

- 0: No presente o sin influencia.
- 1: Influencia incidental o presencia incidental.
- 2: Influencia moderada o presencia moderada.
- 3: Influencia media o presencia media.
- 4: Influencia significativa o presencia significativa.
- 5: Fuerte influencia o fuerte presencia del actor.

	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido.	2	4	El sistema es altamente distribuido.	8
T2	Objetivos de performance o tiempo de respuesta.	1	4	La velocidad no es limitada por las entradas previstas por el usuario. Los	4

Capítulo 4: Estudio de factibilidad y Validación del sistema

				tiempos de respuesta deben ser altos.	
T3	Eficiencia del usuario final.	1	1	El usuario final requiere la adecuada preparación para usar el sistema.	1
T4	Procesamiento interno complejo.	1	5	Hay cálculos complejos.	5
T5	El código debe ser reutilizable.	1	5	Se requiere que el código sea reutilizable.	5
T6	Facilidad de instalación.	0.5	3	La instalación es relativamente sencilla.	1.5
T7	Facilidad de uso.	0.5	4	El sistema cuenta con una interfaz gráfica amigable.	2
T8	Portabilidad.	2	0	El sistema debe ser multiplataforma, siendo usado bajo los Sistemas Operativos Windows y Linux.	0
T9	Facilidad de cambio.	1	3	Se requiere un costo moderado de mantenimiento.	3
T10	Concurrencia.	1	5	Hay concurrencia. Algunos procesos dependen de otro para su funcionamiento.	5
T11	Incluye objetivos especiales de seguridad.	1	3	Seguridad normal.	3
T12	Provee acceso directo a terceras partes.	1	5	Los usuarios web tienen acceso directo.	5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1	Pocos usuarios internos, sistema fácil de usar.	1
Total					44

$$TCF = 0.6 + 0.01 * 44$$

$$TCF = 1.04$$

Capítulo 4: Estudio de factibilidad y Validación del sistema

Para calcular EF

$EF = 1.04 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$ (Donde Valor_i es un número del 0 al 5).

Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	El grupo está bastante familiarizado con el modelo.	6
E2	Experiencia en la aplicación.	0.5	4	La mayoría del grupo ha trabajado mucho tiempo en esta aplicación.	2
E3	Experiencia en orientación a objetos	1	4	La mayoría del grupo programa en objetos.	4
E4	Capacidad del analista líder	0.5	0	No se contrató a un especialista.	0
E5	Motivación	1	4	El grupo está altamente motivado.	4
E6	Estabilidad de los requerimientos	2	2	Se esperan cambios.	4
E7	Personal part-time	-1	5	El tiempo no será full time.	-5
E8	Dificultad del lenguaje de programación	-1	2	Se usará php	-2
Total					20

$$EF = 1.4 - 0.03 * 20$$

$$EF = 0.8$$

$$\text{Luego UCP} = 88 * 1.04 * 0.8 = 68.64$$

Paso 3. Calcular esfuerzo del flujo de trabajo de Implementación

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de Conversión.

Para calcular CF

CF = 20 horas-hombre (si $Total_{EF} \leq 2$).

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$).

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$).

$Total_{EF} = Cant_{EF} < 3$ (entre E1 – E6) + $Cant_{EF} > 3$ (entre E7, E8).

Como

Total_{EF} = 2 + 0

Total_{EF} = 2

CF = 20 horas-hombre (porque $Total_{EF} \leq 2$)

Luego **E = 68.64* 20 horas-hombre**

E = 1372.8 horas-hombre

Paso 4. Calcular esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	343.2-hombre
Diseño	20%	686.4-hombre
Implementación	40%	1372.8 horas-hombre
Prueba	15%	514.8 horas-hombre
Sobrecarga	15%	514.8 horas-hombre
Total	100%	3432 horas-hombre

El valor del esfuerzo calculado representa el esfuerzo del flujo de trabajo de Implementación, y a partir del mismo se calculan los esfuerzos correspondientes a los demás flujos de trabajos. La suma de ellos es el **esfuerzo total (E_T)**, que en este caso es de 3432 horas-hombre.

En este caso es muy factible realizar el proyecto propuesto ya que cuando uno calcula el Factor de Conversión, está tratando de buscar cuáles de los factores pudieran estar afectando el desarrollo del mismo, lo podría retrasar o podría hacer incluso que éste no fuera viable y que por tanto haya que tomar algún tipo de medida radical. En el resultado de este factor influye la Cant $EF < 3$ (entre E1 –E6) ya que en este rango se encuentran aquellos factores que mientras tengan un valor más alto, cuyo resultado se acerque a 5 potenciará el hecho de que se realice un menor esfuerzo y además indicará que el equipo de proyecto está sumamente preparado para enfrentar este reto, es decir, que existe una alta motivación, una gran capacidad de su analista líder, una gran estabilidad en los requisitos, que tienen experiencia en la aplicación, etc., por lo que se escogen de ellos precisamente los que son menores que 3, que serían aquellos que pudieran afectar el desarrollo exitoso del proyecto, los que pueden influir en que el resultado final sea negativo. De igual forma pasa con los factores E7 y E8, pero en sentido contrario, mientras más se acerquen a 0 indica que el equipo es capaz de enfrentar el desarrollo de este proyecto pues querrá decir que el equipo no tiene dificultad alguna en el lenguaje de programación que emplearán en el desarrollo del software y que el equipo estará a full time, es decir, estará trabajando a tiempo completo en su ejecución, en este caso se escogen los mayores que 3 pues tienen el efecto contrario, se trata de escoger lo peor que puede pasar para llegar a un resultado final, que dirá que si $Total_{EF} \leq 2$ como es este caso la factibilidad es muy alta.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 20 días de trabajo; la cantidad de horas que puede trabajar una persona en 1 mes es de 160 horas.

Si $E_T = 3432$ horas-hombre y por cada 160 horas se tiene 1 mes, eso daría un $E_T = 21.45$ mes-hombre

Esto quiere decir que 1 persona puede realizar el proyecto analizado en 1 año y 9 meses aproximadamente.

Si en el equipo hay 4 personas y todas realizan el mismo esfuerzo entonces el proyecto analizado puede terminarse en aproximadamente 5 meses y medio (5.3625 meses).

Después de realizados estos cálculos de estimación se llega a la conclusión de que la duración del proyecto será de cinco meses y será un proyecto factible por lo que se considera que el riesgo de fracaso del mismo es demasiado bajo.

Análisis Costo-Beneficio del proyecto.

Una vez desarrollada la técnica Planificación de Casos de Uso se procede a llevar a cabo el cálculo de los costos.

CH: Cantidad de hombres.

TTP: Tiempo total del proyecto.

SM: Salario mensual.

TEF: Total de esfuerzo.

CP: Costo del proyecto.

En caso de ser estudiante:

Asumiendo que el salario básico de un estudiante es de \$ 100.00 mensual (estipendio).

CH = 4 hombres

SM = \$100

TEF = 5

CP = SM * CH * TEF

CP = 100 * 4 * 5

CP = \$2000

TTP = TEF/CH

TTP = 5/4

TTP = 1, 2

TTP = 1mes

Se estima que el proyecto tenga un costo total de \$2000, trabajando 4 estudiantes, durante 1mes.

En caso de ser profesor:

Asumiendo que el salario básico de un profesor es de \$ 400.00 mensual.

CH = 4 personas

SM = \$400

$$\text{TEF} = 5$$

$$\text{CP} = \text{SM} * \text{CH} * \text{TEF}$$

$$\text{CP} = 400 * 4 * 5$$

$$\text{CP} = \$8000$$

$$\text{TTP} = \text{TEF}/\text{CH}$$

$$\text{TTP} = 5/4$$

$$\text{TTP} = 1.2$$

$$\text{TTP} = 1\text{mes}$$

Se estima que el proyecto tenga un costo total de \$8000, trabajando 4 profesores, durante 1mes.

Por lo que:

Producto a las características propias de la Universidad de las Ciencias Informáticas de desarrollar proyectos utilizando herramientas de software libre, los gastos en cuanto a las licencias no son necesarios. Además, la universidad posee los recursos tecnológicos para la puesta en marcha del desarrollo del sistema, así como el personal calificado. Es por ello que luego de realizar los cálculos y el análisis pertinente de los costos del proyecto, se llega a la conclusión de que la implementación del Sistema de Información Geográfica destinado para la representación de los centros universitarios del país es factible.

4.2 Validación de Requisitos.

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos de software se han establecido de manera precisa, que se han detectado las inconsistencias, omisiones y errores y que estos han sido corregidos y que además los productos de trabajo cumplen con los estándares establecidos para el proceso, proyecto y producto. La validación de requisitos trata de mostrar que estos definen realmente el sistema que desea el cliente.

4.2.1 Técnicas de validación de requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Lowe & Hall, 1999). Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos.

Existen algunas técnicas que pueden aplicarse para ello:

- **Reviews o Walk-throughs:** Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.
- **Listas de chequeo:** Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados. Las Listas de Chequeo están basadas en la identificación de las técnicas de Prueba para evaluar cada Subcaracterística de las Características de Calidad. Una lista de chequeo es un formulario de preguntas, las cuales dependen del objetivo para el cual son usadas. Estas listas están clasificadas según las etapas del Proceso de Prueba. Para dar respuesta a cada pregunta se considera una escala del 1 al 5, en donde el uno (1) siempre es la respuesta menos significativa y cinco (5) la más significativa. (Grimán, Pérez, E Mendoza)
- **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist (**Listas de Chequeo**) predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.
- **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (Olsina, 1999).
 - Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución.

- Un prototipo puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.
- Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.
- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (Durán, Bernáldez, Ruíz & Toro, 1999). Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

La validación de requisitos es una actividad muy importante, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo. Después de un estudio de las técnicas de validación anteriormente mencionadas se escogieron las matrices de trazabilidad y las listas de chequeo ya que a través de las mismas se identifican y se les da seguimiento desde sus orígenes con mayor facilidad a los requisitos además se detectan las no conformidades de estos y de los casos de usos, así como las del diagrama de casos de uso del sistema, para que posteriormente estas inconsistencias sean corregidas

A continuación se muestran como fueron aplicadas.

Matriz de trazabilidad.

La trazabilidad de requisitos se define como la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, a través de todas las especificaciones generadas durante el proceso de desarrollo de software. La trazabilidad de requisitos es clave para conseguir una exitosa gestión de los mismos.

Para facilitar el trabajo de determinar las relaciones entre los requisitos y el seguimiento de los mismos, se utilizan matrices de trazabilidad. Cada matriz de trazabilidad identifica los requisitos relacionados con uno o más aspectos del sistema o su entorno.

Capítulo 4: Estudio de factibilidad y Validación del sistema

RF/CU	CU-1	CU-2	CU-3	CU-4	CU-5	CU-6	CU-7	CU-8	CU-9	CU-10
RF1	X									
RF2	X									
RF3	X									
RF4	X									
RF5							X			
RF6							X			
RF7							X			
RF8							X			
RF9								X		
RF10									X	
RF11									X	
RF12										X
RF13				X						
RF14				X						
RF15				X						
RF16		X								
RF17			X							
RF18			X							
RF19					X					
RF20						X				

Figura 8. Matriz de trazabilidad

Listas de Chequeo

Se aplicaron las listas de chequeo (Ver Anexo 1 y 2), se identificaron no conformidades (Ver Anexo 3) que fueron resueltas.

Conclusiones Parciales.

En este capítulo, la realización de la estimación de tiempo, esfuerzo y costo del proyecto para el análisis de factibilidad del mismo, mediante la aplicación de las técnicas de validación correspondientes, arrojó como resultado una respuesta positiva, determinando como conclusión que se podrá llevar a cabo la implementación del SIG, en un tiempo de desarrollo relativamente corto, con un grupo de desarrollo pequeño, que se encuentra en condiciones de asumir el reto gracias a la preparación que tienen sus miembros en el desarrollo de este tipo de tecnología. A su vez, la realización de este producto trae como beneficio que una vez concluido, el ahorro de los gastos económicos del mismo sea el máximo.

Por su parte, la validación de los requisitos, permitió asegurar que los mismos se hayan establecido de manera precisa y se hayan detectado los errores u omisiones que pudieran existir, para garantizar que las expectativas del cliente sean cumplidas

CONCLUSIONES

Una vez concluida la presente investigación en la cual se realizó el análisis de un SIG para la representación de los centros universitarios del país, se puede arribar a las siguientes conclusiones:

- La caracterización del proceso de desarrollo de los Sistemas de Información Geográfica permitió que se tuvieran los elementos fundamentales a tener en cuenta para la construcción de un sistema de este tipo. También que se conociera la complejidad o envergadura que supone llevar a cabo un SIG y a su vez que se concluyera que no existe ningún sistema a nivel nacional o internacional donde se represente la información socioeconómica de todas las universidades de un país. Por su parte, contribuyó a identificar las funcionalidades comunes que deben tener los SIG y algunas de interés particular para una u otra universidad.
- La herramienta, metodología y lenguaje de modelado identificado a partir de la investigación de las tendencias y tecnologías actuales para el desarrollo de software garantizó la realización del correcto modelado de un conjunto de artefactos necesarios para el posterior diseño e implementación de la aplicación.
- La estimación de tiempo, costo y recurso, permitió tener una visión completa de la gran factibilidad que tendrá la realización del producto que se analiza, trayendo como beneficio que el desarrollo del mismo sea exitoso.

Por todo lo antes mencionado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

RECOMENDACIONES

De acuerdo al trabajo realizado, la importancia conferida a la realización de un SIG para la representación de los centros universitarios del país y su información socioeconómica, y teniendo en cuenta los resultados o beneficios que proporciona esta investigación, se recomienda:

- Identificar y adicionar nuevas funcionalidades al sistema teniendo en cuenta las necesidades del cliente.
- Adicionarle nuevos flujos de actividades a la funcionalidad: Crear Mapa Temático.
- Concebir la documentación ingenieril correspondiente a la fase de elaboración, construcción y transición.
- Analizar y aplicar consecuentemente el resultado de esta investigación en pos de lograr el correcto diseño e implementación de la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

1. **Guerra, Y. F. (2009).** Diseño de la plataforma soberana para el desarrollo de Sistemas de Información Geográfica del Polo Geoinformática. Ciudad de La Habana.
2. **Bosque Sendra, J. (1992)** Sistemas de Información Geográfica. Rialp. Madrid.
3. **Fornaris, M. S. (2009).** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. Ciudad de la Habana.
4. **Ortiz, Gabriel (2007).** ¿Qué son los Sistemas de Información Geográfica? Tipos de SIG y modelos de datos. Un artículo introductorio para entender las bases de los SIG. Disponible en: <http://www.gabrielortiz.com/index.asp?Info=012>.
5. **Medina Martínez, Juan Carlos. 2004.** Análisis Comparativo de Técnicas, Metodologías y Herramientas de Ingeniería de Requisitos. D.F. México: s.n., 2004. Disponible en <http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2004/>
6. **YAGÜEZ, J. C. D. y LANGHI, R. 2002.** Sistema de Información Geográfica (S.I.G). 2002. Disponible en: http://www.inta.gov.ar/barrow/info/documentos/SIG/que_es_sig.htm.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Proceso Unificado de Desarrollo de Software. Madrid: Series Editors, 2000.
8. **MARTÍNEZ, E. A. 2005.** Metodología. Hágase la luz. 2005. Disponible en: <http://www.crisol.cc/eamftec/metodologias/Metodologias.pps>
9. **RODRÍGUEZ, N. y SANABRIA, R. D. M. 1998.** Los Sistemas de Información Geográfica.1998. Disponible en: <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001#top>.
10. **NCGIA:** Centro Nacional de Análisis de Sistemas de Información Geográfica en EE.UU.
11. **Visual Paradigm. Visual Paradigm for UML - UML tool for software application development.** [En línea] January 11, 2010. [Citado el: 19 de diciembre de 2010.] <http://www.visual-paradigm.com/product/vpum/>.
12. **Herrera Valencia, Joan Ignasi, Pereira Cao, Mercè y Insua de la Fuente, Carlos. 2008.** Infraestructuras de datos para la localización y acceso inteligente a la información disponible a través de servicios web de catálogo. 2008.
13. **Zuluaga Giraldo, Carlos Alexander. 2008.** Enterprise Architect y UML. 2008.

14. **Molpeceres, Alberto. 2002.** Procesos de desarrollo: RUP, XP y FDD. 2002. Disponible en <http://www.javaHispano.org>
15. Pressman, Roger S. 2005. Ingeniería de Software. Un enfoque práctico. Sexta Edición. 2005.
16. **Comas, David y Ruiz, Ernests. 1993.** Fundamentos de los Sistemas de Información Geográfica. s.l.: Ariel Geografía, 1993.
17. **Babylon, Diccionario.** Diccionario Babylon.
<http://diccionario.babylon.com/Georreferenciaci%C3%B3n>.
18. **Lowe, D., Hall, W. (1999).** Hypermedia and the Web. An Engineering approach. John Wiley & Son.
19. **Olsina, L. (1999).** Metodología cualitativa para la evaluación y comparación de la calidad de sitios web. Ph. Tesis. Facultad de Ciencias Exactas. Universidad de la Pampa. Argentina.
20. **Durán A, Bernárdez, B., Ruiz, A, Toro M.** (1999). A Requirements Elicitation Approach Based in Templates and Patterns. Workshop the Requirements Engineering. Buenos Aires, Argentina.
21. **Anna. C Grimán, María Pérez, Luis. E Mendoza:** http://www.lisi.usb.ve/publicaciones/03/evaluación/evaluación_15.pdf.
22. **Murillo Alfaro, Félix. 1999.** Herramientas Case. 1999.

BIBLIOGRAFÍA

1. **Chacón, Julio César Rueda.** Aplicación de la Metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE. Guatemala: Universidad de San Carlos de Guatemala. Facultad de Ingeniería. Escuela de Ingeniería en Ciencias y Sistemas, **2006**.
2. Popkin Software and Systems. Modelado de Sistemas con UML. 2008.
3. **Orallo, Enrique Hernández.** El Lenguaje Unificado de Modelado (UML). 2007.
4. Enciclopedia Libre Universal en Español. Lenguaje de programación. [En línea] [Citado el: 18 de diciembre de 2010.]
5. http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n.
6. **Ochoa, Jesús Alberto.** VII Semestre Ingeniería de Sistemas. Ingeniería de Software II. [En línea] <http://members.fortunecity.com/software1/herramie.htm>.
7. **González Blanco, Rubén y Pérez Tobalina, Sergio.** Introducción a Rational Rose. s.l.: Facultad de Informática de Barcelona. Departamento de Lenguajes y Sistemas Informáticos. Laboratorio de Ingeniería de Software.
8. Visual Paradigm. Visual Paradigm for UML - UML tool for software application development. [En línea] January 11, 2010. [Citado el: 19 de diciembre de 2010.]
<http://www.visual-paradigm.com/product/vpuml/>.
9. **Gracia, Joaquín.** Ingeniero Software. Patrones de diseño, Diseño de Software Orientado a Objetos. [En línea] 27 de Mayo de 2005.
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
10. Gestión de la Información. http://www.logisticaonline.org/programas/Curso_Informacion.pdf
11. **MARTÍNEZ, E. A. 2005.** Metodología. Hágase la luz. 2005. Disponible en:
<http://www.crisol.cc/eamftec/metodologias/Metodologias.pps>
12. eXtreme Programming (XP). **2007.** Disponible en:
<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
13. **Jiménez Garzón, Darwin. 2007.** Ingeniería de Software II, RUP. 2007
14. **Baeza Pablo ,Nicolás. 2008.** Visual Paradigm, DB Visual ARCHITECT SQL. 2008.
15. Cesar Rodríguez, Hendrik. 2008. Guía metodológica para el desarrollo de Sistemas de Información Geográfica en la Universidad de las Ciencias Informáticas. Ciudad de la Habana : , 2008.

16. **Ibarra, Armando F. 2007.** Rational Unified Process. 2007.
17. **LORA YEPES, EDDIE. 2005.** Sistemas de Información Geográfica. Aplicación en Hidráulica de Ríos y Costas. 2005.
18. **Martínez, Alejandro y Martínez, Raúl. 2008.** Guía a Rational Unified Process. 2008.
19. **Mendoza Sánchez, María A. 2004.** Metodologías.
20. Metodologías de Desarrollo. **2007.** 2007. Disponible en:
<http://www.um.es/docencia/barzana/IGP/lagp2.html#BM1>
21. **Molpeceres, Alberto. 2002.** Procesos de desarrollo: RUP, XP y FDD. 2002. Disponible en:
<http://www.javaHispano.org>
22. **Murillo Alfaro, Félix. 1999.** Herramientas Case. 1999.
23. **Ponce Alcalde, Emilio. 2008.** Una herramienta SIG para el análisis socioeconómico de la Región Metropolitana de Barcelona. 2008.
24. Plataformas GIS, quiénes toman ventaja? 2007. Disponible en:
http://blogeame.net/gis/entry/plataformas_gis_quienes_toman_ventaja
25. **RODRÍGUEZ, N. y SANABRIA, R. D. M. 1998.** Los Sistemas de Información Geográfica. 1998. Disponible en:
<http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001#top>.
26. **SANCHEZ, M. A. M. 2004.** Metodologías de Desarrollo de Software. 2004. Disponible en:
http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.
27. **Sanz Salinas, Jorge Gaspar. 2007.** Sistemas de Información Geográfica, Software Libre e Infraestructuras de Datos Espaciales. 2007. Disponible en: <http://www.geomaticblog.net>
28. **Tinoco Guevara, Roberto. 2007.** Definición y algunas aplicaciones de los sistemas de información geográfica. 2007. Disponible en:
<http://www.monografias.com/trabajos14/informageogra/informageogra.shtml>
29. **Zuluaga Giraldo, Carlos Alexander. 2008.** Enterprise Architect y UML. 2008.
30. <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
31. <http://www.monografias.com/trabajos5/desof/desof.shtml#funda>.
32. http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo
33. Metodologías Ágiles en el Desarrollo de Software. 2003. Disponible en:
<http://issi.dsic.upv.es/tallerma>

34. **Gómez Francisco Javier** (1992). Los Sistemas de Información Geográfica. Su importancia y su utilidad en los estudios medioambientales.
35. **Herrera Valencia, Joan Ignasi, Pereira Cao, Mercè y Insua de la Fuente, Carlos.** 2008. Infraestructuras de datos para la localización y acceso inteligente a la información disponible a través de servicios web de catálogo. 2008.
36. **NCGIA:** Centro Nacional de Análisis de Sistemas de Información Geográfica en EE.UU.
37. **Ortiz, Gabriel (2007).** ¿Qué son los Sistemas de Información Geográfica? Tipos de SIG y modelos de datos. Un artículo introductorio para entender para las bases se los SIG.
38. <http://www.gabrielortiz.com/index.asp?Info=012>.
39. **Pressman, Roger S.** 2005. Ingeniería de Software. Un enfoque práctico. Sexta Edición.
40. **Pressman, Roger S.** 2005 Libro Texto “Ingeniería de Software. Un enfoque práctico”, Quinta Edición.