



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6

*TÍTULO: Análisis y Diseño del Sistema de Información
Geográfica para la Universidad de Ciencias Informáticas basado en
la Plataforma gvSIG Mini.*

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

Autora: Viviana Hinojosa Carrión.

Tutor: Lic. Mario Liosbel Díaz Abreu

Co-Tutor: Ing. Odriel Estrada Molina

<La Habana>, <Junio, 2011>.

“Año 53 de la Revolución”



... aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos...

Che

Declaración de Autoría

Declaro que soy la única autora del trabajo titulado: Análisis y Diseño del Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas para celulares basado en la Plataforma gvSIG Mini.

Y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Viviana Hinojosa Carrión

Lic. Mario Liosbel Díaz Abreu

Firma del Autor

Firma del Tutor

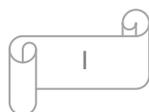
Agradecimientos

Lo que importa verdaderamente en la vida no son los objetivos que nos enmarcamos, sino los caminos que seguimos para lograrlo''. A lo largo de estos 5 años he transitado por un camino lleno de adversidades, pero también lleno de amor, de cariño de amistad, en fin de momentos felices al lado de todas las personas que han hecho posible mi sueño de hacerme ingeniera en la carrera de Ciencias Informáticas y son a estas personas a las que les quiero dar las gracias.

Agradezco de todo corazón: A **mi padre** que aunque no ha estado conmigo todo el tiempo se sentirá orgulloso de que su hija haya logrado ser ingeniera. A **mi padrastro** Juan Carlos que me ha enseñado muchas cosas de la vida, que me ha apoyado en todo y que siempre se ha preocupado por mi bienestar.

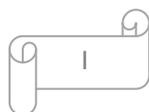
A **todos los chicos** que he tenido la bendición de conocer y compartir con ellos mi corazón durante todo este tiempo en la UCI, esos que siempre han estado en todo momento ofreciéndome su apoyo incondicional y todo su amor: a (Iván, Yani, Lilo, Yessy, Milo, Yosnai, Luis, Misbel, a la Profe Yari, al Profe Dieter), en fin a todos ellos, saben que en mi corazón siempre tienen su lugarcito. A los **profes del Proyecto**, a **Lisandra** que tanto la molesté, a **mi Tutor Mario** por siempre estar arriba de mi inculcándome preocupación, a **Odiel** que como co-tutor fue excelente nunca tuvo reparos en ayudarme en fin a todos por su paciencia, esmero, por su ayuda incondicional. A los **miembros del tribunal** por su exigencia y apoyo.

Igual agradezco a esos que un día fueron parte de mis vivencias aquí en la Universidad y hoy ya no son, de todos he aprendido un poquito para ser mejor cada día. Agradezco eternamente a **nuestra Revolución** por ofrecerme esta oportunidad de forjarme y prepararme como toda una profesional. A nuestro comandante en Jefe, único e incomparable: Fidel Castro Ruz por esta idea extraordinaria al hacer realidad los sueños e ideas de Martí de una sociedad donde todos somos iguales.



Dedicatoria

Dedico el resultado de esta investigación, con mucho amor: A mi mamita linda por ser mi esperanza, fuente de inspiración, la luz que siempre me ilumina y guía por el camino correcto, por ser tan genial, porque sin su amor y apoyo nunca hubiera logrado hacer realidad este gran sueño que parecía inalcanzable, por ser más que mi vida: mi todo. A mis abuelitos que son el tesoro más grande que me ha podido dar la vida, porque siempre han estado conmigo cuando necesité derramar lágrimas tanto de alegría como de dolor, y sé que están orgullosos de su nietecita, por reunir todas las fuerzas del mundo y ser así como ustedes, decidida y con carácter optimista. Mi abuelita Isabel que es mi segunda madre, la viejita más dulce que se pueda conocer, mi ángel guardián, ella que me ha enseñado tanto, me ha enseñado a querer a quien te quiere, a respetar a los demás, a llorar solo cuando es necesario, gracias por ser tan especial. A los tíos más espectaculares de la Tierra por haberme querido siempre y haberme apoyado y ayudado como a una hija en todos estos años de universidad: Mi tiito Iván y Yoyi, por siempre brindarme todo su cariño y su corazón. A mi hermanita de toda la vida Maidolis, no hay nada mejor que tener una amiga tan especial, le doy las gracias por darme siempre fuerzas para continuar y brindarme todo el amor del mundo, por soportar siempre a esta chica tan malcriada, por ser incondicional. Por estar en las buenas y malas, por apoyarme aun cuando todo parecía desvanecerse, por demostrarme que aun cuando tienes muchas razones para no seguir, debes hacerlo porque tienes muchas más por las cuales si hacer las cosas, por haberme ayudado a ser mejor persona, mejor amiga, por enseñarme que cuando se quiere se puede cambiar, por su cariño y por su linda amistad. Te quiero inmensamente grande mi hermanita, siempre estarás en mi corazón. A mi otra hermanita Yula, que ha estado ahí siempre apoyándome, aconsejándome y guiándome por el camino correcto.

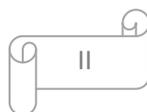


Resumen

El surgimiento y desarrollo de los Sistemas de Información Geográfica (SIG) ha traído consigo numerosas ventajas, logrando marcar una notable diferencia en el mundo de la Informática y las Telecomunicaciones. Actualmente se han desarrollado sistemas de este tipo para ordenadores y también para dispositivos móviles, pues un SIG que puede representarse en un móvil, puede ser consultado por el usuario cualquier día, hora y lugar en el que se encuentre.

El documento realizado refiere un conjunto de resultados obtenidos durante el desarrollo de la investigación y brinda argumentos que justifican la fundamentación de la solución propuesta, a la situación problémica existente. También deja plasmadas todas las funcionalidades que brindara este producto informático, modelándolas según su descripción y análisis con un alto nivel de detalle. Se emplea como Lenguaje de Modelado (sus siglas en ingles UML), como metodología de desarrollo Proceso Unificado de Desarrollo (sus siglas en ingles RUP), y la herramienta Visual Paradigm para realizar los diagramas correspondientes al sistema, además de proporcionar toda la documentación del software. Por último, se muestra los diagramas de diseño de la aplicación así como un grupo de conclusiones y recomendaciones de enorme valía para la continuidad del trabajo.

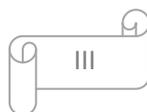
Palabras Clave: SIG.



Abstract

The emergence and development of the Geographic Information Systems (GIS) has resulted in several advantages in many branches, thus achieving a notable difference in the world of the Computing and Telecommunications. Systems of this type have been currently developed for computers and also for mobile devices, since a GIS that can be represented in a mobile, can be consulted by the user at any time, day and place where the person is.

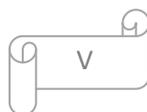
The document made shows a set of results obtained during the development of the investigation and provides arguments that justify the base of the proposed solution to the existing problem. It also shows all the functionalities that this computing product will provide, modeling them according to its description and analysis with a high detail level. It is used as Modeling Language (UML its abbreviations in English), as a methodology of development Unified Process of Development (RUP its English abbreviations) and the tool Visual Paradigm to make the corresponding diagrams in the system, also providing all the software documentation. Finally, it shows the application design diagrams, as well as a group of conclusions and recommendations of high importance for the continuity of the work.



Índice

<i>Introducción</i>	1
<i>Capítulo 1. “Fundamentación Teórica”</i>	7
1.1 Introducción	7
1.2 Conceptos asociados al dominio del problema	7
1.3 Objeto de Estudio	9
1.4 Descripción General.....	9
1.5 Funcionalidades de los SIG	10
1.6 Componentes de un Sistema de Información Geográfica	10
1.7 Funciones de los Sistemas de Información Geográfica	11
1.8 Análisis de otras soluciones existentes.....	12
1.9 Desarrollo de los SIG en Cuba.....	14
1.10 Conclusiones Parciales	16
<i>Capítulo 2. “Tendencias y Tecnologías Actuales de Desarrollo”</i>	17
2.1 Introducción	17
2.2 Metodologías de Desarrollo de Software.....	17
2.3 Lenguaje Unificad de Modelado – UML	20
2.4 Herramientas y Tecnologías.....	21
2.5 Herramientas CASE	22
2.6 Entorno de Desarrollo Integrado (IDE).....	24
2.7 Lenguaje de Programación.....	25
2.7.1 Java	25
2.8 Conclusiones Parciales	26
<i>Capítulo 3. “Presentación de la Solución Propuesta”</i>	28
3.1 Introducción	28
3.2 Modelo de Dominio.....	28
3.3 Diagrama de Clases del Dominio	29
3.4 Descripción del Diagrama de Clases del Dominio	29
3.5 Glosario de Términos del Dominio	29
3.6 Captura de Requisitos	30

3.7	Requisitos Funcionales del Sistema.....	31
3.8	Requisitos No Funcionales del Sistema	32
3.9	Descripción del Sistema Propuesto.....	33
3.9.1	Descripción de los Actores.....	33
3.10	Diagrama de Caso de Uso del Sistema.....	33
3.11	Descripción Textual de los Casos de Uso.....	34
3.11.1	Descripción Textual del Caso de Uso Localizar_Edificio.	35
3.12	Conclusiones Parciales	37
<i>Capítulo 4. “Construcción de la Solución Propuesta”</i>		<i>38</i>
4.1	Introducción.....	38
4.2	Modelo del análisis.....	38
4.3	Diagramas de Clases del Análisis	38
4.4	Diagramas de Interacción	40
4.5	Modelo de Diseño.....	40
4.5.1	Diagramas de Clases del Diseño.....	41
4.6	Arquitectura definida para el Sistema	42
4.7	Modelo de Despliegue	44
4.8	Validación de los Requisitos.....	44
4.9	Estudio de la validación por fases	45
4.10	Métrica para la Calidad de la Especificación de los Requisitos de Software..	50
4.11	Conclusiones.....	55
<i>Conclusiones Generales.....</i>		<i>56</i>
<i>Recomendaciones</i>		<i>57</i>
<i>Anexos</i>		<i>58</i>
<i>Bibliografía.....</i>		<i>68</i>
<i>Glosario de Términos</i>		<i>70</i>



Índice de Tablas y Figuras

Figura 1 Métodos Científicos	5
Figura 2 Descripción de los Sistemas de Información Geográfica.....	8
Figura 3 Diagrama de Clases del Domino.....	29
Figura 4 Diagrama de Casos de Uso del Sistema.....	34
Figura 5 Diagrama de Clases del Análisis CU Localizar Edificio	39
Figura 6 Diagrama de Colaboración CU Localizar Edificio	40
Figura 7 Diagrama de Clases del Diseño CU Localizar_Edificio	41
Figura 8 Representación Gráfica de la Arquitectura 3 Capas.....	43
Figura 9 Modelo de Despliegue del Sistema	44
Figura 10 Diagrama de Clases del Análisis Localizar_Persona	62
Figura 11 Diagrama de Clases del Análisis Mostrar_Información_Elemento	62
Figura 12 Diagrama de Clases del Análisis Exportar_Mapa.....	63
Figura 13 Diagrama de Colaboración del Análisis Localizar_Persona	63
Figura 14 Diagrama de Colaboración del Análisis Mostar_Información_Elemento.....	64
Figura 15 Diagrama de Colaboración del Análisis Exportar_Mapa.....	64
Figura 16 Diagrama de Clases del Diseño Mostrar_Información_Elemento.....	65
Tabla 1 Descripción Textual del Caso de Uso Localizar_Edificio.	37
Tabla 2 Ejemplo de elementos definidos en la lista de chequeo Especificación de Requisitos.....	49
Tabla 3 Matriz de Trazabilidad	52
Tabla 4 Aplicación de métricas al Diagrama de Casos de Uso del Sistema.	54
Tabla 6 Descripción Textual CU Localizar Persona	60
Tabla 7 Descripción Textual CU Mostrar Información Elemento	62

Introducción

En el mundo actual los cambios son constantes y suceden a la misma velocidad a la que puede ser compartida la información. La revolución digital de finales del siglo pasado permitió que la información geográfica se volviera más accesible para la mayoría de las personas utilizando sistemas que permiten este proceso de georreferenciación.

Un Sistema de Información Geográfica (SIG) es una integración organizada de hardware, software y datos geográficos, diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas, la información geográficamente referenciada, con el fin de solucionar dificultades complejas relacionadas con los procesos de planificación y gestión. Igualmente, se definen por los especialistas como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, construido para satisfacer necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. (1)

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 presentar los resultados de todas estas operaciones. Este funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común con los objetos gráficos de un mapa digital. De esta forma, señalando un objeto, se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos, se puede saber su localización en la cartografía.

La utilización de Sistemas de Información Geográfica (SIG) permite a personas y organizaciones la toma de decisiones utilizando información proveniente de un rango amplio de disciplinas. Los SIG permiten relacionar información de cualquier tipo con una localización geográfica. Los ejemplos de utilización de estos son virtualmente ilimitados, así como lo es el tipo y cantidad de información que se puede asociar con lugares particulares de la tierra por citar algún ejemplo. Actualmente el mundo se encuentra ante una nueva realidad, donde cada vez se incorporan nuevas tecnologías, y en estas a su vez se vuelve más necesaria la utilización de estándares y la independencia tecnológica como aspecto fundamental.

En la actualidad los Sistemas de Información Geográficas son herramientas utilizadas en muchos ámbitos, debido a que permite obtener cartografía¹ temática sobre cualquier aspecto ambiental y socioeconómico de la superficie terrestre. Lo que diferencia a un SIG de otros sistemas de información es que se dispone de información geográfica (espacial y temática), y se trabaja con cartografía y una base de datos al mismo tiempo, configurando de esta manera una base de datos geográfica².

En Cuba se desarrolla su utilización con el objetivo de facilitar la gestión de la estadística de salud. Los especialistas confeccionan una aplicación SIG, que permite cartografiar y realizar diferentes tipos de análisis con respecto a diversos aspectos de la salud como: morbilidad, mortalidad, población, recursos y servicios; el diseño de la base de datos, creada para este sistema, puede emplearse en otros sistemas de información geográfica, utilizados para desarrollar diversos análisis epidemiológicos como el Sistema de Información Geográfica de Epidemiología (SIG-EPI).

Actualmente la Universidad de Ciencias Informáticas (UCI) tiene diversos centros y departamentos que se encargan de guiar el trabajo productivo en diferentes proyectos. El departamento Geoinformática del centro GeySED de la facultad 6, cuenta con el proyecto SIG-Móviles, donde se desarrollan sistemas de información geográfica para dispositivos móviles.

La UCI presenta una aplicación WEB que brinda a la comunidad universitaria un sistema capaz de ofrecer y analizar información geográfica detallada y actualizada sobre la universidad. Desde cualquier ordenador de la universidad se puede acceder a dicho sistema.

Teniendo en cuenta que muchas veces existen factores que impiden un mejor uso de las tecnologías ,como es la falta de fluido eléctrico ,la conexión inhabilitada , el número de computadoras que no abastecen las verdaderas necesidades de la comunidad universitaria es que el proyecto SIG-Móviles se ve en la necesidad de crear una aplicación de escritorio para celulares, donde todos los usuarios que cuenten con este dispositivo puedan servirse de toda la información que necesiten, sin correr el riesgo

¹ Técnica que permite la realización de Mapas.

² Modelo que representa las características del espacio geográfico en una base de datos.

que ocurran las deficiencias mencionadas anteriormente, además contando con la posibilidad de que pueden conectarse desde cualquier lugar y a la hora que decidan, sin tener que dirigirse a un ordenador.

De este modo se integran los dispositivos móviles con las tecnologías, dando como resultado un software con prestaciones específicas de lo que en realidad el usuario necesita.

Partiendo de la situación problemática anteriormente mencionada se plantea el siguiente **problema a resolver**: ¿Cómo contribuir mediante un análisis y diseño correcto a resolver la inaccesibilidad existente en el SIG de la Universidad de las Ciencias Informáticas, para que todos los usuarios de la comunidad universitaria puedan acceder a la información georeferenciada de la institución, desde cualquier punto de referencia?

El **objeto de estudio** de esta investigación se encuentra enmarcado en: Sistema de Información Geográfica para celulares teniendo como **campo de acción**: Análisis y Diseño de los Sistemas de Información Geográfica de escritorio para dispositivos celulares.

Por lo tanto se plantea el **Objetivo** de esta investigación: Desarrollar una propuesta de Análisis y Diseño del Sistema de Información Geográfica para la UCI basado en la plataforma gvSIG mini.

Idea a defender: Si se diseña correctamente todos los artefactos de software correspondiente al diseño del Sistema de Información Geográfica para los usuarios de dispositivos celulares de la Universidad, entonces se obtendrán los principales artefactos necesarios para su posterior implementación.

Para darle cumplimiento a lo antes mencionado se debe tener en cuenta una serie de tareas que guiarán el desarrollo de esta investigación como son:

- ✚ Caracterizar las tendencias actuales, tecnologías y conceptos más importantes relacionados con Sistemas de información geográficas.
- ✚ Caracterizar las principales herramientas de código abierto y estándares que soportan la Plataforma gvSIG Mini.

- ✚ Seleccionar la metodología de desarrollo de software y herramientas que podrían utilizarse en el desarrollo de la aplicación.
- ✚ Diseñar la aplicación informática sobre la Plataforma gvSIG Mini utilizando para ello patrones de diseño.

Es por ello que se considera como posibles resultados:

- ✚ Obtener un diseño del sistema de información geográfica para celulares a la Universidad de las Ciencias Informáticas sobre la plataforma gvSIG Mini.
- ✚ Documentación generada por los artefactos de Ingeniería de software.

Métodos Científicos

Durante todo el proceso investigativo realizado se utilizaron un conjunto de métodos científicos (Figura # 1). Estos métodos se clasifican en:

- ✚ **Teóricos:** Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado.
- ✚ **Empíricos:** Estos métodos permiten extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de observaciones, del uso de técnicas opináticas y la propia experimentación.

Dentro de los teóricos se emplearon los siguientes:

- ✚ **Histórico-Lógico:** Se utilizará para el estudio de los procesos de gestión de información, trabajos e investigaciones anteriores, es decir para un mejor entendimiento y análisis del estado del arte lo que permitirá un eficaz desarrollo de la presente investigación.
- ✚ **Analítico-Sintético:** Se define con el objetivo de analizar los diversos documentos relacionados con el proceso de desarrollo de los Sistemas de Información Geográfica y las metodologías existentes para la construcción de los artefactos de ingeniería de software necesarios para futuros implementación de la aplicación.

- ✚ **Modelación:** Se emplea para obtener los diferentes diagramas y componentes que se construyen como resultado del proceso de Ingeniería de Software.

Dentro de los empíricos se empleó:

- ✚ **Observación:** Para realizar un registro visual de lo que ocurre en el entorno del problema para aportar nuevos elementos que puedan ser de interés científico y para observar cómo se realiza el proceso de desarrollo de los Sistemas de Información Geográfica con el objetivo de lograr un mejor entendimiento de esta aplicación.

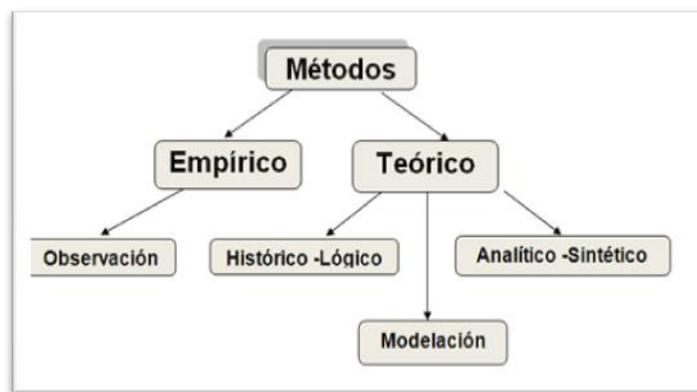


Figura 1 Métodos Científicos

Capítulos de la Investigación

El presente documento está estructurado por 4 capítulos, a continuación se expone brevemente una descripción de cada uno.

- ✚ En el primer capítulo titulado: “**Fundamentación teórica**”, se realiza un análisis a nivel mundial de los sistemas de información geográfica y son expuestos los principales conceptos y argumentos que esclarecen el objeto de estudio planteado.
- ✚ En el segundo capítulo “**Tendencias y Tecnologías actuales de desarrollo**” se explica el uso de las tecnologías, metodologías y herramientas que se utilizarán para el desarrollo de la aplicación.

- ✚ El tercer capítulo “**Presentación de la solución propuesta**”. Se profundizará en el Dominio del problema. Conjuntamente se levantarán los requisitos propios del sistema, tanto funcionales como no funcionales, realizando también las descripciones y la documentación correspondientes.
- ✚ El cuarto capítulo “**Análisis y Diseño de la Solución propuesta**”, permite mostrar los detalles descriptivos de la solución propuesta. Clases, métodos e interfaces, entre otros elementos del diseño de aplicaciones, se describen, de modo que se comprenda la estructura interna del software a desarrollar.

Capítulo 1. “Fundamentación Teórica”

1.1 Introducción

Este capítulo contiene las definiciones y conceptos que serán de utilidad para el entendimiento de la investigación. Se profundiza en el tema del desarrollo de los sistemas de información geográfica y su aplicación en Cuba, el análisis de soluciones existentes y se abordará con mayor profundidad la situación problemática que da origen a este trabajo.

1.2 Conceptos asociados al dominio del problema

Con el objetivo de lograr un mejor entendimiento de los temas que serán abordados en la investigación, se hace necesario enunciar a continuación varios de los conceptos asociados a éstos.

Sistemas de Información: Un sistema de información es un conjunto de procedimientos ordenados que, al ser ejecutados, proporcionan información para apoyar la toma de decisiones y el control de la institución. La información se define como una entidad tangible o intangible que permite reducir la incertidumbre acerca de algún estado o suceso. (2)

Hay tres actividades en un sistema de información que se toman en cuenta para la toma de decisiones, controlar operaciones, analizar problemas y crear nuevos productos o servicios. (Profesores, 2007)

Sistemas de Información Geográfica (SIG): Integración organizada de hardware, software y datos geográficos diseñado 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.(Figura # 2)(1)

Capítulo 1. “Fundamentación Teórica”



Figura 2 Descripción de los Sistemas de Información Geográfica.

Dispositivos celulares: Se le denomina celular debido a que el sistema de comunicación está basado en una división geográfica dentro de la cual se encuentran grupos (células), conformadas de varias antenas de telecomunicaciones. Anteriormente estos dispositivos solamente tenían la función de enviar y recibir llamadas telefónicas, mientras que en algunos casos tenían la función para envío de SMS ("Small Message System") o sistema de mensajes cortos. Sin embargo, actualmente integran una gran gama de funciones que los hace verdaderos dispositivos multifunciones y en gran medida también microcomputadoras. (3)

Mapa: Representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que por lo general suele ser plana, aunque también puede ser esférica como en el caso de los globos terráqueos. (Definición de mapa - Qué es, Significado y Concepto, 2009) (4)

Georreferenciación: Se basa en el posicionamiento en el que se concreta la localización de un objeto espacial (representado mediante punto, vector, área, volumen) en un sistema de coordenadas. La misma es muy utilizada en los Sistemas de Información Geográfica. (Diccionario Babylon, 2009). (5)

La georreferenciación posee una definición tecno científica, aplicada a la existencia de las cosas en un espacio físico, mediante el establecimiento de relaciones entre las imágenes de ráster o vector sobre una proyección geográfica o sistema de coordenadas.

Capítulo 1. “Fundamentación Teórica”

Datos Geográficos: Específicamente en geografía, los datos o antecedentes seleccionados para entender una idea, en este caso un fenómeno geográfico, y/o deducir su distribución y consecuencias, tienen dos características particulares: La primera de ellas consiste en hacer parte de una observación o soporte (unidad de observación, objeto geográfico, individuo) que tiene una posición espacial definida; y la segunda consiste en que dicho dato tiene la posibilidad de presentarse como una variable o atributo temático, “es decir que los objetos espaciales están dotados de propiedades intrínsecas las cuales se pueden medir”. (Bosque, 1992). (6)

Datos Espaciales: Un dato espacial puede entenderse como la representación de un objeto en dos o tres dimensiones, la cual tiene atributos inherentes al espacio; por lo que, por sí mismo, cuenta con los atributos de dimensión y de localización. (7)

Cartografía: La cartografía es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales. Además se puede definir como el conjunto de mapas producidos por una institución, relativos a un determinado territorio. Es la técnica geográfica que estudia la secuencia de etapas y procesos ejecutados para la visualización de un espacio geográfico mediante la producción de mapas, cartas, planos o croquis. (8)

1.3 Objeto de Estudio

Para el desarrollo de la investigación se definió como objeto de estudio “Sistema de Información Geográfica para celulares”.

Para lograr una aplicación SIG en dispositivos celulares, es necesario profundizar en el estudio de estos y su evolución a través de los años, así como profundizar en el estudio de los dispositivos móviles especialmente el celular.

1.4 Descripción General

En la actualidad el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC), han desencadenado un avance enorme en los Sistemas de Información Geográfica (SIG), los cuales han evolucionado colosalmente y sus aplicaciones se han dispersado por todas las ramas de la sociedad.

Las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que sólo pueden ser relacionadas geográficamente o mediante distribución espacial. Sólo la tecnología SIG permite almacenar y manipular

Capítulo 1. “Fundamentación Teórica”

información usando la representación geográfica para analizar patrones, relaciones y tendencias en la información, todo para contribuir a tomar mejores decisiones.

1.5 Funcionalidades de los SIG

- ✚ Permite realizar análisis vicariantes, es decir, permite realizar comparaciones entre escalas y perspectivas emulando una cierta capacidad de representación de diferentes lugares al mismo tiempo.
- ✚ Permite diferenciar entre cambios cualitativos y cuantitativos; aportando una gran capacidad de cálculo.
- ✚ Permite gestionar un gran volumen de información a diferentes escalas y proyecciones.
- ✚ Integra espacialmente datos tabulares y geográficos junto a cálculos sobre variables (topología).
- ✚ Admite multiplicidad de aplicaciones y desarrollos, poniendo a disposición herramientas informáticas estandarizadas.

1.6 Componentes de un Sistema de Información Geográfica

Un SIG integra cinco componentes principales: **Hardware, Software, Datos, Recursos Humanos y Métodos.**

Equipos: El hardware es el computador donde opera el SIG. Hoy en día los SIG se pueden ejecutar en una amplia variedad de plataformas que pueden variar desde servidores (computador central) a computadores desktop (escritorio) o Laptop (portátil) que se utilizan en las configuraciones de red o desconectado.

Programas: Los programas de los SIG proveen las funciones y las herramientas que se requieren para almacenar, analizar y desplegar información geográfica. Los componentes más importantes son:

- ✚ Herramientas para la entrada y manipulación de la información geográfica.
- ✚ Un sistema de administración de base de datos (DBMS).
- ✚ Herramientas que permitan búsquedas geográficas, análisis y visualización.

Capítulo 1. “Fundamentación Teórica”

- ✚ Interfaz gráfica para el usuario (GUI) para acceder fácilmente a las herramientas.

Datos: El componente más importante de un SIG son los datos. Los datos geográficos y tabulares relacionados pueden colectarse en la empresa, en terreno o bien adquirirlos a quien implementa el sistema de información, así como a terceros que ya los tienen disponibles. El SIG integra los datos espaciales con otros recursos de datos y puede incluso utilizar los administradores de base de datos (DBMS) más comunes para organizar, mantener y manejar los datos espaciales y toda la información geográfica.

Recursos Humanos: La tecnología SIG está limitada si no se cuenta con el personal adecuado que opere, desarrolle y administre el sistema, y llevar a cabo los planes de desarrollo para aplicarlos a los problemas del mundo real. Entre los usuarios de SIG se encuentran los especialistas técnicos, que diseñan y mantienen el sistema para aquellos que los utilizan diariamente en su trabajo.

Metodología y Procedimientos: Para que un SIG tenga éxito, este debe operar de acuerdo a un plan bien diseñado y estructurado y acorde con las reglas de la empresa o institución, que son los modelos y prácticas operativas características de cada organización. (9)

1.7 Funciones de los Sistemas de Información Geográfica

Un SIG almacena la información en capas temáticas que pueden enlazarse geográficamente. Este concepto simple pero altamente poderoso y versátil ha probado ser crítico en la resolución de muchos problemas que van desde el rastreo de vehículos de reparto, registrando los detalles de la aplicación de planificación hasta el modelamiento de la circulación atmosférica global.

Referencia Geográfica: La información geográfica contiene una referencia geográfica explícita tal como la latitud y la longitud, coordenadas nacionales o una referencia implícita tal como una dirección, un código postal, el nombre de un área censal, el identificador de un área boscosa o el nombre de un camino. Para crear referencias geográficas explícitas (múltiples ubicaciones) a partir de referencias implícitas (descripciones tales como direcciones) se utiliza un proceso automatizado llamado geocodificación. Estas referencias geográficas permiten localizar elementos en la

Capítulo 1. “Fundamentación Teórica”

superficie terrestre para ser analizados, por ejemplo, el stand de un negocio, bosques o eventos tales como un terremoto. (10)

¿Cómo funciona un SIG?

Los sistemas de información geográfica trabajan con dos tipos fundamentales de modelos geográficos, el “**modelo vector**” y el “**modelo ráster**”.

En el **modelo vector**, la información de puntos, líneas y polígonos es codificada y almacenada como una colección de coordenadas (x, y). La ubicación de un elemento punto puede ser descrito como una simple coordenada (x, y). Los elementos lineales tales como caminos y ríos pueden ser almacenados como una colección de coordenadas de puntos. Los elementos poligonales, por ejemplo territorios de venta o cuencas de ríos, pueden guardarse como bucles cerrados de las coordenadas. El modelo vectorial es extremadamente útil para describir elementos discretos, pero no lo es tanto para describir elementos de variabilidad continua como los tipos de suelos o el costo de accesibilidad de un hospital.

El **modelo ráster** ha evolucionado para modelar elementos continuos. Una imagen ráster está integrada por una colección de celdas (grillas) más bien como una cartografía o foto escaneada. Tanto los modelos ráster como los vectoriales poseen ventajas y desventajas propias para el almacenamiento de datos geográficos. Los SIG modernos tienen la capacidad de manejar ambos modelos. (11)

1.8 Análisis de otras soluciones existentes

Los Sistemas de Información Geográfica se han convertido, gracias al desarrollo de los medios informáticos, en una potente herramienta de soporte a la gestión y conocimiento del territorio insospechado, existe una gran variedad de SIG, pero todos con arquitecturas y funcionalidades diferentes, todo esto se debe a la evolución y avance de los mismos en todos estos años.

Ejemplo:

- 🌐 **SIG Web:** Los cuales permiten la visualización de datos y acceder a funcionalidades de análisis y consulta de servidores SIG a través de internet o intranet.

Capítulo 1. “Fundamentación Teórica”

Generalmente se distinguen por ser un cliente ligero, (por ejemplo, un navegador web para visualizar mapas de Google) sólo proporcionan una funcionalidad de visualización y consulta.

En la actualidad se pueden encontrar como principales SIG a nivel mundial:

- ✚ **ArcGIS:** Desarrollado por ESRI (Environmental Systems Research Institute) España Geosistemas S.A.

En este se concentran varias aplicaciones para la captura, edición, análisis, tratamiento, diseño, publicación e impresión de información geográfica. Estas aplicaciones se engloban en familias temáticas como:

- ✚ ArcGIS Server, para la publicación y gestión web.
- ✚ ArcGIS Móvil para la captura y gestión de información en campo.
- ✚ ArcGIS Desktop, la familia de aplicaciones SIG de escritorio, es una de las más ampliamente utilizadas, incluyendo en sus últimas ediciones las herramientas ArcReader, ArcMap, ArcCatalog, ArcToolbox, ArcScene y ArcGlobe, además de diversas extensiones. ArcGIS Desktop se distribuye comercialmente bajo tres niveles de licencias que son, en orden creciente de funcionalidades y coste: ArcView, ArcEditor y ArcInfo. (ESRI, 1991) (12)
- ✚ **GvSIG:** Sistema de Información Geográfica en Software Libre de la Generalitat Valenciana.

Es un proyecto de desarrollo de Sistema de Información Geográfica en software libre, que incluye principalmente las aplicaciones GvSIG Desktop y GvSIG Mobile.

Es una herramienta orientada al manejo de información geográfica. Se caracteriza por una interfaz amigable, siendo capaz de acceder a los formatos más usuales de forma ágil tanto ráster como vectoriales. Integra en una vista datos tanto locales como remotos.

GvSIG Desktop fue la primera aplicación que se desarrolló dentro del proyecto GvSIG, por lo que se conoce también como GvSIG. (13)

- ✚ **MapInfo:** Distribuidor, Desktopmapping Software.

Capítulo 1. “Fundamentación Teórica”

MapInfo está basado en la cartografía y la aplicación de análisis geográfico de los expertos en la inteligencia de la ubicación. Diseñado para visualizar fácilmente las relaciones entre los datos y la geografía.

MapInfo Professional ayuda a los analistas de negocios, planificadores, profesionales de SIG - incluso los no usuarios de SIG a adquirir más conocimientos de sus mercados, compartir información, mapas y gráficos y mejorar la toma de decisiones estratégicas. (Erdas, 2000)

Es necesario destacar que estos SIG líderes mundiales son todos realizados con herramientas propietarias, por lo que el costo de adquisición es bastante alto.

1.9 Desarrollo de los SIG en Cuba

La isla de Cuba no se ha quedado exenta del desarrollo de los SIG, en 1987 surge el Instituto de Geografía de la Academia de Ciencias de Cuba con el objetivo fundamental de actualizar el Atlas Nacional.

El Departamento de Computación y Matemática Aplicada del Instituto Cubano de Hidrografía (hoy GEOCUBA) desarrolla a partir de 1990 el producto TeleMap que en su versión actual constituye una herramienta muy poderosa para el diseño de un SIG y ha sido ampliamente generalizado en todo el país.

Otro sector que utiliza los Sistemas de Información Geográfica es la Salud, donde especialistas han creado un SIG de la Estadística de Salud de Cuba (SIG-ESAC) que permite la representación cartográfica de las estadísticas de salud de Cuba de una forma muy sencilla y de fácil aprendizaje. Este permite cartografiar y realizar diferentes tipos de análisis con respecto a diversos aspectos de la salud como: morbilidad, mortalidad, población, recursos y servicios; el diseño de la base de datos, creada para este sistema, puede emplearse en otros sistemas de información geográfica, utilizados para desarrollar diversos análisis epidemiológicos como el Sistema de Información Geográfica de Epidemiología (SIG-EPI).

También se han desarrollado los SIG en proyectos relacionados con el planeamiento urbano, la gestión del medio ambiente mediante el seguimiento de las zonas de posibles inundaciones, éste software se considera relevante para la gestión de riesgos y constituye una necesidad en las etapas de prevención de desastres naturales y de simulación de los daños que estos producen.

Capítulo 1. “Fundamentación Teórica”

El Instituto de Meteorología de Cuba posee un Sistema de Información Geográfica, que tiene como objetivo brindarle a la población la información meteorológica del país actualizada. Además permite realizar análisis sobre dicha información, constituye un producto de gran utilidad en Cuba ya que sirve de ayuda para la prevención de catástrofes naturales como son los huracanes, ciclones.

Está constituido por varias funcionalidades, algunas de estas son:

- ✚ Vista satelital, permite visualizar una imagen recibida desde un satélite mostrándole a los usuarios una información confiable de los nublados en el área.
- ✚ Vista de radar, ofrece la posibilidad de mostrar una imagen captada desde un radar en tiempo real, lo cual es de gran utilidad para los usuarios.
- ✚ Mapa del tiempo, en el mismo se visualiza como se encuentra la presión superficial en las diferentes zonas.
- ✚ Pronóstico del tiempo, se puede observar cómo se va a comportar el pronóstico del tiempo para los futuros días, teniendo en cuenta las condiciones del clima en la zona.

La Universidad de las Ciencias Informáticas (UCI), las Fuerzas Armadas Revolucionarias (FAR) y los Especialistas de GEOCUBA desarrollaron la plataforma GENESIG la cual constituye una herramienta informática, que surge como necesidad de contar con un producto soberano que sirva como soporte al desarrollo de Sistemas de Información Geográfica en entornos web con tecnologías libres.

El objetivo fundamental de GENESIG es realizar la representación geoespacial de la información asociada a negocios específicos, además debe permitir realizar análisis sobre dicha información. Cuenta con la particularidad de poseer información que otros SIG existentes no brindan, por ejemplo le ofrece al usuario la posibilidad de configurar la representación del mapa en cuanto a estilos y simbología mediante el módulo de catálogo.

Esta es una herramienta poderosa para ampliar las personalizaciones de sistemas de información geográfica bajo los estándares internacionales y un resultado que apoya al componente investigativo aportado para nuevas extensiones y colaboraciones de

Capítulo 1. “Fundamentación Teórica”

entidades de la rama de la Geomática³ que utilizan los Sistemas de Información Geográfica para su funcionamiento y toma de decisiones.

Sobre la base de GENESIG se ha implementado en la Universidad de las Ciencias Informáticas el SIGUCI, que brinda a la comunidad universitaria un sistema capaz de ofrecer y analizar información geográfica detallada y actualizada de la Universidad.

1.10 Conclusiones Parciales

Luego de haber abordado los principales conceptos asociados al problema y profundizado en el objeto de estudio se han identificado los principales procesos que llevan a cabo los Sistemas de Información Geográfica, sus componentes, aplicaciones e importancia y también se ha realizado un estudio del estado del arte de los sistemas nacionales e internacionales para llevar a cabo la modelación del Análisis y Diseño del sistema.

³ Comprende la ciencia, ingeniería y manejo de información geográficamente referenciada.

Capítulo 2. “Tendencias y Tecnologías Actuales”

Capítulo 2. “Tendencias y Tecnologías Actuales de Desarrollo”

2.1 Introducción

En este capítulo se explica mediante una descripción detallada el uso de las herramientas y tecnologías a utilizar para el desarrollo del SIGUCI para celulares basados en la Plataforma GvSIG mini. Además de explicar las principales ventajas que estas ofrecen para el perfeccionamiento de la aplicación.

2.2 Metodologías de Desarrollo de Software

La necesidad de que la forma de trabajo sea organizada, es un principio inviolable a la hora de desarrollar cualquier software informático. Debe contarse con un proceso que ordene e integre las múltiples etapas del desarrollo. Es importante tener una guía para organizar las actividades del equipo de trabajo, así como una estrategia para ordenar las tareas que debe acometer el mismo y establecer los artefactos que deben ser producidos para lograr los objetivos propuestos. Por todo esto es necesario definir una metodología de desarrollo de software, que no es más que un conjunto de procedimientos y pasos que deben ser seguidos para desarrollar un sistema informático. Entre las más utilizadas se encuentran la Programación Extrema (XP), Scrum y el Proceso Unificado de Software (RUP).

Extreme Programming (XP)

Está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP está guiada por una rápida programación basándose en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, en la reutilización de código, en la realización de pruebas a los principales procesos con el objetivo de tratar de obtener los posibles errores futuros. (Beck, y otros, 1999) (14)

Capítulo 2. “Tendencias y Tecnologías Actuales”

SCRUM

Metodología desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, es un término utilizado en el juego de rugby que llevado a la ingeniería significa que el equipo se agrupará para llegar con el apoyo de cada uno de los miembros del proyecto a obtener un producto con calidad que es el objetivo fundamental. Define un proceso empírico, iterativo e incremental de desarrollo, está especialmente indicada para proyectos con un rápido cambio de requisitos. El principio básico es que es muy difícil contar desde el principio con un catálogo completo de funcionalidades, ya que los requisitos van surgiendo conforme el propietario del producto y los usuarios del mismo van haciendo sucesivas aportaciones. El desarrollo de software se realiza mediante iteraciones, denominadas sprints. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Otra característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Calderón, 2007) (15)

RUP (Rational Unified Process - Proceso Unificado de Rational)

El Proceso Unificado de Desarrollo de Software (Rational Unified Process) conocido como RUP, es una metodología de software que permite el desarrollo de aplicaciones a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Aunque con el inconveniente de generar mayor complejidad en los controles de administración del mismo. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto. El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requeridos en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas. RUP es una de las metodologías robustas o pesadas, que presenta entre sus características ser un proceso de desarrollo orientado a objetos, utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Este proceso unificado define “Quién”, “Cómo”, “Cuándo” y “Qué” debe hacerse en el proyecto. Tiene tres características fundamentales: es iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos.

Capítulo 2. “Tendencias y Tecnologías Actuales”

- ✚ **Dirigido por casos de uso:** “Los casos de uso representan los requisitos de software capturados durante el flujo de trabajo de requisitos, la planificación del proyecto se hace en términos de casos de uso, los desarrolladores crean realizaciones de casos de uso en términos de clases y subsistemas, los componentes se incorporan en los incrementos y cada uno realiza un conjunto de casos de uso, y por último se verifica que el sistema implemente los casos de uso correctos para el usuario. En otras palabras los casos de uso guían la arquitectura del sistema, enlazan todas las actividades del desarrollo y dirigen el proceso de desarrollo”.
- ✚ **Centrado en la arquitectura:** “La arquitectura representa la forma del futuro sistema en términos de vistas arquitectónicas, sobre la cual el equipo de desarrollo y los usuarios deben estar de acuerdo, ya que estas describen los elementos del modelo más importantes para su desarrollo, la arquitectura va madurando en las interacciones comenzando con los casos de uso relevantes desde el punto de vista arquitectónico”.
- ✚ **Iterativo e incremental:** “El Proceso Unificado propone que cada fase se desarrolle en iteraciones, ya que el incremento en la complejidad de los sistemas actuales hace que sea factible dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros”. El período de vida del software esta particionado en ciclos, cada ciclo consta de cuatro fases: concepción o inicio, elaboración, construcción y transición y cada vez que termina un ciclo se produce una versión del sistema. Es ideal para proyectos cuyos requisitos no son variables y para grandes equipos de desarrollo. Sin embargo puede adaptarse a diferentes condiciones. En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros son flujos de ingeniería y los 3 últimos de apoyo.

La utilización de las metodologías ágiles estudiadas anteriormente (XP y Scrum) serían subutilizadas teniendo en cuenta que una de sus principales ventajas radica justamente en que son apropiadas para proyectos cuyos requisitos varían con frecuencia, no ocurriendo así con la metodología RUP que es ideal para desarrollar aplicaciones con requisitos constantes. Debido a esta característica de RUP es

Capítulo 2. “Tendencias y Tecnologías Actuales”

posible, como se requiere en el actual proyecto, que las decisiones sean tomadas desde el principio y seguir al pie de la letra una planificación que desembocará en un producto con las funcionalidades deseadas.

RUP propone abundante documentación, característica que es fundamental para el proyecto, ya que en el futuro puede existir la necesidad de reutilizarlo, entenderlo y hasta modificarlo para ajustes a nuevas necesidades que puedan surgir.

Esta metodología cubre todo el ciclo de vida y desarrollo de software, además al ser caracterizada por ser iterativo e incremental, permite el refinamiento constante del sistema y la adición de nuevas funcionalidades a cada una de las iteraciones, y como es dirigido por casos de usos y centrado en la arquitectura asegura que el resultado final del producto cumpla con los requisitos planteados por el cliente o usuario.

Basado en todo lo planteado, en los excelentes resultados ya obtenidos por la Universidad de las Ciencias Informáticas con el uso de esta metodología y que es la misma que utilizan los proyectos del departamento para el desarrollo de sus aplicaciones, no queda duda de que RUP es la metodología más apropiada para lograr el objetivo general de esta investigación.

2.3 Lenguaje Unificad de Modelado – UML

Hoy en día, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo.

Es un lenguaje que permite visualizar, especificar, construir y documentar, modelos de sistemas de software, incluyendo su estructura y diseño, que capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Es un lenguaje de propósito general para el modelado visual y orientado a objetos, que permite una abstracción del sistema y sus componentes, al mismo tiempo posibilita establecer una serie de requisitos y estructuras necesarias para plasmar en un sistema de software previo al proceso intensivo de escribir código. (Larman, Craig, 1999) (16)

En el desarrollo de la aplicación se utilizará este lenguaje de modelado ya que permite construir modelos por ingeniería inversa a partir de sistemas existentes. Presenta un conjunto de herramientas que permiten modelar (analizar y diseñar) sistemas

Capítulo 2. “Tendencias y Tecnologías Actuales”

orientados a objetos, es fácil de aprender y utilizar, con la modelación de los artefactos durante las primeras fases del ciclo de vida del software, posibilita que en fases posteriores los implementadores tengan un mayor dominio y una mejor comprensión sobre qué es lo que se debe implementar, permitiendo tanto al cliente como a los desarrolladores tener una representación real del alcance y la factibilidad que puede o no llegar a tener el producto.

Es importante tener en cuenta que cuanto más complejo es el sistema que se quiere realizar, más importante es el uso de UML.

2.4 Herramientas y Tecnologías

En la actualidad existen varias herramientas que permiten el desarrollo de aplicaciones para celulares, entre ellas se encuentra el GvSIG Mobile la cual permite trabajar con varios tipos de dispositivos móviles, tales como PDAs (Asistente Digital Personal o Computadora de bolsillo), Smartphone (Teléfono inteligente), Tablet PCs (Computadora Portátil), de ahí su diferencia con el gvSIG mini, que es solamente para celulares con funcionalidades básicas y de poca complejidad, esta herramienta es la seleccionada a utilizar ya que el desarrollo de este sistema está dirigido a celulares que cumplan con estas características.

gvSIG mini: Es una aplicación diseñada para teléfonos celulares estándar que permite la visualización y navegación sobre cartografía digital estructurada en tiles o teselas, procedente de servicios web OGC (Open Geospatial Consortium) como WMS (Web Mapping Service) y de servicios como OpenStreetMap (OSM), Yahoo Maps, Maps Bing, así como el almacenamiento en caché para reducir al mínimo el ancho de banda.

Además permite un conjunto de funcionalidades útiles no solo para perfiles técnicos sino también para el uso diario en la vida cotidiana, como son la búsqueda de direcciones o el cálculo de rutas.

gvSIG mini consta de dos productos:

- 🚦 gvSIG mini para Android. Dirigido a teléfonos móviles con sistema operativo Android.

Capítulo 2. “Tendencias y Tecnologías Actuales”

- ✚ gvSIG mini para Java. Dirigido a teléfonos móviles que permiten aplicaciones Java CLDC (Configuración de dispositivos conectados limitada) /MIDP (Información de Perfil del dispositivo móvil).

Es una aplicación que puede ser descargada y usada libremente, convirtiéndose en una plataforma para el desarrollo de nuevas soluciones y aplicaciones en el campo de los servicios basados en ubicación.

gvSIG mini puede acceder a servicios geoespaciales como NameFinder, para la búsqueda de puntos de interés y YOURS (Yet Another Open Street Map Routing Service) para el cálculo de rutas y la renderización de la información vectorial del lado del cliente.

El análisis inicial de gvSIG mini incluye algunos usos básicos de la aplicación, siendo los casos iniciales los expuestos a continuación:

- ✚ Navegación sobre el mapa: El usuario podrá visualizar el mapa y moverse por él, según cuatro direcciones (arriba, abajo, derecha e izquierda) además de poder ejecutar el zoom más y el zoom menos.
- ✚ Gestión de capas (añadir/borrar capas): El usuario podrá pedir nuevas capas a un servicio de mapas WMS a través de la operación GetCapabilities y posteriormente visualizar las mismas.
- ✚ Gestión básica de puntos de ruta (añadir/borrar puntos de ruta): El usuario podrá añadir y borrar puntos de una ruta.
- ✚ Cliente NameFinder: El usuario podrá encontrar direcciones y puntos de interés y mostrarlos sobre el mapa.
- ✚ Cliente de rutas: Una vez que el usuario define dos puntos una ruta podrá calcular la ruta y mostrarla sobre el mapa. (17)

2.5 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computación) constituyen sistemas para el modelado de software. En el mercado de las

Capítulo 2. “Tendencias y Tecnologías Actuales”

herramientas CASE se identifican las dos más usadas en el modelado de sistemas, Rational Rose Enterprise Edition y Visual Paradigm (VP).

Tanto Rational Rose como Visual Paradigm facilitan la organización de los modelos de acuerdo con el orden lógico que propone RUP.

Visual Paradigm, permite visualizar, diseñar y documentar diagramas de UML 2.1 en un entorno de diseño intuitivo (Visual Paradigm, 2010). Mientras que el Rational Rose Enterprise Edition solo permite el diseño de diagramas de UML 1.x (IBM, 2010). (18)

Visual Paradigm permite realizar una rápida construcción de la aplicación con calidad, además de mostrar características que son favorables para el trabajo con tecnologías libres. Brinda distintas perspectivas del sistema, es independiente de la plataforma y dotado de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección del SIGUCI, por lo que se garantiza una muy buena calidad del producto final.

Esta herramienta CASE es una de las que soporta el análisis textual, una técnica que se utiliza para la captura de requisitos. Posibilita generar código a partir de los diagramas de clases de diseño, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código. Intercambia información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose.

Visual Paradigm ayuda al equipo de desarrollo a controlar el progreso del proyecto y brinda un medio de comunicación. Facilita el modelado de un sistema donde cada involucrado puede crear sus propias vistas arquitectónicas (Vista Lógica, Vista de Procesos, Vista de Desarrollo, Vista Física y Vista de Casos de Uso).

Ante cambios que surjan en un negocio, permite adaptar rápidamente los modelos realizados a dichas modificaciones, lo cual evita escribir código sin analizar los cambios en el modelo. Además es posible descomponer un modelado de un sistema en unidades controladas, generar código en un lenguaje específico a partir de los diseños realizados.

Por todo lo anterior se optó por modelar la solución informática a construir con Visual Paradigm – UML 6.4 Enterprise Edition.

2.6 Entorno de Desarrollo Integrado (IDE)

Un IDE es un programa compuesto por un conjunto de herramientas que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. A continuación se exponen algunas de las características de NetBeans y Eclipse con el objetivo de seleccionar la más completa para el desarrollo de la aplicación.

NetBeans, es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++.

NetBeans dispone de soporte para crear interfaces gráficas de forma visual, control de versiones, colaboración entre varias personas y creación de aplicaciones compatibles con teléfonos móviles. Tiene una interfaz muy amigable e intuitiva, característica fundamental en un IDE. Es de múltiples lenguajes y permite la confección de diagramas UML.

NetBeans tiene un muy buen soporte de webapps (.war, jsp y servlets) del cual eclipse carece (los plugins por lo general no son ni tan potentes ni tan sencillos como el módulo que viene preinstalado en NetBeans).

Eclipse, es una plataforma de software de código abierto independiente que ha sido y es muy utilizada para desarrollar entornos de desarrollo, del mismo modo se puede usar para otros tipos de aplicaciones cliente. Lo mejor de Eclipse es que tiene una gran comunidad de usuarios extendiendo constantemente las aplicaciones.

Eclipse tiene buen soporte de refactorización del cual NetBeans carece (solo hay un pequeño módulo gratuito con pocas refactorizaciones) así como ayuda a la codificación (con las maravillosas sugerencias).

Después de haber realizado el estudio de acuerdo a las características que presenta cada uno de estos IDE se ha seleccionado NetBeans porque permite vincular todas las

Capítulo 2. “Tendencias y Tecnologías Actuales”

tecnologías seleccionadas para desarrollar la aplicación, adquirirlo reportaría muchísimos beneficios y con un costo mínimo ya que es de código abierto y gratuito.

2.7 Lenguaje de Programación

2.7.1 Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se construye un programa en Java podrá funcionar en cualquier ordenador del mercado.

Características de Java

Orientación a objetos

En este aspecto Java fue diseñado partiendo de cero, no siendo derivado de otro lenguaje anterior y no tiene compatibilidad con ninguno de ellos. En Java el concepto de objeto resulta sencillo y fácil de ampliar. Además se conservan elementos "no objetos", como números, caracteres y otros tipos de datos simples.

Riqueza semántica

Pese a su simpleza se ha conseguido un considerable potencial, y aunque cada tarea se puede realizar de un número reducido de formas, se ha conseguido un gran potencial de expresión e innovación desde el punto de vista del programador.

Robusto

Java verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse, de manera que se consigue un alto margen de codificación sin errores. Se realiza un descubrimiento de la mayor parte de los errores durante

Capítulo 2. “Tendencias y Tecnologías Actuales”

el tiempo de compilación, ya que Java es estricto en cuanto a tipos y declaraciones, y así lo que es rigidez y falta de flexibilidad se convierte en eficacia. Respecto a la gestión de memoria, Java libera al programador del compromiso de tener que controlar especialmente la asignación que ésta hace a sus necesidades específicas. Este lenguaje posee una gestión avanzada de memoria llamada gestión de basura, y un manejo de excepciones orientado a objetos integrados.

Modelo de objeto rico

Existen varias clases que contienen las abstracciones básicas para facilitar a los programas una gran capacidad de representación. Para ello se contará con un conjunto de clases comunes que pueden crecer para admitir todas las necesidades del programador. Además la biblioteca de clases de Java proporciona un conjunto único de protocolos de Internet. El conjunto de clases más complicado de Java son sus paquetes gráficos AWT (Abstract Window Toolkit) y Swing. Estos paquetes implementan componentes de una interfaz de usuario gráfica básica común a todos los ordenadores personales modernos.

Fácil aprendizaje

El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Así se ha creado un lenguaje simple, aunque eficaz y expresivo pudiendo mostrarse cualquier planteamiento por parte del programador sin que las interioridades del sistema subyacente sean reveladas.

Interactivo y animado

Uno de los requisitos de Java desde sus inicios fue la posibilidad de crear programas en red interactivos, por lo que es capaz de hacer varias cosas a la vez sin perder rastro de lo que debería suceder y cuándo (19).

2.8 Conclusiones Parciales

A partir de las comparaciones de los distintos Lenguajes de Programación, Herramientas Case, Entornos de Desarrollo y Metodologías para el desarrollo del Sistema de Información Geográfica de la Universidad, se lograron identificar las

Capítulo 2. “Tendencias y Tecnologías Actuales”

siguientes: Java como lenguaje de programación, NetBeans como entorno de desarrollo. La metodología utilizada para guiar el proceso de desarrollo es RUP, se Emplea como lenguaje de modelado UML, como herramienta CASE para visualizar y modelar los diagramas, Visual Paradigm.

Capítulo 3. “Presentación de la Solución Propuesta”

Capítulo 3. “Presentación de la Solución Propuesta”

3.1 Introducción

Alcanzar cierto nivel de conocimiento sobre el problema en cuestión es un paso fundamental en los inicios del proceso de desarrollo de un software. En este capítulo se comienza la investigación del medio donde se implementará la aplicación con el objetivo de definir las condiciones o capacidades que debe ofrecer el sistema. Serán abordados los principales aspectos de los flujos de trabajo Modelamiento del Negocio y Requisitos, que son, entre los flujos de trabajo de ingeniería que propone RUP, los que tienen mayor peso en la fase de Inicio.

Se conceptualiza el entorno mediante un modelo de dominio en el cual se analizan cada una de las entidades y conceptos presentes en el contexto donde se insertaría la aplicación, además de las relaciones existentes entre cada uno de estos, se especifican los requisitos funcionales y no funcionales que deberá cumplir la solución propuesta, se presentan los diagramas de casos de uso del sistema, se describen los actores y se detallan los casos de uso del sistema.

3.2 Modelo de Dominio

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.

El modelo de dominio captura los tipos de objetos más importantes que existen, teniendo como características que permite ayudar a los usuarios, clientes y desarrolladores a utilizar un vocabulario común, con el objetivo de lograr una mejor comprensión del contexto en que se realiza el sistema.

3.3 Diagrama de Clases del Dominio

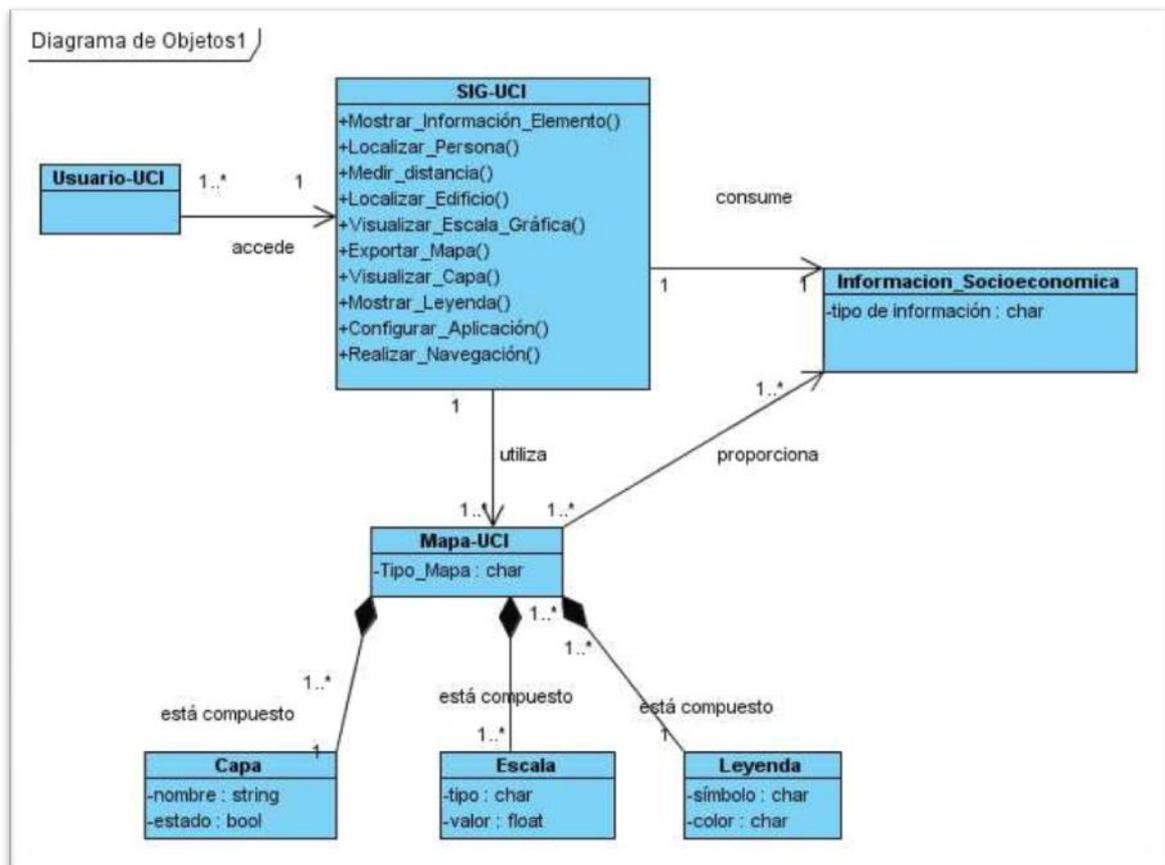


Figura 3 Diagrama de Clases del Dominio

3.4 Descripción del Diagrama de Clases del Dominio

Un Usuario-UCI es la persona que tenga en su poder un celular, éste puede acceder a la aplicación (SIG-UCI) donde están implementadas cada una de las funcionalidades, para esto consume una gran cantidad de información socioeconómica. El sistema utiliza una cartografía relacionada al mapa de la universidad, que está compuesto por capas, estos mapas se agrupan por capas independientes y contienen además la escala y leyenda para lograr un mejor entendimiento por parte de los usuarios.

3.5 Glosario de Términos del Dominio

Usuario_UCI: Usuario portador de un dispositivo celular y que utilizará las funcionalidades de la aplicación.

Capítulo 3. “Presentación de la Solución Propuesta”

SIG_UCI: Aplicación final con fines de localización que brindará servicios que serán utilizados por la comunidad universitaria.

Mapa_UCI: Representación gráfica y métrica de una porción de territorio sobre una superficie generalmente plana y bidimensional. Las propiedades métricas permiten que sobre éste se puedan tomar medidas de distancias, ángulos, etc. Obteniendo así resultados aproximadamente exactos.

Escala: Relación que existe entre la distancia que separa a dos puntos en un mapa y la distancia real de estos en la superficie. Cuanto mayor sea la escala, mayor será la exactitud de los datos. Por lo general los mapas con escala pequeña son para grandes extensiones y menos detallados que los de escala grande. Las escalas se pueden expresar de tres formas distintas en los mapas, con escala gráfica, con expresiones o palabras, y con porciones o fracciones.

Leyenda: Es la explicación de los símbolos, colores, sombras y tramas que se emplean en los mapas, su objetivo principal es brindar información muy valiosa a la hora de leer un mapa. Esta suele encontrarse en la parte inferior del mapa o en un recuadro.

Capa: Es un mapa monotemático que se integra a una serie de mapas para formar uno de carácter más específico en un sistema de información geográfica.

Información Socioeconómica: Conjunto de datos procesados y organizados que contienen información referente al aspecto social y económico de un lugar específico.

3.6 Captura de Requisitos

Se aborda una visión general del problema que se está resolviendo y las áreas clave que se deben tratar en la solución. Es el flujo de trabajo que da una visión de qué es necesario hacer para dar respuesta a las solicitudes del usuario, dado que tiene como objetivo principal describir que hará el sistema. En el mismo se capturan los requisitos funcionales y no funcionales que el sistema debe poseer lo que posibilita a desarrolladores y clientes un entendimiento común, además provee una base para planificar el costo y la duración del proyecto, establecer y mantener un acuerdo entre el equipo de desarrollo y los clientes acerca de lo que el sistema debe hacer.

Capítulo 3. “Presentación de la Solución Propuesta”

3.7 Requisitos Funcionales del Sistema

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. El Sistema de Información Geográfica debe ser capaz de:

RF1. Permitir al usuario localizar un edificio en el mapa. Ésta localización se puede ejecutar por el siguiente criterio:

-número del edificio (Formato: numérico, Obligatorio: Si)

RF2. Permitir al usuario localizar una persona en el mapa. Ésta localización se puede ejecutar por el siguiente criterio:

-usuario (Formato: alfanumérico, Obligatorio: Si)

-solapín (Formato: alfanumérico, Obligatorio: Si)

-carnet de identidad (Formato: numérico, Obligatorio: Si)

RF3. Permitir al usuario obtener información de un elemento seleccionado previamente.

RF4. Permitir al usuario exportar un mapa en formato jpg.

RF5. Permitir al usuario visualizar y ocultar la Escala Gráfica sobre el mapa.

RF6. Permitir al usuario visualizar y ocultar una capa en el mapa.

RF7. Permitir al usuario configurar la aplicación de acuerdo a sus necesidades.

RF8. Permitir al usuario medir la distancia entre puntos seleccionados en el mapa. Ésta medición se puede ejecutar por el siguiente criterio:

-punto1 (Formato: numérico, Obligatorio: Si)

-punto2 (Formato: numérico, Obligatorio: Si)

RF9. Permitir al usuario visualizar y ocultar la leyenda sobre el mapa.

RF10. Permitir al usuario aumentar, disminuir y mover el mapa.

3.8 Requisitos No Funcionales del Sistema

Los requisitos no funcionales sirven para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no funcionales (como cuán usable, seguro, conveniente y agradable) pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Por lo tanto la aplicación debe cumplir con las siguientes propiedades:

Hardware

-  Teléfono Móvil con soporte para Java ME.

-  Dispositivos con configuración CLDC.

-  Dispositivos con perfil MIDP 2.0.

-  **Restricciones de diseño:** El análisis y diseño de la aplicación estará basado en la Metodología RUP con el uso del lenguaje de modelado UML .Se utilizará como Herramienta CASE Visual Paradigm para el modelado de los artefactos que se generan para cada uno de los flujos de trabajo. Como lenguaje de Programación se propone Java y el entorno de desarrollo a utilizar es el NetBeans.

-  **Rendimiento:** La velocidad de procesamiento de la información, el tiempo de respuesta y la actualización estarán dados por la cantidad de información a procesar.

-  **Interfaz de Usuario:** Las funcionalidades principales del sistema estarán orientadas a íconos para un mayor reconocimiento por parte del usuario.

-  **Soporte:** La aplicación recibirá mantenimiento cada cierto tiempo, determinado por el equipo de desarrollo. La cartografía será actualizada por la UCI en el período de tiempo estimado.

-  **Confiabilidad:** La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.

-  **Usabilidad:** La aplicación podrá ser usada por todos los usuarios que tengan en su poder un dispositivo celular que cuente con los requisitos de hardware para soportarlo.

Capítulo 3. “Presentación de la Solución Propuesta”

- ✚ **Requisitos de licencia:** El producto final podrá ser catalogado como libre, ya que las herramientas a utilizar tienen licencias libres, por lo que podrá ser utilizado por terceros que lo puedan modificar, distribuir y utilizar sin la autorización de los desarrolladores.

3.9 Descripción del Sistema Propuesto

El sistema que se propone debe estar compuesto por los actores y casos de uso que se describen a continuación.

3.9.1 Descripción de los Actores

Un actor no es parte del sistema en desarrollo, es un agente externo que interactúa con el mismo en pos de obtener un resultado esperado.

Actor	Descripción
Usuario_UCI	Es la persona que utilizará las funcionalidades de la aplicación.

3.10 Diagrama de Caso de Uso del Sistema

Luego de haber realizado anteriormente la descripción de los requisitos, quienes constituyen las funcionalidades esenciales del sistema, se le da paso a la siguiente etapa de construcción de un software que es la descripción del Diagrama de Casos de Uso del Sistema (DCUS). Este describe un sistema en cuanto a su utilización, es un modelo que contiene actores, casos de uso y las relaciones que se establecen entre ellos.

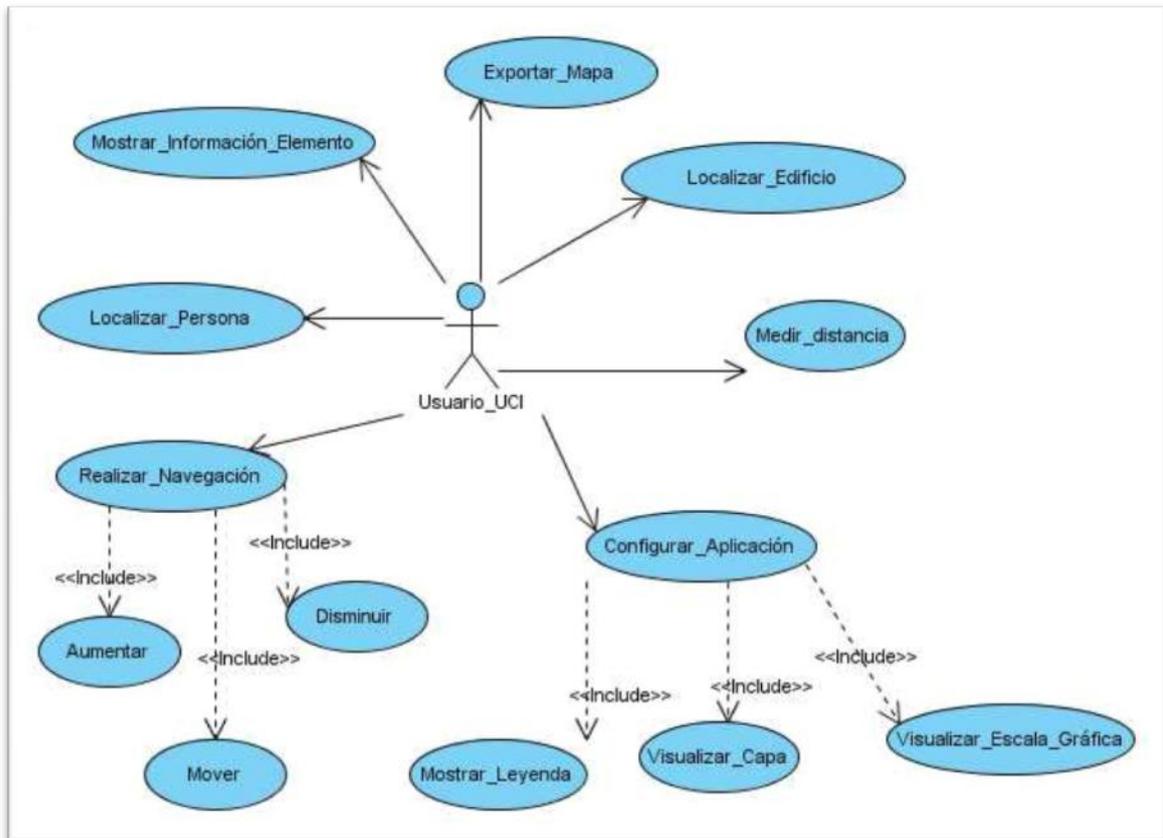


Figura 4 Diagrama de Casos de Uso del Sistema.

3.11 Descripción Textual de los Casos de Uso

Cada uno de los casos de uso identificados presenta características particulares que para su mejor entendimiento se hace necesario describirlas textualmente. Además para lograr un mejor rendimiento del tiempo en la construcción del software es recomendable identificar los casos de uso arquitectónicamente significativos, que en la descripción textual se diferencian porque tienen la prioridad de crítico.

Capítulo 3. “Presentación de la Solución Propuesta”

3.11.1 Descripción Textual del Caso de Uso Localizar_Edificio.

Caso de Uso:	Localizar_Edificio	
Actor:	Usuario_UCI	
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar un edificio en el mapa.	
Resumen:	El caso de uso se inicia cuando el actor selecciona la opción de acceder al formulario de búsqueda y termina cuando se le muestra el edificio pintado en el mapa, además de la visualización de los datos del edificio.	
Precondiciones:	Para poder mostrar la información, la capa de edificios tiene que ser seleccionada y estar visible.	
Referencias	RF1	
Prioridad	Critico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor selecciona del menú principal la opción: - Localizar Edificio	2. El sistema le muestra al usuario una ventana para que introduzca el número del edificio del que necesita conocer la información.	
3. El actor introduce el número del edificio y selecciona el botón Aceptar a través del teclado del teléfono.	4. El sistema valida que sea correcto (no debe admitir caracteres extraños, campo número de edificio debe estar lleno, y el número a introducir debe ser menor que 150) y los procesa. Si introduce caracteres extraños (“@_*-#.+”) o letras(A-Z ,a-z) ir a la Sección “Caracteres Extraños o Letras” Si deja el campo vacío ir a la Sección “Campo Vacío” Si el número introducido es mayor que 150 ir a la Sección “No existe Edificio”	

Capítulo 3. “Presentación de la Solución Propuesta”

	5. El caso de uso termina cuando el sistema muestra el edificio que cumple las condiciones anteriores, pintado de otro color en el mapa y muestra los datos del edificio en una ventana (cantidad apartamentos, cantidad residentes, cantidad de pasos escalera).
	6. El sistema devuelve el mapa centrado y vuelve a la interfaz principal.
Prototipo de Interfaz	
<i>Interfaz 1</i>	
Sección “Caracteres Extraños”	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje de error informando “No se admiten caracteres extraños”.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 2</i>	
Sección “Campo Vacío”	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje de error informando “Introduzca el número del edificio”.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 3</i>	

Capítulo 3. “Presentación de la Solución Propuesta”

Sección “No Existe Edificio”	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje de error informando “No existe edificio”.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 4</i>	
Poscondiciones	Se debe mostrar una ventana con los datos asociados al edificio introducido por el usuario.

Tabla 1 Descripción Textual del Caso de Uso Localizar_Edificio.

3.12 Conclusiones Parciales

El Modelo de Dominio realizado a partir de los procesos identificados permitió conocer todos los términos y conceptos presentes en el entorno, los cuales fueron descritos y especificados en un glosario de términos y representados en un diagrama de clases el cual permitió comprender mejor el funcionamiento e interrelación de los mismos.

Mediante la realización del Levantamiento de Requisitos, se consiguió identificar las funcionalidades que la aplicación debe brindar y las restricciones sobre las que va a operar.

La conformación del Diagrama de Casos de Uso del Sistema permitió traducir los requisitos funcionales en interacciones del actor con el sistema, mostrando así las funcionalidades que brindará el software.

La descripción detallada del flujo de eventos de cada caso de uso en un lenguaje formal y entendible permitió comprender en detalle los procesos incluidos en el cumplimiento de los objetivos trazados para cada caso de uso.

Capítulo 4. “Construcción de la Solución Propuesta”

Capítulo 4. “Construcción de la Solución Propuesta”

4.1 Introducción

En el presente capítulo se realizan los modelos de Análisis y Diseño. El Diseño presenta cómo está construido el sistema, lo cual se realiza a través de los diagramas de clases, que tienen la finalidad de describir la interacción entre las distintas secciones de la aplicación. También se muestran los principios de diseño utilizados, se muestra el Modelo de Despliegue, que contiene el Diagrama de Despliegue y la explicación de sus componentes físicos.

4.2 Modelo del análisis

“Durante el análisis, se analizaron los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura”. (Jacobson y otros, 2000)

Aunque en el modelo del análisis hay un refinamiento de los requisitos, no se tiene en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución.

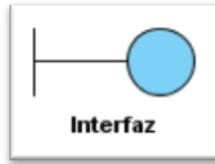
El modelo del análisis es un artefacto que usualmente se genera para entender claramente los requisitos y realizar mejor el diseño. Es usado para representar la estructura global del sistema, describe la realización de casos de uso y sirve como una abstracción del Modelo de Diseño y puede contener: las clases y paquetes de análisis, las realizaciones de los casos de uso, las relaciones y los diagramas.

4.3 Diagramas de Clases del Análisis

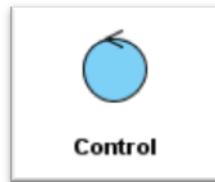
En el diagrama de clases de análisis se representan los conceptos fundamentales que comprende el dominio del problema. Estos diagramas muestran las clases que participan en las realizaciones de los distintos casos de uso, representan las definiciones y relaciones entre las clases. Las clases del análisis se clasifican en Interfaz, de Control o Entidad.

Capítulo 4. “Construcción de la Solución Propuesta”

- ✚ **Clase Interfaz:** Modela la interacción entre el sistema y sus actores.



- ✚ **Clase Control:** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.



- ✚ **Clase Entidad:** Modelan información que posee larga vida y que es a menudo persistente.



La siguiente imagen muestra el diagrama de clases del análisis correspondiente al caso de uso Localizar Edificio. Representa el actor y su interacción con el sistema mediante clases interfaces, así como el resto de las clases que intervienen y sus relaciones.

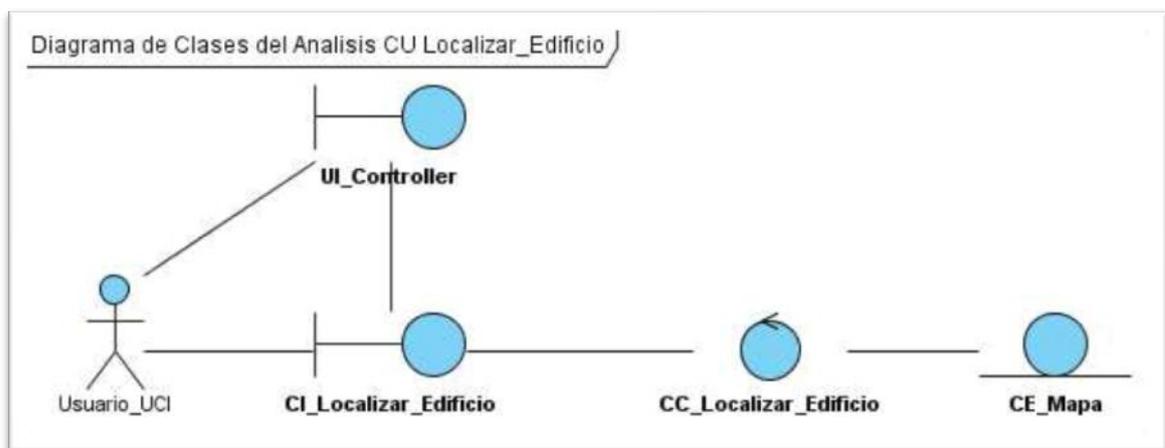


Figura 5 Diagrama de Clases del Análisis CU Localizar Edificio

Capítulo 4. “Construcción de la Solución Propuesta”

4.4 Diagramas de Interacción

Los diagramas de interacción son diagramas que describen cómo grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso. Los diagramas de interacción capturan el comportamiento de los casos de uso y tienen dos formas de expresarse: diagramas de secuencia y diagramas de colaboración.

- ✚ **Diagramas de secuencia:** Muestran las interacciones expresadas en función de secuencias temporales. Destaca la ordenación temporal de los mensajes.
- ✚ **Diagramas de colaboración:** Muestran las relaciones entre los objetos y los mensajes que intercambian. Destaca la organización estructural de los objetos que envían y reciben mensajes.

La siguiente imagen expone el diagrama de colaboración del análisis del caso de uso Localizar Edificio.

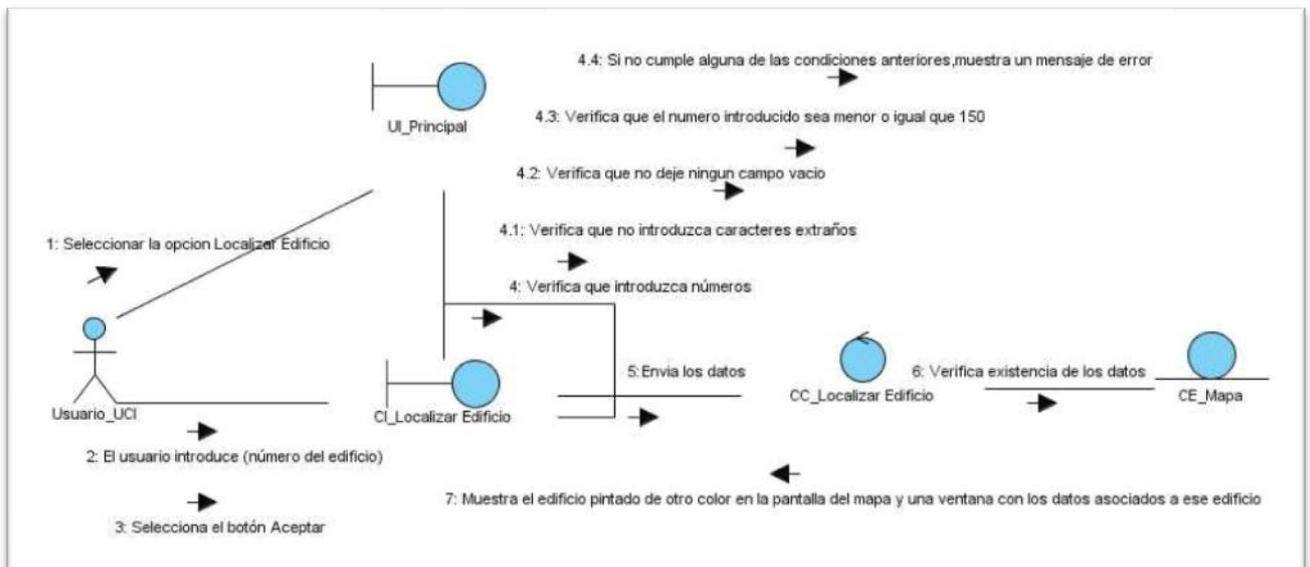


Figura 6 Diagrama de Colaboración CU Localizar Edificio

4.5 Modelo de Diseño

“El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto

Capítulo 4. “Construcción de la Solución Propuesta”

con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el Modelo de Diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación”. (Jacobson y otros, 2000)

4.5.1 Diagramas de Clases del Diseño

Los diagramas de clases del diseño son los diagramas principales en el flujo de trabajo análisis y diseño para obtener un sistema. En él, la estructura de clases del sistema se especifica con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. La siguiente imagen muestra el Diagrama de Clases del Diseño del Sistema CU Localizar Edificio.

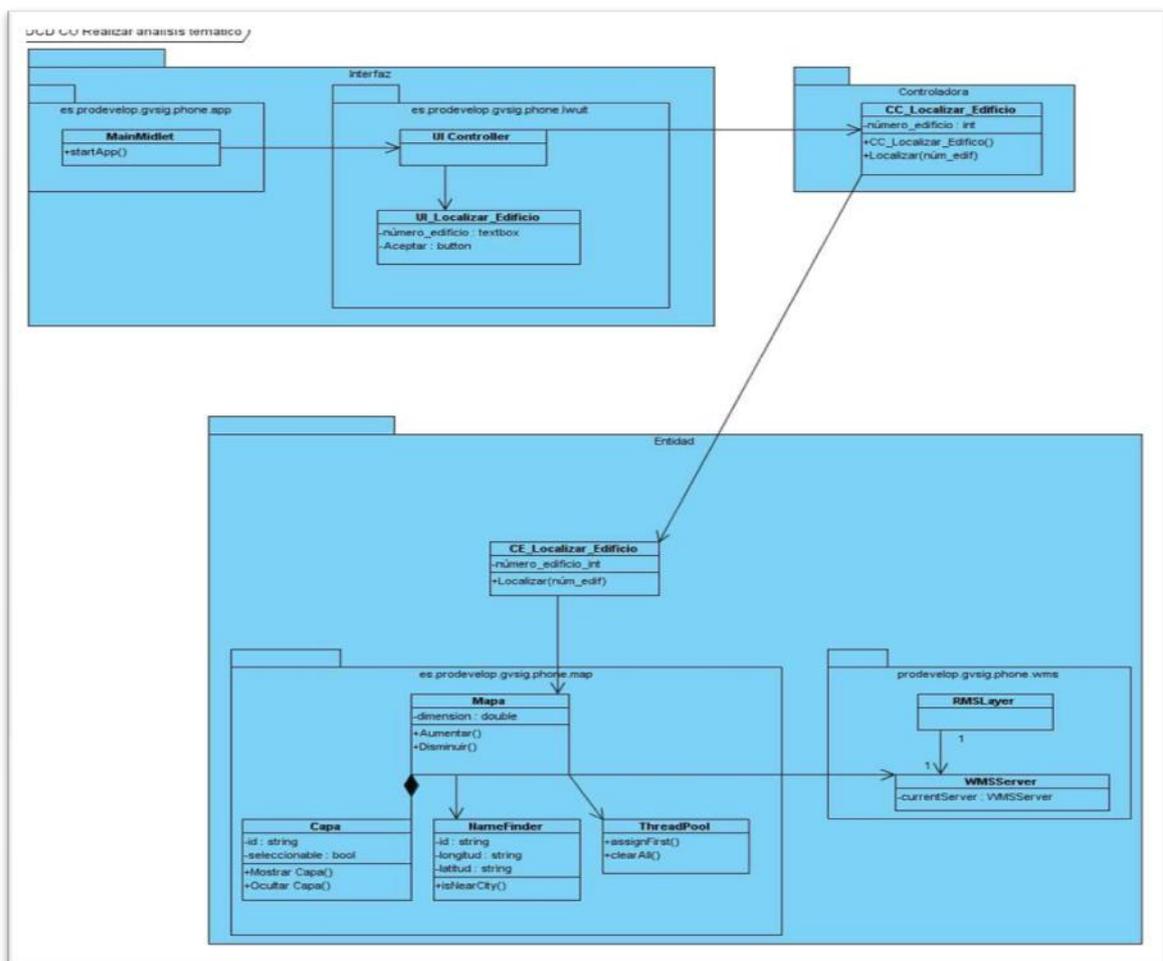


Figura 7 Diagrama de Clases del Diseño CU Localizar_Edificio

4.6 Arquitectura definida para el Sistema

“La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requisitos de desempeño de un sistema, así como requisitos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.” (Kruchten, 1995) (20)

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema informático. Lo usual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Precisamente, las arquitecturas más universales son:

- ✚ **Monolítica:** Donde el software se estructura en grupos funcionales muy acoplados.
- ✚ **Cliente-servidor:** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- ✚ **Arquitectura 3 Capas/N Capas:** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia).

Arquitectura 3 Capas.

Capítulo 4. “Construcción de la Solución Propuesta”



Figura 8 Representación Gráfica de la Arquitectura 3 Capas

La capa de presentación: Esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Básicamente, se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa, manteniendo una comunicación exclusiva con la capa de negocio.

La capa de negocio: Es la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación, las peticiones que el usuario ha realizado, y responsabilizándose de que se le envíen las respuestas adecuadas a la capa de presentación. Se considera una capa intermedia, a medio camino entre la capa de presentación y la capa de datos, puesto que se relaciona con ambas y por supuesto, procesa también la información devuelta por la capa de datos.

La capa de acceso a datos: Por último, la capa donde se almacenan los datos. Mediante la capa de negocio, se puede encargarse de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor (o los gestores) de bases de datos que la aplicación requiera.

Las principales ventajas de este modelo son las siguientes:

- ✚ **Separación de funciones:** Todo lo relacionado con la interfaz del usuario va en una capa, las reglas de negocio en otra y el manejo de datos en una tercera capa. No se mezcla en una capa código correspondiente a otra.

Capítulo 4. “Construcción de la Solución Propuesta”

- ✚ **Reutilización:** El código correspondiente a una capa puede ser reutilizado desde varias partes de la capa inmediatamente superior.
- ✚ **Escalabilidad:** Sabiendo dónde está el código correspondiente a cada capa, pueden realizarse modificaciones dentro de una capa para mejorar o aumentar el tamaño del sistema de software, con un mínimo impacto en las capas restantes.
- ✚ **Facilidad de mantenimiento:** Mediante esta división, es mucho más sencillo localizar errores en el código o efectuar mejoras.

4.7 Modelo de Despliegue

El Diagrama de Despliegue se utiliza para modelar la arquitectura en tiempo de ejecución de un sistema. Muestra la configuración de los elementos de hardware (nodos) utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Se representa mediante un grafo de nodos unidos por conexiones de comunicación, los nodos son elementos físicos de ejecución que representa un recurso computacional, que generalmente tiene por lo menos memoria y a veces también capacidad de proceso. Los nodos sirven para modelar la topología del hardware sobre el que se ejecuta el sistema. Un nodo representa, generalmente, un procesador o un dispositivo sobre el que se pueden desplegar los componentes y debe tener un nombre asignado que lo distinga del resto de nodos.

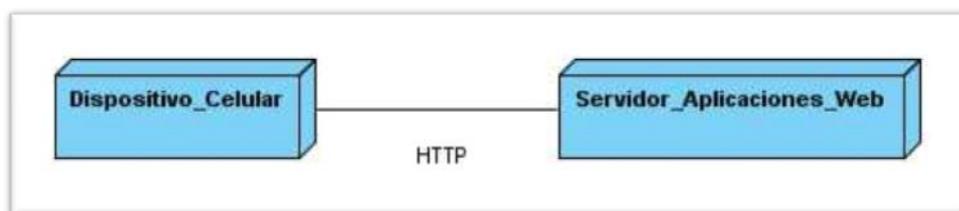


Figura 9 Modelo de Despliegue del Sistema

4.8 Validación de los Requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos es muy importante, pues un levantamiento de requisitos con errores que no se detecten a

Capítulo 4. “Construcción de la Solución Propuesta”

tiempo conduce a resultados inesperados provocando costos excesivos y gran pérdida de tiempo (Ingeniería, 2010).

Existen algunas propuestas de técnicas para la realización de la validación, la mayoría se destinan a revisar los modelos obtenidos en la definición de requisitos. A continuación se muestran las que se usarán para verificar la calidad de los artefactos del análisis.

- ✚ **Revisiones (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.
- ✚ **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.
- ✚ **Auditorías:** Revisión de la documentación, esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.
- ✚ **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (Durán A., 1999). Es necesario ver qué objetivos cubre cada requisito, de tal forma se puede detectar inconsistencias u objetivos no cubiertos.

A continuación se presenta una combinación de las diferentes técnicas aquí explicadas, aplicando las mismas a los artefactos del análisis para validar la calidad de los resultados obtenidos.

4.9 Estudio de la validación por fases

La validación se refiere a un conjunto de diferentes actividades, que aseguran que el software construido se ajusta a los requisitos del cliente. El cual, en la Ingeniería de Requisitos permite demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente (Pressman, 2002).

Para garantizar la calidad de los artefactos del análisis del Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas, se aplicará el procedimiento para la validación de requisitos de software (PQRS), realizado en un

Capítulo 4. “Construcción de la Solución Propuesta”

trabajo de diploma que tiene como título: “Procedimiento para la validación de requisitos de software”.

Éste tiene como propósito aplicarse a todos los proyectos de la Universidad de las Ciencias Informáticas, que contribuirá a mejorar la calidad de los requerimientos que son entregados al cliente (Padrón Blanco, 2009). (21)

Se pretende validar los documentos que aportan gran flujo de información referente a los requisitos de software, los cuales son:

- ✚ Documento de especificación de requisitos de software.
- ✚ Modelo de casos de uso del sistema.

El PVRS presenta un conjunto de actividades que servirán de guía para el proceso de validación, este se ha dividido en tres fases para mejor organización:

I: Validar los requisitos de software.

II: Validar el documento de especificación de requisitos de software.

III: Validar el modelo de casos de uso del sistema.

A continuación se explica cada una de las actividades realizadas en cada fase del procedimiento, la cuales se realizaron en 3 revisiones dando a conocer los resultados obtenidos de cada validación.

Fase I. Validación del listado de requisitos de software

En esta fase se validó el listado de requisitos definidos en el sistema, se aplicó la técnica revisiones para corregir la información del listado de los mismos. Se utilizó para revisar que los requerimientos se encuentren contenidos en un documento tangible tanto para los clientes y desarrolladores. Además de verificar que los requerimientos no sean ambiguos, siendo estos posibles de probar, así como que describa de forma concisa qué es lo que debe hacer el sistema. También su descripción debe ser lo más detallado posible, siendo conciso en su información.

Para esta primera fase se aplicó además, las listas de chequeo del Centro de Calidad de Software (Calisoft), con estas listas se aplicarán un conjunto de preguntas a los requisitos, que permitirá verificar la ambigüedad, factibilidad y si estos pueden ser probados. Además de identificar los elementos de entrada y salida y evaluar si su

Capítulo 4. “Construcción de la Solución Propuesta”

impacto es positivo. En esta fase se desarrollaron 3 revisiones para validar el listado de requisitos, a continuación se presentan los resultados de cada una de ellas.

Revisión 1:

En esta primera revisión se detectó que 3 de los requisitos eran ambiguos, y 3 no eran posibles de probar. Además, faltaban 6 requerimientos que no tenían especificado la prioridad. Se identificó que 2 de los requerimientos no describían las necesidades que debía cumplir el software. También se observó que faltaban las interfaces para cada sección de la aplicación.

Revisión 2:

Para esta revisión se corrigieron los problemas encontrados anteriormente, pero aún con los arreglos previamente hechos se encontró errores en el listado de los requerimientos. Se observaron 1 requisito que estaba redactado en uno solo y se decidió separarlo por el nivel de complejidad. Se detectó que las entradas y salidas de cada requerimiento no estaban bien especificadas, y no describían todas las entradas que en realidad contenían cada uno.

Revisión 3:

Al realizar la tercera revisión se habían solucionado los errores que se detectaron en las anteriores. Se observó que se eliminaron las ambigüedades e inconsistencias, además de las redundancias. Los requerimientos eran posibles de verificar y no faltaba ninguno principal; aunque se encontró 2 requerimientos que les hacía falta abundar en su especificación para dejar claro conceptos que podrían confundir.

Como resultado de las 3 revisiones, se puede afirmar que los requisitos obtenidos en el análisis realizado y según los resultados obtenidos en esta etapa de definición de los mismos son correctos, por esto se puede asegurar que en cada revisión se fue perfeccionando cada requisito.

Fase II. Validación del documento de especificación de requisitos

Para esta segunda fase se validó el documento de especificación de requisitos de software, para lograr una mejor calidad en del mismo. Se aplicó la técnica de auditoría con la lista de chequeo especificación de requisitos como se muestra en la siguiente tabla (Ver Tabla # 3), para verificar una parte del documento.

Capítulo 4. “Construcción de la Solución Propuesta”

Indicadores a Evaluar	Evaluación	Cant. de Elementos Afectados	Comentarios
1. ¿Está el documento acorde con la plantilla estándar del proyecto o del expediente de proyecto?	B	Ninguno	
2. ¿Contiene las secciones obligatorias definidas en el expediente?	B	Ninguno	
Elementos definidos por la metodología			
Indicadores a Evaluar	Evaluación	Cant. de Elementos Afectados	Comentarios
1. ¿Debería especificarse algún requisito con más detalle?	B	Ninguno	
2. ¿Han sido abordadas e identificadas los valores de entradas y salidas?	B	Ninguno	
3. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?	B	Ninguno	
4. ¿No aparece un mismo requisito en más de un lugar del documento de especificación?	B	Ninguno	
5. ¿Existe correspondencia entre el modelo de caso de uso, las Especificaciones Suplementarias y las especificaciones de requerimientos?	B	Ninguno	

Capítulo 4. “Construcción de la Solución Propuesta”

6. ¿Se puede verificar cada requisito? (Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento, ejemplo la especificación del caso de uso).	B	Ninguno	
7. ¿Se han identificado los requerimientos de software y de hardware?	B	Ninguno	
Semántica del Documento			
Indicadores a Evaluar	Evaluación	Cant. de Elementos Afectados	Comentarios
1. ¿Ha identificado errores ortográficos?	B		
2. ¿Se entiende claramente lo que se ha especificado en el documento?	R	2	Hay especificaciones que tienen varias interpretaciones.
3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?	B		
4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?	B		

Tabla 2 Ejemplo de elementos definidos en la lista de chequeo Especificación de Requisitos.

Capítulo 4. “Construcción de la Solución Propuesta”

Como resultado de esta lista de chequeo se detectó que hay 2 especificaciones que tienen varias interpretaciones. Después de aplicar al documento de Especificación de Requisitos de Software la lista de chequeo especificación de requisitos, se procede al refinamiento del mismo para que todos los elementos tengan una evaluación de B y así garantizar que no presente ningún error.

Para que el documento de Especificación de Requisitos de Software cumpliera con las características solicitadas, se hizo necesario iterar 2 veces el procedimiento. En la primera iteración se detectó el problema explicado anteriormente que fue solucionado, dando paso a la próxima iteración. En esta última iteración se evidenció el avance de corrección del documento en cuanto al error anterior, quedando con evaluación de B todos los elementos de la lista y garantizando la calidad del documento.

4.10 Métrica para la Calidad de la Especificación de los Requisitos de Software

Existe una lista de características para poder valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: especificidad, corrección, complejidad, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. Aunque muchas de las características anteriores pueden ser de naturaleza cuantitativa (Pressman, 2002). (22)

Para medir las características de la especificación, es necesario profundizar cuantitativamente en la especificidad y en la completitud.

La métrica es el resultado de dividir el número de requisitos para los que los revisores tuvieron interpretaciones idénticas (nui) entre la cantidad de requisitos de software (nr).

$$Q1 = \text{nui} / \text{nr}$$

$$\text{nr} = \text{nf} + \text{nnf}$$

Dónde:

nf: cantidad de requisitos funcionales.

nnf: cantidad de requisitos no funcionales.

Cuanto más cerca de 1 esté el valor de Q1 menor será la ambigüedad de la especificación.

Capítulo 4. “Construcción de la Solución Propuesta”

Se realizaron dos revisiones para llegar a obtener un buen análisis, sin ambigüedad en los requisitos, en las cuales se obtuvieron los siguientes resultados:

Revisión 1:

En esta primera revisión, los revisores tuvieron una interpretación bastante parecida de los requerimientos. A continuación se aplica la ecuación anterior.

$$nf = 10, nnf = 8$$

$$nr = 8 + 7, nr = 15$$

$$Q1 = 15/18$$

$$Q1 = 0.833$$

Revisión 2:

En los requisitos presentados en el documento ERS, todos los revisores tuvieron la misma interpretación, quedando de la siguiente manera la ecuación planteada anteriormente:

$$nf = 10, nnf = 8$$

$$nr = 10+8, nr = 18$$

$$Q1 = 18/18$$

$$Q1 = 1$$

Después de aplicar la ecuación se observó que los requisitos en la primera revisión tenían un nivel bajo de ambigüedad, porque el valor de Q se aproxima bastante a 1. Mientras que en la segunda revisión no presentan ningún nivel de ambigüedad, porque el valor de Q es 1. Con este resultado se puede decir que se tiene un documento de especificación de requisitos que presenta las características exigidas en su validación.

Fase III. Validación del modelo de casos de uso del sistema

En la última fase se validó el documento de casos de uso del sistema para corregir errores que pueda tener. Se aplicó la técnica de matriz de trazabilidad, prototipos de interfaz y métricas de CU con el objetivo de verificar la calidad requerida en el proceso de análisis realizado. Se verificó que cada requisito se vean reflejado en al menos un

Capítulo 4. “Construcción de la Solución Propuesta”

caso de uso, además de observar la completitud, consistencia y complejidad de los CU.

Aplicando la técnica matriz de trazabilidad se verificó cada requerimiento contra los casos de uso existentes, para lograr que todos los requisitos se encontraran en al menos un CU. En esta técnica se realizó 2 iteraciones, obteniendo en la primera que 3 requerimientos no se reflejaban en ningún caso de uso, ya en la segunda iteración fueron otro los resultados. Se detectó que todos los requerimientos identificados responden al menos a un CU del sistema y viceversa. A continuación se presenta la matriz resultante del análisis realizado (Ver Tabla # 4).

Requisitos Funcionales	Casos de Uso									
	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10
RF1	X									
RF2		X								
RF3			X							
RF4				X						
RF5					X					
RF6						X				
RF7							X			
RF8								X		
RF9									X	
RF10										X

Tabla 3 Matriz de Trazabilidad

Capítulo 4. “Construcción de la Solución Propuesta”

Una vez realizada la técnica anterior se procedió a la aplicación de la métrica de CU. Éste consta de cuatro atributos que se tuvieron en cuenta, los cuales son: Consistencia, Correctitud, Completitud y Complejidad. Dónde:

- ✚ **Completitud:** grado en que se han detallado los casos de uso más relevantes.
- ✚ **Correctitud:** grado en que las interacciones entre actor y sistema soportan el modelo de negocio.
- ✚ **Complejidad:** grado de presentación de los elementos que describe el contexto y la claridad del sistema.
- ✚ **Consistencia:** grado en que los casos de uso del sistema describen las interacciones entre el usuario y el sistema.

A continuación se muestra una tabla con los resultados de las métricas que deben cumplir los requisitos del software (Ver Tabla # 5).

Factor Completitud	Métrica Asociada	Sistema
Factor 7. ¿Están definidos todos los requerimientos que justifican la funcionalidad del Caso de Uso?	Métrica 7: Número de requerimientos omitidos por Caso de Uso. Umbral < 10%	Total de casos de uso: 10 Número de Caso de Uso con requerimientos sin definir: 0 Representa: 0%
Factor 8. ¿Existen requerimientos que no han sido considerados en algún Caso de Uso?	Métrica 9: Número de requerimientos que no son considerados en ningún Caso de Uso.	Total de RF: 10 Total de RF sin incluir en algún Casos de Uso: 0 Representa: 0%
Factor 13. ¿Todos los Casos de Uso han sido clasificados de acuerdo con su relevancia?	Métrica 15: Número de Casos de Uso que no han sido clasificados Umbral: < 10%	Total de Casos de Uso: 10 Número de Casos de Uso sin clasificar: 0 Representa: 0%

Capítulo 3. “Presentación de la Solución Propuesta”

Factor Consistencia	Métricas Asociadas	Sistema
<p>Factor 14. ¿El nombre dado a los Casos de Uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?</p>	<p>Métrica 16: Número de Casos de Uso que tienen un nombre incorrecto Umbral < 20%</p>	<p>Total de Casos de Uso: 10 Número de Casos de Uso que tienen un nombre incorrecto: 0 Representa: 0%</p>
<p>Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?</p>	<p>Métrica 25: Número de Casos de Uso complejos que no tienen separación del flujo básico y de flujos alternos. Umbral: < 20%</p>	<p>Total de Casos de Uso: 10 Número de Casos de Uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Representa: 0%</p>
Factor Correctitud	Métricas Asociadas	Sistema
<p>Factor 23: ¿Representa el Caso de Uso requerimientos comprensibles por el usuario?</p>	<p>Métrica 28: Número de Casos de Uso en que los requerimientos representados no son comprensibles por el usuario. Umbral: < 5%</p>	<p>Total de Casos de Uso: 10 Número de Casos de Uso en que los requerimientos representados no son comprensibles por el usuario: 0 Representa: 0%</p>
Factor Complejidad	Métricas Asociadas	Sistema
<p>Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p>	<p>Métrica 36: Número de elementos del diagrama que requieren reubicación. Umbral: < 30%</p>	<p>Total de Casos de Uso: 10 Número de elementos del diagrama que requieren reubicación: 0 Representa: 0%</p>

Tabla 4 Aplicación de métricas al Diagrama de Casos de Uso del Sistema.

Al aplicar estas métricas se evaluaron los factores completitud, consistencia, correctitud y complejidad de los CU del sistema. Se demostró que los casos de uso fueron descritos detalladamente, en ellos se evidencia el flujo normal y alterno que proporciona una mayor legibilidad. Todos los CU tenían un nombre correcto, además los requisitos son implementados en al menos en un caso de uso. Estos resultados

Capítulo 3. “Presentación de la Solución Propuesta”

permitió demostrar que el artefacto caso de uso del sistema posee la calidad deseada en la validación.

4.11 Conclusiones

En este capítulo se han mostrado las principales funcionalidades que podría tener el sistema a través de una serie de modelos, que garantizan un mayor entendimiento del mismo.

Se definieron las principales clases del análisis, el diseño e implementación mostrando sus relaciones así como la colaboración entre ellas mediante diferentes diagramas. Para lograr un diseño más legible se utilizaron algunos patrones que garantizan la claridad de los diagramas.

Se aplicaron las métricas de calidad a los artefactos generados llegándose a la conclusión, por los resultados obtenidos en las revisiones, que estos artefactos presentan poco nivel de ambigüedad, lo que trae consigo que tenga una buena calidad el análisis realizado.

Conclusiones Generales

Toda construcción de un software informático y la documentación pertinente conlleva a resultados. Es por ello, que guiados por el proceso de desarrollo de software RUP acompañado del lenguaje modelado UML, se realizó el Análisis y Diseño del SIGUCI para celulares, pudiéndose decir que la documentación creada pretende mejorar el tiempo de realización de este sistema.

Se concluye que:

- ✚ Se realizó el Análisis y Diseño del SIGUCI para celulares, plasmándose los requerimientos definidos.
- ✚ Se seleccionó RUP como metodología a seguir para desarrollar dicho producto, acompañado del lenguaje modelado UML porque permiten desarrollar artefactos que consolidan la comunicación mediante un único lenguaje entre todas las personas implicadas en el proyecto.
- ✚ Se creó un documento donde se recogió todo el proceso investigativo para la implementación del Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas basándose en la Plataforma gvSIG mini.

Recomendaciones

Una vez concluida la investigación, se recomienda:

- ✚ Se implemente el SIG_UCI a partir del diseño de clases propuesto.
- ✚ Se actualice la documentación en correspondencia con las nuevas necesidades que se planteen.

Anexos

Descripción de los casos de usos del sistema.

Caso de Uso:	Localizar_Persona
Actor:	Usuario_UCI
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar una persona en el mapa.
Resumen:	El caso de uso se inicia cuando el actor selecciona la opción de acceder al formulario de búsqueda y termina cuando se le muestra los datos asociados a la persona.
Precondiciones:	Debe existir al menos una capa visible y seleccionable.
Referencias	RF2
Prioridad	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona del menú principal la opción: - Localizar Persona.	2. El sistema le muestra al usuario una ventana para que seleccione uno de los diferentes criterios de búsqueda.
3. El usuario selecciona el criterio de búsqueda (usuario, solapín y Carnet de identidad) e introduce el valor según el criterio seleccionado. Presiona en el botón Localizar.	4. El sistema valida que sea correcto ,para ello debe cumplir las siguientes condiciones: Si el número de carnet introducido tiene más de 11 dígitos ir a la Sección “Carnet Incorrecto” . Si introduce caracteres extraños (“@_*-#.+) o letras(A-Z ,a-z) ir a la Sección “Caracteres Extraños o Letras” Si deja el campo vacío ir a la Sección “Campo Vacío” Si el número de solapín tiene más de 5 dígitos ir a la Sección “Solapín Incorrecto” . Si ha introducido mayúsculas ir a la Sección “Mayúsculas” .

Capítulo 3. “Presentación de la Solución Propuesta”

5. El usuario selecciona la acción de Localizar.	6. El caso de uso termina cuando el sistema muestra una ventana con la información de la persona (edificio donde vive, teléfono, apartamento) y su localización en el mapa marcado con una tachuela.
	7. El sistema devuelve el mapa centrado y vuelve a la interfaz principal.
Prototipo de Interfaz	
<i>Interfaz 5</i>	
Sección “Carnet Incorrecto”	
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje con el siguiente error: 'El número de carnet de identidad tiene 11 dígitos’.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 6</i>	
Sección “Caracteres Extraños”	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje de error informando “No se admiten caracteres extraños”.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 7</i>	
Sección “Campo Vacío”	

Capítulo 3. “Presentación de la Solución Propuesta”

Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje de error informando “Entre el número del edificio”.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 8</i>	
Sección “Solapín Incorrecto”	
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje con el siguiente error: 'El número de solapín debe tener 5 dígitos'.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	
<i>Interfaz 9</i>	
Sección “Mayúsculas”	
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje con el siguiente error: 'No debe entrar Mayúsculas'.
4.2 El actor selecciona el botón Aceptar a través del teclado del teléfono.	4.3 Vuelve a la acción 3 del flujo normal de eventos.
Poscondiciones	Se debe mostrar una ventana con los datos asociados a la persona.

Tabla 5 Descripción Textual CU Localizar Persona

Caso de Uso:	Mostrar_Información_Elemento.	
Actores:	Usuario_UCI	
Propósito	Este caso de uso tiene como objetivo permitir al actor acceder a la información de un elemento.	
Resumen:	El caso de uso se inicia cuando el actor selecciona la opción de ver la información de un elemento seleccionado en el mapa y termina cuando se le muestra la información de dicho objeto así como su localización en el mapa.	
Precondiciones:	Debe existir al menos una capa visible y el elemento haber sido seleccionado previamente por el usuario.	
Referencias	RF3	
Prioridad	Critico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor selecciona del menú principal la opción -Mostrar Información Elemento	2. El sistema le muestra al usuario una ventana con un listado para que seleccione el elemento (Docentes, Áreas Deportivas, Áreas Recreativas, Comedores, Plazas, Restaurantes, Consultorios, Cafeterías).	
3. El actor selecciona el elemento a través del teclado del teléfono y selecciona el botón Aceptar.	4. El caso de uso finaliza cuando el sistema muestra una ventana con la información del elemento (teléfono, Jefe de Area) así como su localización en el mapa pintado de otro color. Vuelve a la interfaz principal.	
Prototipo de Interfaz		
 Interfaz 3		
Poscondiciones	Se debe mostrar una ventana con la información del elemento.	

Capítulo 3. "Presentación de la Solución Propuesta"

Tabla 6 Descripción Textual CU Mostrar Información Elemento

Diagrama de Clases del Análisis

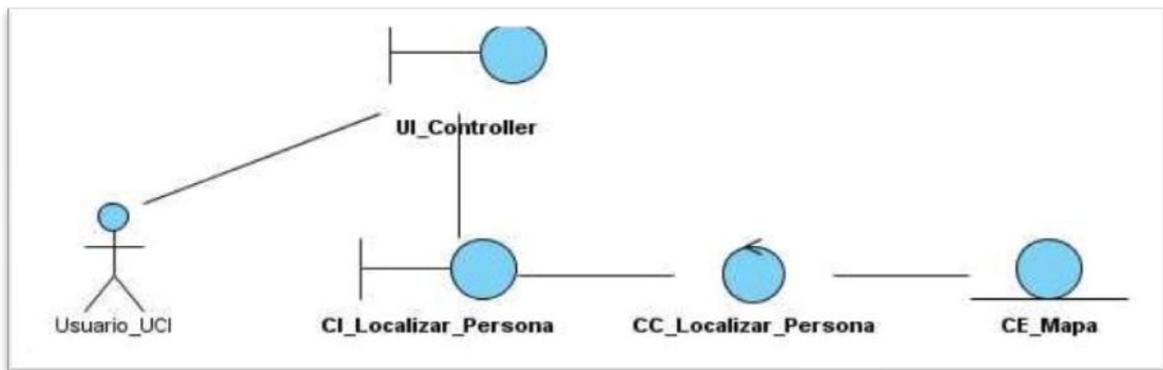


Figura 10 Diagrama de Clases del Análisis Localizar_Persona

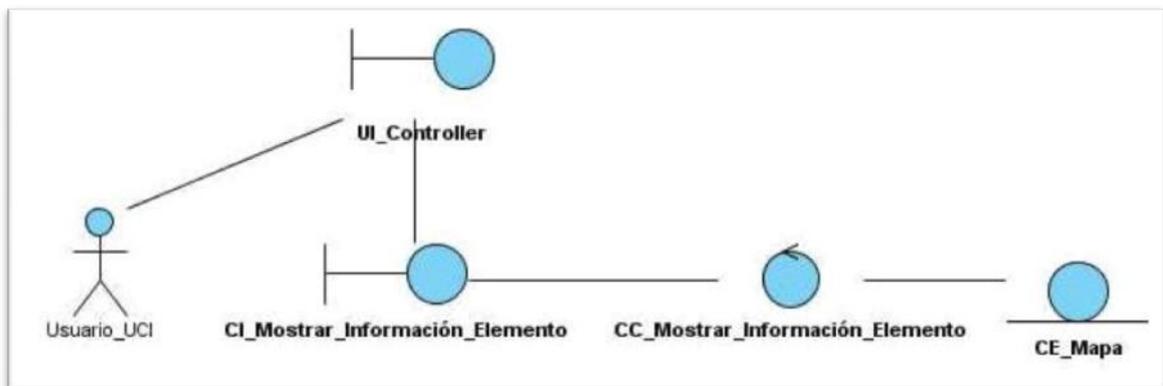


Figura 11 Diagrama de Clases del Análisis Mostrar_Información_Elemento

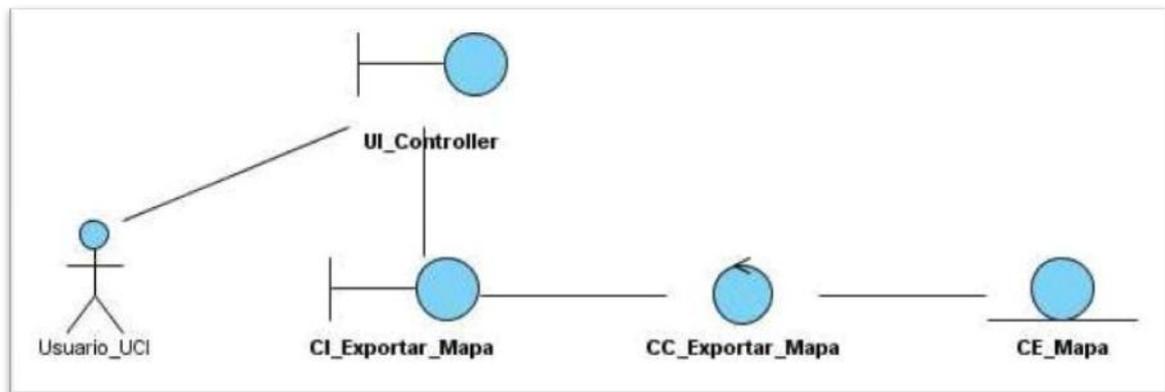


Figura 12 Diagrama de Clases del Análisis Exportar Mapa

Diagramas de Colaboración del Análisis

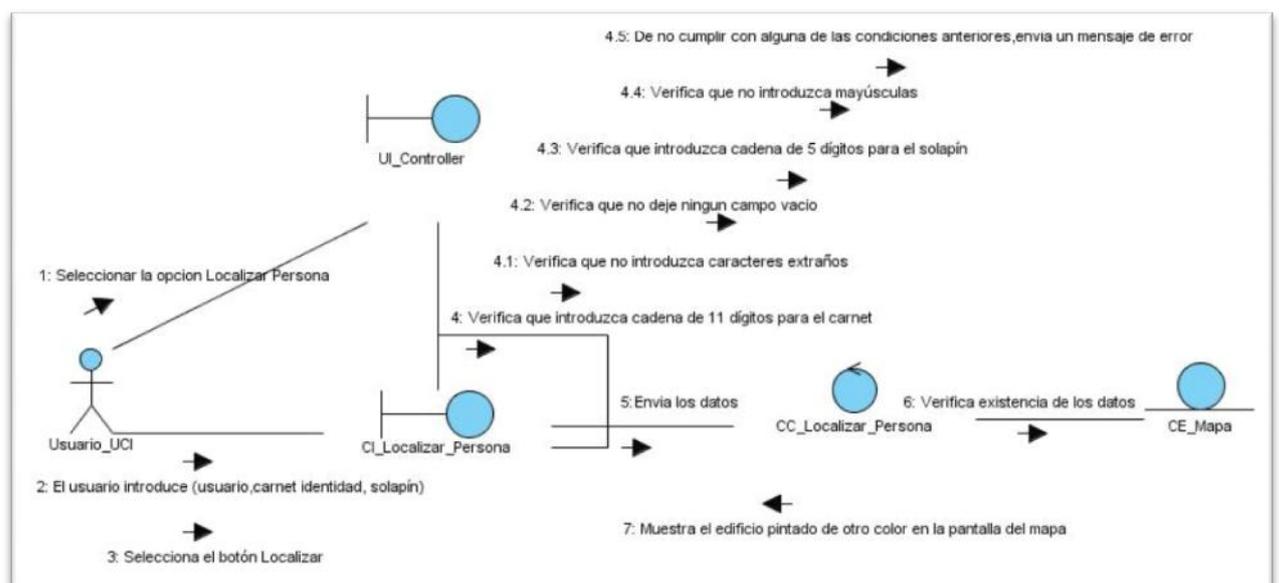


Figura 13 Diagrama de Colaboración del Análisis Localizar_Persona

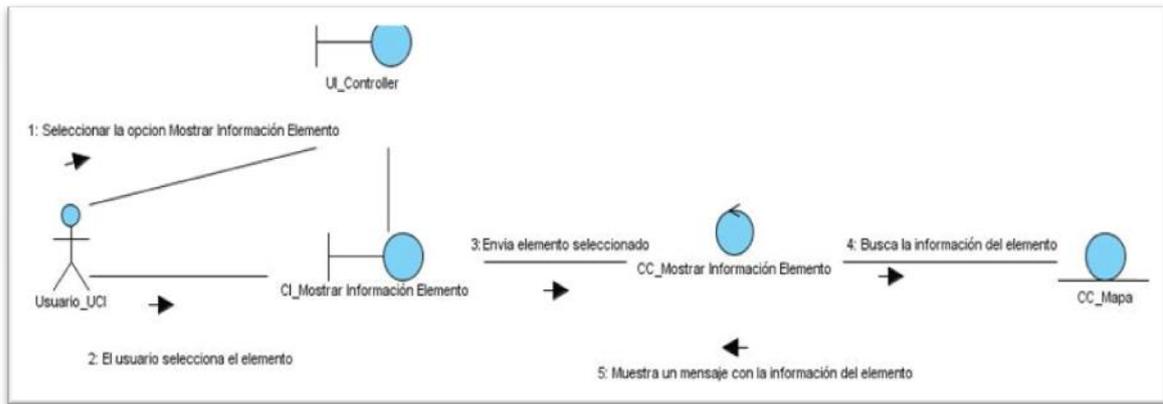


Figura 14 Diagrama de Colaboración del Análisis Mostrar_Información_Elemento.

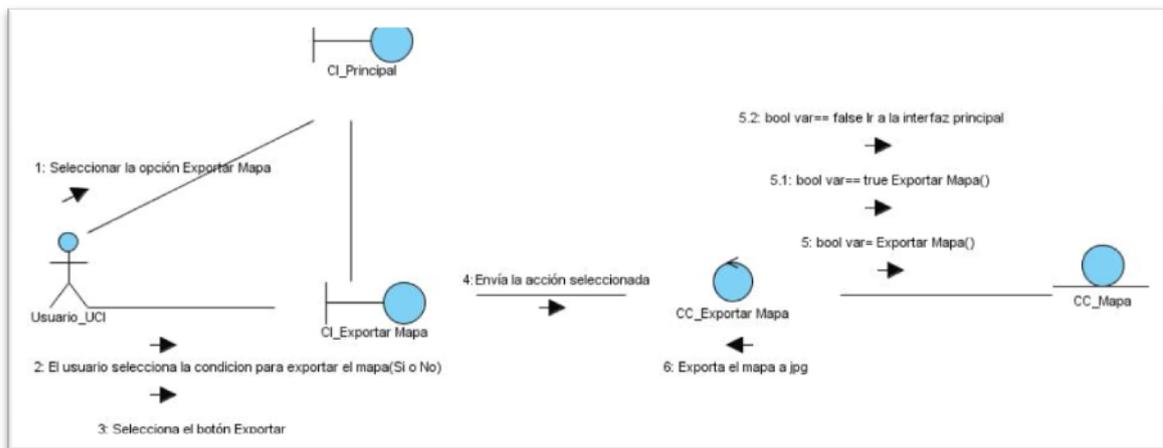


Figura 15 Diagrama de Colaboración del Análisis Exportar Mapa

Diagrama de Clases del Diseño

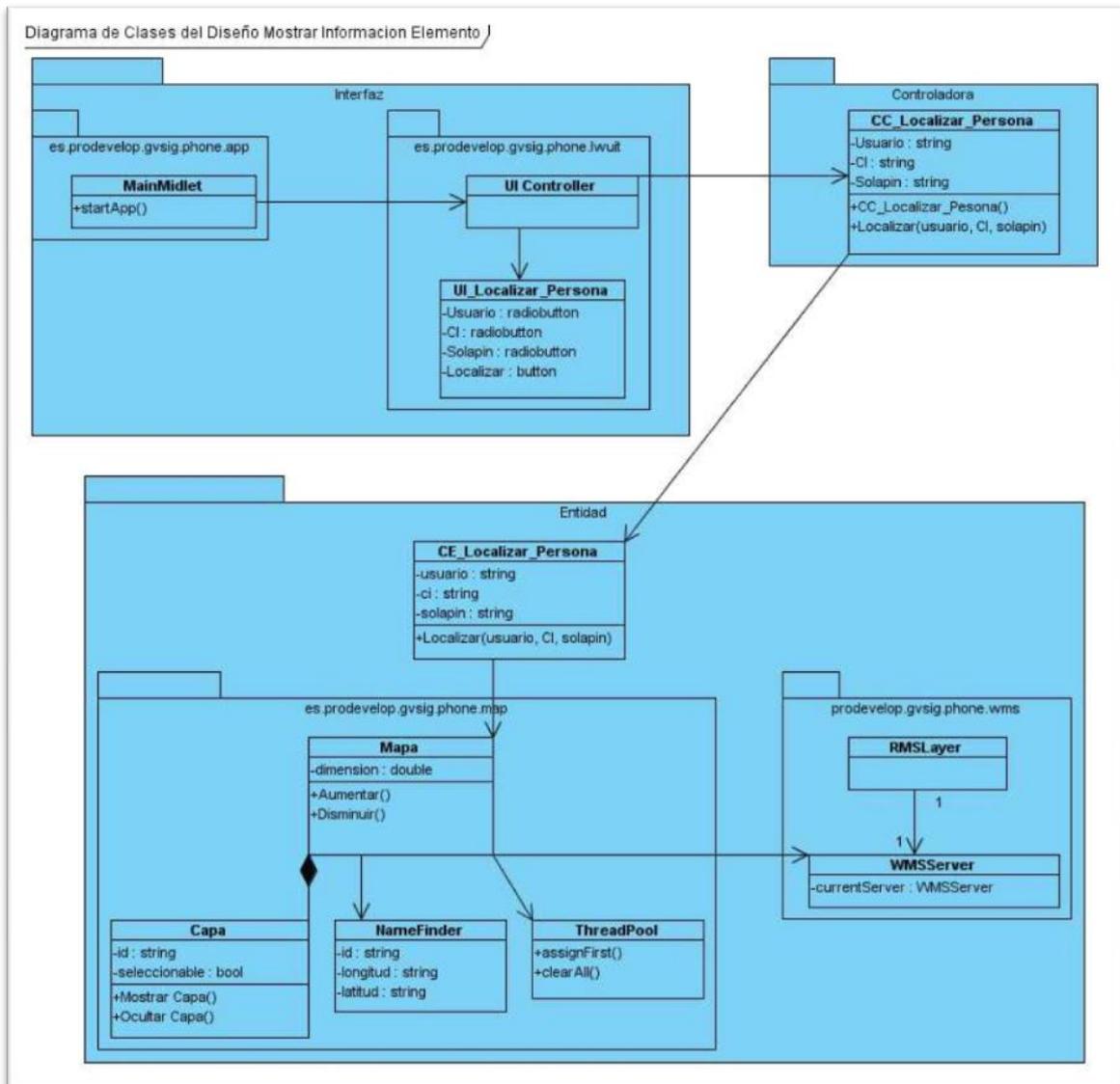


Figura 16 Diagrama de Clases del Diseño Mostrar_Información_Elemento.

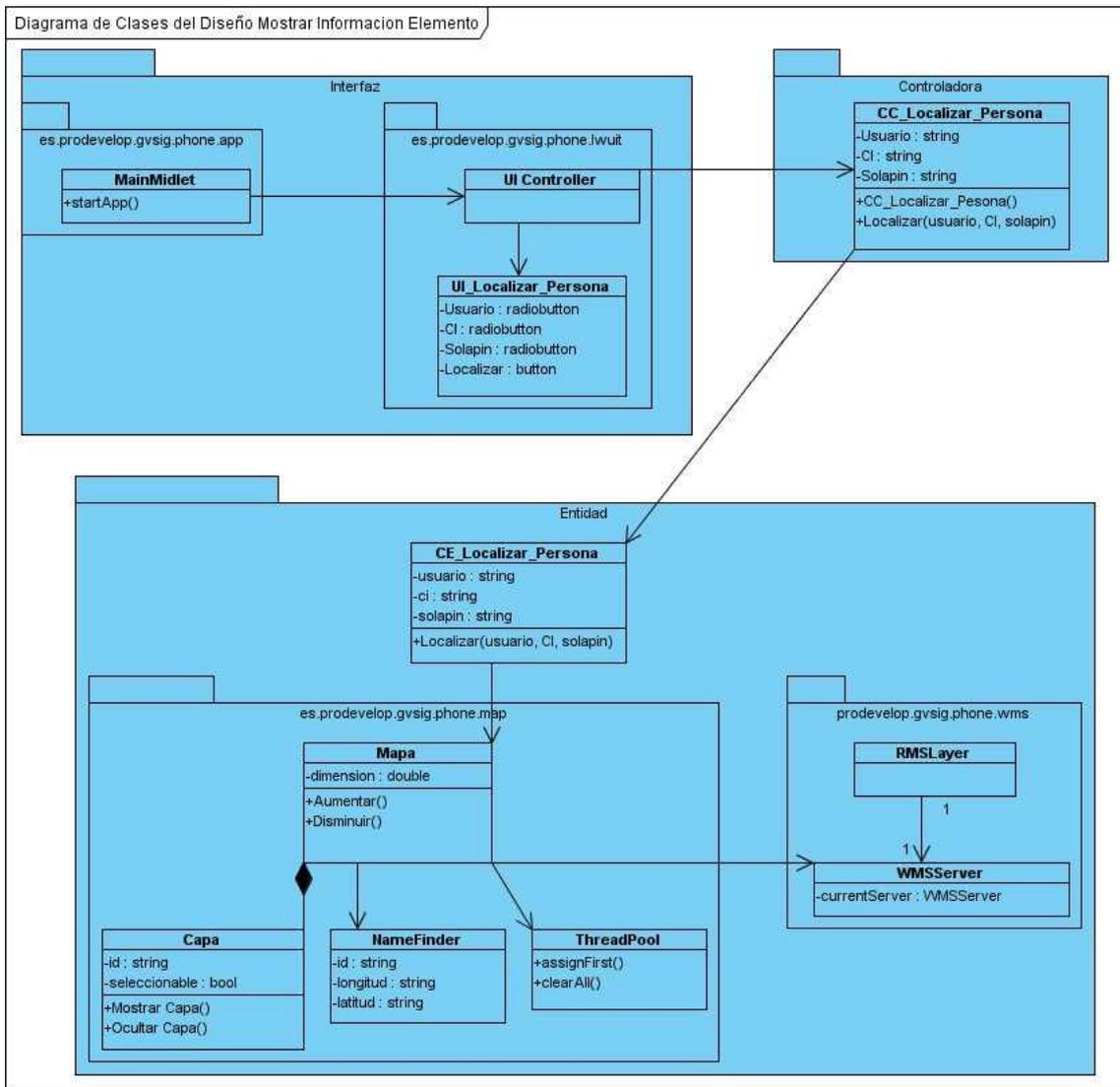


Figura 17 Diagrama de Clases del Diseño Localizar_Persona.

Bibliografía

1. Wikipedia. *La Enciclopedia Libre*. [En línea] [Citado el: 25 de Noviembre de 2010.] http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica.
2. Siiiau. [En línea] [Citado el: 25 de Noviembre de 2010.] http://www.siiiau.udg.mx/html/pronad/doctos/introduccion_siia.pdf.
3. Informática Moderna. [En línea] [Citado el: 2010 de Noviembre de 25.] www.informaticamoderna.com..
4. Wikiepedia. *La Enciclopedia Libre*. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://es.wikipedia.org/wiki/Mapa>.
5. Wikipedia. *La Enciclopedia Libre*. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://es.wikipedia.org/wiki/Georreferenciación>.
6. Bosque, 1992.
7. CartoMap. [En línea] http://www.cartomap.cl/utfsm/apuntes_sig.pdf.
8. Cartografía. *Arte y Ciencia de Trazar Mapas*. [En línea] <http://cartografia.supaw.com/>.
9. Sistemas de Información Geográficos. *Sistemas de Información Geográficos(Animales)*. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.rlc.fao.org/es/prioridades/transfron/sig/intro/compo.htm>.
10. Sistemas de Información Geográfica. *Sistemas de Información Geográfica*. [En línea] <http://www.rlc.fao.org/es/prioridades/transfron/sig/intro/compo.htm>.
11. Insituto de Investigación de Recursos Biológicos. [En línea] <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001..>
12. ESRI.
13. Wikipedia.
14. Beck, 1999.
15. Calderon, 2007.
16. Larman, Craig,1999.
17. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.sigte.udg.edu/jornadassiglibre2010/uploads/Articles/a41.pdf..>
18. IBM, 2010.

19. [En línea] http://es.wikipedia.org/wiki/Java_lenguaje_de_programacion..
20. Krutchén, 1999.
21. Padron Blanco, 2009.
22. Pressman, 2002.

Glosario de Términos

CLDC (Connected Limited Device Configuration): Es una configuración diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada como teléfonos móviles y asistentes personales.

Internet: Red de computadoras alrededor de todo el mundo que comparten información unas con otras por medio de páginas o sitios.

MIDP (Mobile Information Device Profile): Es el perfil para dispositivos de información móviles que combina con la configuración CLDC para proporcionar un entorno de ejecución para dispositivos móviles.

OGC (Open GIS Consortium): Consorcio encargado de definir los estándares a seguir por los SIG. Es un consorcio internacional formado por 256 empresas, organismos estatales y universidades, que participan en un proceso para el desarrollo de especificaciones de interfaces disponibles para el público en general.

SIG: Sistemas de Información Geográfica (GIS por sus siglas en inglés).

WMS (Web Map Services): Servicios de Mapas en Web. Especificación emitida por OGC.

Tile: Una tile es la parte gráfica de cada aplicación que puede ser utilizada para completar partes de un fondo por medio de un tileset (conjunto de tiles).