

Universidad de las Ciencias Informáticas

Facultad 6



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

**TÍTULO: Sistema de Información Geográfica para el Grupo Estatal
AZCUBA**

AUTOR: Aida Socarrás Carrasco.

TUTOR: Ing. Lisandra Escalona Griff.

CO-TUTOR: Ing. Yenier Jiménez Morales.

La Habana, Junio de 2013

"Año 55 de la Revolución"

Declaración de Autoría

Declaro ser autora del presente Trabajo de Diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Aida Socarrás Carrasco

Firma del Autor

Ing. Lisandra Escalona Griff

Firma del Tutor

Datos de Contacto

Nombre y Apellidos: Ing. Lisandra Escalona Griff

Correo Electrónico: lgriff.uci.cu

Año de Graduación: 2010

Profesión: Ingeniero en Ciencias Informáticas

Breve Descripción: Graduado en la Universidad de las Ciencias Informáticas. Actualmente es jefa de equipo de “Gestión de Proyecto” de la Línea de Productos de *Software* Aplicativos SIG.

Nombre y Apellidos: Ing. Yenier Jiménez Morales

Correo Electrónico: yimorales.uci.cu

Año de Graduación: 2010

Profesión: Ingeniero en Ciencias Informáticas.

Breve Descripción: Graduado en la Universidad de las Ciencias Informáticas. Actualmente es miembro de la Línea de Productos de *Software* Aplicativos SIG del Departamento Geoinformática del Centro de Desarrollo “Geoinformática y Señales Digitales” y desempeña el rol de implementador.

Agradecimientos

A mi mami Amarilys, porque le debo mi vida, por estar conmigo siempre, por ser lo más grande que tengo, mi razón de ser, mi ejemplo y por darme toda su confianza, comprensión y amor. Te quiero mamita.

A mimá, mi abuela que siempre cuidó de mí y hoy agradezco todo su amor y entrega.

A todos mis primos pequeños: Diana, Dayana, Delenis, Darianna, Dario, Darian, Daniel, Dairon, Adriana, Marlice, Glenda, Oriana, Érica, Kenny, Abrahan, Lazarito, Cintia, Jesús, Yandel, y Leyanis, que quiero muchísimo y se que ellos me adoran, sin ellos mi mundo sería pequeño, aunque cuando están juntos me vuelven loca.

A mi tía Maco, que fue mi segunda mamá y tuvo que soportar mis malcriadeces desde pequeña.

A mi prima Yanelis que ha sido la mejor persona del mundo, brindándome su cariño y ayuda incondicional cada vez que la he necesitado.

A toda mi familia por el gran apoyo que me proporcionó en todos estos años de estudios, A Claribel, Rambo, Duanny, Ronin, Bruno, Caridad e Ismaray que son otra familia para mí.

A Norlyn que me brindó muchísima ayuda, además del cariño de padre que nunca tuve.

A mi novio Marcos que me ha brindado su apoyo y ayuda en buenos y malos momentos.

A mi mejor amiga Disbey y a su nenecito Diuslen por todo su cariño, sus consejos, sus críticas y por apoyarme siempre.

A mis amigos del barrio por formar parte de mi infancia.

A mis amigos de la universidad que si ellos nada de esto tuviera sentido:

Una personita a la que quiero mucho que ha sido muy fuerte desde un principio y ha sabido superar todos los obstáculos que le ha impuesto la vida, Yanet, que me ha acompañado en el aula desde primer año y que me ha dado muy buenos consejos cuando los he necesitado.

Al piquete que ha mejorado (Grettel, Ari y Tahi) por las fiestas, las discusiones, las comidas en familia, y los buenos momentos.

A las niñas del 9203 y a las chicas de mi apto Susana, Blanca, Miosotis, Dailenka, la CHiki, Anita y Braña por los años de convivencia.

A unos chicos que quiero cantidad y que considero mis mejores amigos: Emir, karel, Asiel y Renier.

A mis compañeros de clase que no puedo mencionar porque no alcanzarían las hojas, ustedes saben que los tengo presente.

A mis tutores Lisandra, Yenier y Alain porque sin ellos no hubiera sido posible realizar este sueño, a mi oponente Mario, por ser tan dedicado y brindarme su ayuda y a los miembros del tribunal por encaminarme con sus consejos oportunos.

Por último pero no menos importante quiero agradecer, a mi bebecito Pirry, a la mejor compañía que una persona puede tener, mi perrito precioso, que a donde quiera que voy me sigue los pasos, que me ha hecho pasar malos ratos por atrevido que es, pero que me ha dado muchísimas alegrías.

Dedicatoria

A mi abuela Aida, por todo su amor.

A mi mamita, por ser madre y padre, darme toda su confianza, comprensión, amor y por apoyar mis decisiones en los momentos más difíciles.

A mis primos que forman una parte muy importante de mi vida.

A mi familia en general que siempre me ha apoyado.

Resumen

El Grupo Estatal AZCUBA que sustituye al Ministerio del Azúcar es el encargado de la producción de azúcar, agropecuaria, electricidad y derivados de la caña como la torula, miel de purga, bagazo, ron, entre otros derivados que suministra la misma. Los procesos asociados a la producción azucarera ya sean agrícolas o industriales como el cálculo de áreas, niveles de riego, acercamiento a centrales azucareros y molienda se manejan de una forma manual; de esta manera el trabajo es exhaustivo, se torna lento y dificulta la toma de decisiones. Debido a esto AZCUBA exige la incorporación de sistemas informáticos debidamente protegidos y asegurados, que sigan estándares informáticos de calidad, que apoyen la gestión, la información y sirvan de apoyo para la toma de decisiones.

En la Universidad de las Ciencias Informáticas específicamente en la Línea de Productos de *Software* Aplicativos SIG perteneciente al Centro de Desarrollo Geoinformática y Señales Digitales de la facultad 6 se despliegan Sistemas de Información Geográfica los cuales dan solución a diversos problemas, debido a su gran flexibilidad y adaptación a cualquier problemática de negocio. Se hace necesario el desarrollo de un Sistema de Información Geográfica para el Grupo Estatal AZCUBA centrándose en el mejor manejo de la información asociada a sus entidades en todo el país.

Palabras claves: Grupo Estatal AZCUBA, Sistema de Información Geográfica.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema	6
1.3 Objeto de estudio	8
1.3.1 Procesos de la agroindustria azucarera.....	8
1.3.2 Descripción de la situación problemática	9
1.4 Descripción general de los SIG.....	10
1.5 Soluciones existentes	11
1.6 Conclusiones parciales.....	14
Capítulo 2. Selección de Herramientas y Tecnologías	15
2.1 Introducción.....	15
2.2 Descripción de las tecnologías y herramientas a utilizar	15
2.2.1 Plataforma GeneSIG	15
2.2.2 Metodología de Desarrollo de <i>Software</i>	17
2.2.3 Herramienta de Modelado de <i>Software</i>	19
2.2.4 Sistemas Gestores de Base Datos	20
2.2.5 Servidores	21
2.2.6 Lenguajes de Programación a utilizar	22
2.2.7 Librería JavaScript.....	24
2.2.8 Otras Herramientas	24
2.3 Conclusiones Parciales	26
Capítulo 3. Presentación de la Solución Propuesta	27
3.1 Introducción	27
3.2 Modelo de Dominio.....	27

3.2.1	Diagrama de clases del Modelo de Dominio	27
3.2.2	Descripción del Modelo de Dominio.....	28
3.2.3	Definición de las clases del Modelo de Dominio	28
3.3	Especificación de Requisitos	30
3.3.1	Requisitos Funcionales (RF).....	30
3.3.2	Requisitos No Funcionales (RNF).....	32
3.4	Descripción del sistema propuesto	35
3.4.1	Descripción de los actores.....	35
3.4.2	Diagrama de Casos de Uso del Sistema (CUS).....	35
3.4.3	Descripción textual de los Casos de Uso del Sistema.	36
3.5	Conclusiones Parciales	38
Capítulo 4. Construcción de la Solución.....		40
4.1	Introducción	40
4.2	Patrones de Arquitectura.....	40
4.3	Modelo de Diseño	42
4.4	Principios de Diseño de Interfaz de Usuario.....	47
4.5	Diseño de la Base de Datos	48
4.6	Modelo de Implementación	50
4.7	Modelo de Despliegue.....	51
4.8	Diseño de Casos de Prueba.....	51
4.9	Conclusiones Parciales	63
Conclusiones		64
Recomendaciones		65
Referencia Bibliográfica		66
Bibliografía.....		68

Índice de Figuras

Figura 1.2: Interfaz de la herramienta: Portal de Información Herbert (HIP) tomado por (Cartografía.cl, 2005).	13
Figura 3.1: Modelo de Dominio.	28
Figura 3.2: Diagrama de Casos de Uso del Sistema.....	36
Figura 4.1: Diagrama de clases del diseño del CU Mostrar Información.	45
Figura 4.2: Modelo Relacional.	49
Figura 4.3: Diagrama de componentes.	50
Figura 4.4: Modelo de Despliegue.	51

Índice de Tablas

Tabla 3.1: Actores del Sistema.	35
Tabla 3.2: Descripción Textual del CU Modificar Información.	36
Tabla 4.1: Secciones a probar en el Caso de Uso.	54
Tabla 4.2: Descripción de variable	60
Tabla 4.3:Matriz de datos	61

Introducción

La evolución de la humanidad a través de la historia presenta etapas de crecimiento marcadas por los adelantos tecnológicos, científicos y de desarrollo social y económico. En los comienzos, el hombre en su necesidad de comunicarse y dejar plasmado todo lo acontecido en su historia se ha dedicado a desarrollar una variedad de técnicas destinadas a preservar, transmitir y proyectar sus conocimientos, las cuales han sido muy útiles para su desarrollo personal y social. Las Tecnologías de la Información y las Comunicaciones (TIC) conforman el conjunto de recursos necesarios para manipular dicha información utilizando los ordenadores y los programas informáticos necesarios para almacenarla, administrarla y transmitirla.

En la actualidad el mundo se ha visto en la necesidad de estar en constante transformación por el avance de las TIC puesto que las mismas se desarrollan continuamente. Numerosos son los logros que se han alcanzado hasta el momento de ahí que se puede decir que las TIC se han insertado en casi todas las ramas de la sociedad. Estos avances han permitido descubrir grandes tecnologías de las que se tienen acceso diariamente tales como: motores de búsqueda, redes sociales, comercio electrónico, televisión digital en línea y Sistemas de Información Geográfica.

Hoy en día la informatización es un elemento fundamental para elevar la calidad de vida de muchos países. Cuba es un país subdesarrollado en el cual se ha trabajado en procesos de transformaciones tales como: la sustitución de las máquinas de escribir por los ordenadores, el fax ahora es sustituido en gran medida por el correo electrónico y la autopista telefónica permite consultar, colaborar e interactuar con el resto del mundo. Todos estos procesos fueron realizados con el objetivo de alcanzar mayor desarrollo en el campo de la informática y las comunicaciones y para lograrlo se han destinado un conjunto de instituciones cubanas, siendo una de estas la Universidad de las Ciencias Informáticas (UCI) que tiene como uno de sus objetivos principales informatizar el país y desarrollar la industria del *software*.

La UCI cuenta con varios centros de desarrollo y producción de *software* que se encargan de dar cumplimiento a este objetivo. Uno de los centros que integran la universidad es el de Geoinformática

y Señales Digitales (GEYSED) de la Facultad 6 y dentro de éste, específicamente en el departamento Geoinformática, recayó la responsabilidad de asumir todo lo relacionado con el desarrollo de Sistemas de Información Geográfica (SIG), por lo cual desde hace algunos años se conformaron equipos de proyecto o Líneas de Productos de *Software* (LPS) que se encargan de desarrollar SIG sobre plataformas de *software* libre, sistemas que se distinguen por su impacto en el ámbito social y económico en las distintas empresas del país.

Los SIG constituyen un sistema integrado por *hardware*, *software* y procedimientos que posibilitan la realización de lectura, edición y almacenamiento, en términos generales, permiten la gestión de datos espaciales. También facilitan el análisis de los datos que puede incluir desde consultas sencillas hasta la elaboración de complejos modelos y puede llevarse a cabo tanto sobre la componente espacial de los datos o la componente temática, además de la generación de resultados tales como: mapas, informes, gráficos (Olaya, 2010). Por la necesidad de tener referenciadas determinadas variables existe un mayor uso de los SIG, de ahí que desde su creación hasta la actualidad se hayan convertido en una herramienta muy útil, ya que pueden emplearse en diversas actividades relacionadas con la tecnología de computadoras para integrar una gran variedad de datos capaces de crear una imagen de la geografía, mapas, medio ambiente y características socioeconómicas de una zona de estudio determinada.

La LPS Aplicativos SIG ha desarrollado sobre la Plataforma GeneSIG¹ varios sistemas destinados a diferentes empresas o instituciones del país, gracias a la amplia aplicación que presentan para cualquier esfera de la sociedad y dada la necesidad de informatizar cada vez más los procesos que se llevan a cabo en las distintas empresas o instituciones cubanas, las cuales deben manipular gran cantidad de información y aún no cuentan con las tecnologías necesarias. Un ejemplo de esto lo constituye el nuevo Grupo Estatal AZCUBA que surge cuando es disuelto el Ministerio del Azúcar (MINAZ) (Cuba, 2011).

El grupo azucarero que asumió gran parte del desaparecido ministerio tiene como objetivo orientar y controlar la producción de azúcar y derivados de la caña, así como proyectar su desarrollo,

¹**GeneSIG:** plataforma creada sobre tecnologías libres para el desarrollo de SIG. Fue desarrollada por acuerdo de colaboración entre la UCI, la unidad de las FAR y el Grupo Estatal GEOCUBA.

priorizando el análisis de algunos procesos tales como: el cálculo de áreas y acercamiento a los centrales, así como los niveles de riego, molienda y fertilizantes (Cuba, 2011).

La producción azucarera en Cuba se divide en dos sectores diferentes, agrario e industrial. El primero se encarga de todo el proceso en la siembra de caña, y el segundo de su procesamiento para lograr subproductos como el azúcar y sus derivados que satisfagan a la población. Para el desarrollo de los procesos vinculados a esta producción azucarera la información que se maneja es realizada de forma manual por profesionales calificados, que aunque los cálculos son confiables debido a la gran experiencia acumulada por los mismos, el procesamiento y comprensión de los datos generados demanda un coste en tiempo considerable, además muchas de las asignaciones de campos de caña a determinados centrales azucareros generan grandes pérdidas de combustibles debido a su lejanía y a las distribuciones de los caminos o líneas de ferrocarril, y sería más eficiente todo el proceso si se contara con un sistema que pudiera analizar la distancia de determinados puntos en relación a los centrales.

Esto provoca que los procesos asociados a la producción agroindustrial requieran de un exhaustivo trabajo, el cual se torna lento, genera pérdidas y dificulta la toma de decisiones. De ahí que se hace necesario para el grupo AZCUBA un mejor procesamiento de la información.

A partir del análisis de la situación problemática existente en el Grupo Estatal AZCUBA se define como **Problema a resolver** la siguiente interrogante: ¿Cómo contribuir a un mejor procesamiento de la información del Grupo Estatal AZCUBA?

Teniendo en cuenta el problema a resolver, el **Objeto de estudio** estará centrado en los procesos de la agroindustria azucarera y el **Campo de acción** estará enfocado en los procesos de la agroindustria azucarera desarrollados por el Grupo Estatal AZCUBA.

Para resolver el problema mencionado anteriormente se plantea como **Objetivo general**: Desarrollar un Sistema de Información Geográfica que permita procesar la información asociada al Grupo Estatal AZCUBA para facilitar la toma de decisiones.

Para dar cumplimiento al objetivo general se trazan las siguientes **Tareas de la investigación**:

- Caracterización de las tendencias actuales en cuanto al desarrollo de SIG tanto a nivel nacional como internacional.
- Caracterización de los principales procesos que se llevan a cabo en las entidades que pertenecen al Grupo Estatal AZCUBA.
- Fundamentación del uso de la metodología, tecnologías y herramientas necesarias a utilizar en el desarrollo de la solución.
- Identificación de las funcionalidades que debe brindar el sistema a desarrollar.
- Diseño de un sistema teniendo en cuenta los requisitos definidos.
- Implementación de un sistema que contemple todas las funcionalidades identificadas.
- Realización de pruebas al sistema para garantizar su correcto funcionamiento.

Se plantea como **Idea a defender**. La implementación de un Sistema de Información Geográfica para el Grupo Estatal AZCUBA permitirá un mejor procesamiento de la información referente al proceso agroindustrial azucarero facilitando así la toma de decisiones.

Como **Posibles resultados** se espera un:

- Sistema de Información Geográfica que permita acceder a la información socioeconómica y geográfica de las entidades que pertenecen al Grupo Estatal AZCUBA.
- Documentación ingenieril asociada al tema de investigación.

Métodos científicos de investigación utilizados

A continuación se hace mención de los métodos teóricos y empíricos que se utilizaron durante el desarrollo de la presente investigación.

De los métodos teóricos existentes se utilizaron: el **Analítico-Sintético**, el mismo se empleó para realizar un análisis de la documentación de los Sistemas de Información Geográfica y de los procesos asociados a la producción agroindustrial azucarera, recopilando toda la información disponible y sintetizando la misma para una mejor comprensión del tema; se utilizó además el método **Modelación** que permitió realizar los diagramas correspondientes según la metodología de desarrollo de *software* empleada. Dentro de los métodos empíricos existentes se utilizó la **Entrevista**, se aplicó

al realizarse encuentros con varios de los trabajadores pertenecientes al Grupo AZCUBA, con el fin de recopilar información acerca de los procesos asociados a la producción agroindustrial azucarera. Se realizó una entrevista cerrada a los 2 especialistas principales de informática y seguridad informática pertenecientes al Grupo AZCUBA. Para orientar la entrevista y hacerla más organizada se elaboró una serie de preguntas, con el objetivo de procesar fácilmente la información transmitida por los entrevistados. En el Anexo 2 se muestra la entrevista formulada.

Estructura de la Investigación

El presente Trabajo de Diploma consta de 4 capítulos, a continuación se describen cada uno de ellos de manera abreviada:

Capítulo 1. *Fundamentación Teórica:* en este capítulo se describen los conceptos fundamentales relacionados con el dominio del tema a examinar, además de hacer un análisis de soluciones existentes tanto nacionales como internacionales relacionadas con la industria azucarera.

Capítulo 2. *Tecnologías y herramientas a utilizar:* en el cual se explica el uso de las tecnologías, metodologías y herramientas necesarias a utilizar para desarrollar el Sistema de Información Geográfica.

Capítulo 3. *Propuesta de solución:* en este capítulo se presenta el modelo de dominio, los requisitos funcionales y no funcionales, así como la descripción de los casos de uso del sistema.

Capítulo 4. *Construcción de la solución propuesta:* en este último se exponen los diagramas de clases del diseño para cada caso de uso, el modelo de implementación y el de despliegue, así como también se realiza el diseño de casos de prueba y se realizan las pruebas a la aplicación.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En el presente capítulo se abordan temas que permiten evidenciar la fundamentación teórica de la presente investigación. La elaboración de un Sistema de Información Geográfica como solución a la gestión de la información requiere de un estudio profundo y un análisis de aquellos términos que ayuden al entendimiento del mismo de forma tal que tribute a un dominio sobre el tema a tratar. Se describe la situación problemática que dio origen a este trabajo, además de realizarse un profundo análisis de las soluciones existentes.

1.2 Conceptos asociados al dominio del problema

A continuación se describen algunos conceptos asociados al dominio del problema para la correcta comprensión de los temas que serán tratados en la presente investigación.

Tecnologías de la Información y la Comunicación (TIC): las TIC son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Abarcan un abanico de soluciones muy amplio. Incluyen las tecnologías para almacenar información y recuperarla después, enviar y recibir información de un sitio a otro, o procesar información para poder calcular resultados y elaborar informes. (serviciosTIC, 2006)

TIC, conjunto de herramientas informáticas y telemáticas que permiten la manipulación de información. Algunas de éstas no solo trabajan en el almacenamiento y organización de datos y contenidos, sino que facilitan el intercambio de los mismos y el diálogo que nace en el proceso de comprensión y acciones conexas a tales datos. (Guzmán, 2006)

Las TIC son un conjunto de herramientas informáticas para gestionar información y agrupan técnicas usadas en el tratamiento y la transmisión de la información, principalmente de informática, Internet y telecomunicaciones.

Capítulo #2 Selección de Herramientas y Tecnologías

Mapas: se llama mapa a toda representación de la superficie terrestre, o de una parte de ella, en una superficie plana. Es una representación reducida a escala del territorio, no aparece en este todos los detalles, sino solo los que se han considerado importantes o de interés (Urrutia, 2006).

Un mapa se define como una representación gráfica de la superficie de la tierra, o de parte de la misma, dibujada a escala en un plano. También se considera que es la representación geográfica en dibujo, de la superficie terrestre o de una parte de ella. Por lo común es una superficie plana representada en dos dimensiones, en la que se reúnen y analizan datos y medidas del lugar que se simboliza gráficamente en una escala reducida, de tal modo que las características de los terrenos se presentan claramente visible (Esparza, y otros, 2000).

De acuerdo a las definiciones anteriores un mapa es una representación reducida de la superficie de la Tierra.

Sistemas de Información Geográfica (SIG): los SIG son una nueva tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato (Llopis, 2006).

Un Sistema de Información Geográfica es un sistema de *hardware*, *software*, datos, personas, organizaciones y convenios institucionales para la recopilación, almacenamiento, análisis y distribución de información de territorios de la Tierra (Navarro, 2011).

Se considera que un SIG es la unión de diferentes componentes como son *hardware*, *software* y datos que posibilitan realizar análisis y manipulación de la información con el fin de resolver un problema específico; o sea, es un sistema creado a partir de un conjunto de datos, que poseen una componente geográfica, almacenados en una base de datos espacial, logrando la visualización a través de un mapa, ayudando así a realizar la toma de decisiones.

Dato Geográfico: Los datos geográficos son entidades espacio temporales que describen y cuantifican la distribución, el estado y los vínculos de los distintos fenómenos u objetos naturales o sociales (Quintero, 2010).

Una vez analizados los principales conceptos asociados al dominio del problema, dados por diferentes autores y asumiendo como autor de la investigación una valoración propia acerca de cada uno, se hace posible entonces proseguir con el desarrollo del presente Trabajo de Diploma haciendo una descripción detallada del objeto de estudio.

1.3 Objeto de estudio

El Objeto de Estudio de la presente investigación lo constituyen los procesos de la agroindustria azucarera, y para lograr un mejor entendimiento del mismo es necesario realizar un análisis de dichos procesos, así como también, una descripción detallada de la situación problemática.

1.3.1 Procesos de la agroindustria azucarera

La caña de azúcar es un cultivo de elevada relevancia a nivel mundial. A lo largo del siglo XX los ingenios azucareros impulsaron su desarrollo y mejoras en las tecnologías de cultivo. Actualmente se cosecha la caña de azúcar y luego se comienza su procesamiento. La caña es un cultivo de zonas tropicales o subtropicales del mundo. Requiere agua y suelos adecuados para crecer bien. Es una planta que asimila muy bien la radiación solar y es un cultivo eficiente ya que puede producir de 100 a 150 toneladas de caña por hectárea por año. Para la obtención de la caña de azúcar se realizan procesos agrícolas, y para la producción del azúcar y otros derivados se realizan procesos industriales, todos vinculados al sistema agroindustrial azucarero.

En el sistema agroindustrial azucarero uno de los procesos que se realizan es la cosecha de la caña. Es necesario saber cuántos trozos de tallo se necesitan en la tierra que se va a cosechar, a qué distancia se encuentra el terreno del central más cercano, además de otros aspectos; muy importante el cálculo de área y perímetro entre dos puntos determinados, a la hora de transportar la caña de un central a otro debido a algún daño ocasionado y en la fertilización de los suelos, esto disminuiría un gasto de combustible innecesario. La caña de azúcar es un cultivo que permanece en el campo durante los 12 meses del año, por consiguiente, requiere la aplicación complementaria de agua, al menos durante el período seco. La demanda de agua del cultivo varía mucho por lo que el suministro de agua de riego, debe ajustarse a los objetivos perseguidos con el cultivo: la mayor producción de azúcar al menor costo posible. Para la fabricación del azúcar una vez madura la planta, las cañas son

cortadas y se apilan a lo largo del campo, de donde se recogen a mano o a máquina, se atan en haces y se transportan al ingenio, que es un molino en el cual se trituran los tallos y se les extrae el azúcar; procesos que se hacen llamar sistema de corte, recogida y molienda.

1.3.2 Descripción de la situación problemática

El Grupo Estatal AZCUBA que sustituye al Ministerio del Azúcar responderá entre sus principales funciones por la producción de azúcar, agropecuaria, derivados de la caña y la electricidad. Entre las responsabilidades fundamentales de AZCUBA se guiará la presente investigación hacia la línea de: orientar y controlar la siembra de caña, producción de azúcar y derivados, priorizando el análisis de algunos procesos tales como: el cálculo de áreas y acercamiento a los centrales, los niveles de riego, sistema de corte, recogida molienda y fertilizantes (Cuba, 2011).

La planificación de la producción azucarera en Cuba se aplica al sector agrario en la plantación de la caña de azúcar y al sector industrial como parte del procesamiento de esta materia prima. En la planificación agrícola de la siembra de caña, los especialistas de AZCUBA para el desarrollo de los procesos vinculados a la producción azucarera trabajan de forma manual para lograr comprender determinadas variables cañeras sobre un terreno seleccionado. Los resultados son confiables debido a la gran experiencia acumulada a través de los años por los profesionales de este sector pero el procesamiento y comprensión de los datos generados demandan un coste en tiempo considerable. Como parte de este proceso es indispensable tener en cuenta varios valores de una misma variable para tomar una decisión, por lo que el cálculo es trabajoso.

El sector industrial tiene características propias que lo distinguen del agrícola pero no puede funcionar desvinculado del sistema agro-industria. Muchas de las asignaciones de campos de caña a determinados centrales azucareros generan grandes pérdidas de combustibles debido a su lejanía y a las distribuciones de los caminos o líneas de ferrocarril, y sería más eficiente todo el proceso si se contara con un sistema que pudiera analizar la distancia de determinados puntos en relación a los centrales. AZCUBA exige la incorporación de sistemas informáticos debidamente protegidos y asegurados, que sigan estándares informáticos de calidad, que apoyen la gestión, la información y sirvan de apoyo para la toma de decisiones.

1.4 Descripción general de los SIG

El desarrollo que la tecnología ha alcanzado y el gran valor de la información, exigen una mayor eficacia a la hora de tomar decisiones. En varias ramas de la sociedad los SIG se han convertido en una herramienta primordial para investigadores y especialistas, ya que están enfocados a visualizar y analizar geográficamente información o datos particulares de cada sistema.

Hoy en día, aproximadamente un 70 % de la información que se maneja en cualquier tipo de disciplina está georeferenciada. La utilización de cartografía ha dado un vuelco radical en el plazo de unas décadas, permitiendo nuevas posibilidades y acercando la información cartográfica como herramienta de primer orden a un público amplio y diverso. En una sociedad donde la información y la tecnología son dos de los pilares fundamentales, los SIG son, sin lugar a dudas, la tecnología estándar para el manejo de información geográfica, y los elementos básicos que canalizan la gestión de todo aquello que, de un modo u otro, presente una componente geográfica susceptible de ser aprovechada (Olaya, 2010).

A partir de las revisiones bibliográficas realizadas se puede resumir que los SIG brindan gran cantidad de información y mejoran la toma de decisiones de diferentes empresas en distintos campos de la actividad humana. Son utilizados en redes de infraestructura de servicios públicos, tales como agua potable, desagüe, electricidad, teléfono, transporte, pesca, y agricultura. En proyectos u obras de ingeniería como estudios para localización de presas, centrales hidroeléctricas, en las cuales se hacen estudios integrales de cuencas hídricas, levantamientos topográficos y análisis de pendiente.

Componentes de un SIG

La composición de los SIG está dividida en cinco elementos principales que se contemplan tradicionalmente en este aspecto:

- Datos: Constituyen la materia prima necesaria para el trabajo en un SIG y son los que contienen la información geográfica vital para la propia existencia del sistema.
- Métodos: Un conjunto de formulaciones y metodologías a aplicar sobre los datos.

- *Software*: Es necesaria una aplicación informática que pueda trabajar con los datos e implemente los métodos anteriores.
- *Hardware*: El equipamiento necesario para ejecutar el *software*.
- *Personas*: Son las encargadas de diseñar y utilizar el *software*, siendo el motor del sistema SIG (Olaya, 2010).

En el anexo 7 se encuentra la Figura 1 donde se muestran los componentes de un SIG y se evidencia cómo están interrelacionados.

Modelos de representación

Los SIG funcionan con dos tipos diferentes de información geográfica: el modelo vectorial y el modelo ráster. En líneas generales se puede decir que el modelo ráster se basa en una división sistemática del espacio, caracterizándolo como un conjunto de unidades elementales. El modelo vectorial, por su parte, no divide el espacio completamente, sino que lo define mediante una serie de elementos geométricos con valores asociados, siendo la disposición de estas no sistemáticas, sino guardando relación con los objetos geográficos presentes en la zona de estudio (Olaya, 2010).

Conociendo lo que es un SIG y sus principales características se hace necesario entonces estudiar algunas soluciones existentes relacionadas con la problemática planteada.

1.5 Soluciones existentes

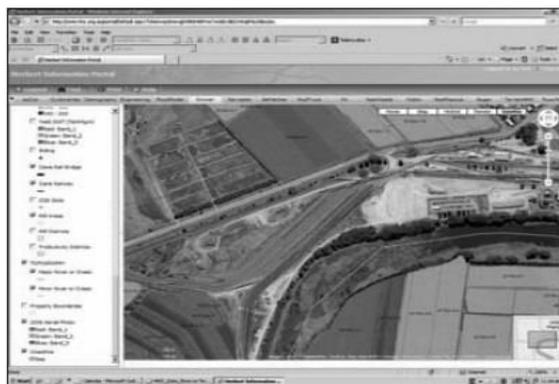
Con el desarrollo de las TIC se ha mejorado la calidad de trabajo de muchas empresas, algunas de ellas gestionan la información mediante sistemas informáticos que facilitan la toma de decisiones y permiten una mayor organización. Hoy en día, en el mundo y en Cuba uno de los sistemas más utilizados por las empresas son los Sistemas de Información Geográfica, por lo que se mencionarán algunos de los que están vinculados directamente con la producción de azúcar.

Capítulo #2 Selección de Herramientas y Tecnologías

- **Portal de Información Herbert (HIP):**

El Centro de Información de Recursos Herbert (HRIC) es una organización sin fines de lucro de Australia que apoya al desarrollo sustentable de la industria de la caña de azúcar en la zona. HRIC también es responsable de la construcción de redes comunitarias entre la industria local, gobierno y residentes. Para un mejor análisis de la caña de azúcar local y la distribución de sus resultados, HRIC lanzó el Portal de Información *Herbert* (HIP), una colaboración con el *software* SIG de ESRI, que se ha convertido en un sistema estatal (Cartografía.cl, 2005).

HIP se ha transformado en un sistema estatal basado en *ArcGIS* de ESRI²*ArcGIS Server*³ y *Desktop* y extensiones. El SIG está basado en la *web* para que los socios HRIC y los agricultores locales puedan acceder fácilmente a la información. HIP es compatible con la mayoría de los procesos críticos de negocio en la cadena de la región de suministros en la industria azucarera. Los mismos incluyen la cartografía y gestión de caña, en tiempo real con seguimiento de caña cosechadora, ferrocarril Sucrogen, seguro con integración y sistemas de monitoreo del rendimiento de la caña. Estas aplicaciones utilizan los SIG para promover la eficiencia, la productividad y mejorar los resultados medioambientales de los socios y los productores de caña de azúcar (Cartografía.cl, 2005).



²*Environment al Systems Research Institute* empresa fundada por Jack Dangermond en 1969.

³ Plataforma completa preparada para el despliegue, publicación y consumo de aplicaciones y servicios basados en Sistemas de Información Geográfica.

Figura 1.2: Interfaz de la herramienta: Portal de Información Herbert (HIP) tomado por (Cartografía.cl, 2005).

- **Sistema Experto**

Investigadores de la Uach⁴ diseñaron un Sistema Experto, que analiza las condiciones ambientales de suelo y de nutrientes, ofrece recomendaciones de aplicación para la producción de caña de azúcar y logra incrementos de producción. La administración de la información es mediante Sistemas de Información Geográfica, que es un *software* para gestionar la base de datos, la cual crea una interfaz que permite a los usuarios administrar una aplicación específica mediante el uso de equipo informático. La aplicación de un Sistema de Información Geográfica es beneficiosa ya que cobra auge en la producción agrícola y en la protección de los recursos naturales, además permite elaborar un manejo integral de los suelos cañero(imta, 2005).

- **Aplicación de un sistema geoespacial de soporte a la toma de decisiones para un desarrollo sostenible y ecológico en la agricultura cañera cubana**

El Ministerio del Azúcar se encontraba enfrascado en un arduo proceso de ordenamiento territorial de sus áreas, para lo cual es necesaria la utilización de la información catastral, la cual es potencialmente útil para muchos propósitos. Por otra parte el INICA⁵ ha alcanzado cierta experiencia en la utilización de la tecnología de Sistemas de Información Geográfica como base para la toma de decisiones concernientes a evaluación, uso y manejo de la agricultura cañera.

Es por ello que cumple el objetivo de desarrollar e implantar un sistema Geoespacial que a través de información temática sobre suelo, cultivos, relieve, hidrología, vegetación e infraestructura, sirva de soporte a la toma de decisiones para un manejo agronómico ecológico y de una agricultura sostenible (Perez, 2009).

⁴Universidad Austral de Chile.

⁵Instituto Nacional de Investigaciones de la Caña de Azúcar.

Análisis de las soluciones existentes

Se detalló a nivel nacional e internacional el análisis de las soluciones existentes con el objetivo de tener una referencia para la realización de este trabajo, ya que podrían ofrecer una guía sobre las funciones que pueden ser desarrolladas en el mismo.

Se puede apreciar que existen algunos Sistemas de Información Geográfica que se especializan en el área de la agroindustria azucarera por ejemplo: el Portal de Información *Herbert* (HIP) es un sistema estatal basado en *ArcGIS*, la familia de *ArcGIS* al ser *software* propietario no sede su código fuente, puesto que habría que comprarlo para poder utilizarlo, además de que esta aplicación está realizada cumpliendo el negocio y las necesidades de la entidad de Australia, y el Grupo Estatal AZCUBA tiene su propia organización y sus propios procesos.

Por otra parte se encuentran el Sistema experto y Aplicación a un sistema geoespacial de soporte a la toma de decisiones para un desarrollo sostenible y ecológico en la agricultura cañera cubana, que aunque algunas de sus funcionalidades se pueden tomar como referencia para el desarrollo de la aplicación a desarrollar, ambas no mencionan la parte industrial de la producción, se basan solamente en los procesos agrícolas y se necesita que la aplicación para AZCUBA sea un sistema que maneje y procese tanto la información agrícola de la caña de azúcar como la producción industrial azucarera. Por lo tanto la presente investigación resulta la primera de su tipo, considerando que a pesar de que existen aplicaciones similares ninguna es tan abarcadora ni se adecua a las necesidades específicas de la empresa cliente.

1.6 Conclusiones parciales

Con la descripción y el análisis de los principales conceptos asociados al dominio del problema se logró un mayor entendimiento de este trabajo. Con el estudio de las soluciones existentes relacionadas con la solución planteada en la investigación, se determinó que a pesar de la existencia de algunos Sistemas de Información Geográfica relacionados con la producción azucarera, ninguno posibilita una solución factible. Es por ello que se requiere la implementación de un sistema sobre tecnologías libres que responda a las necesidades de AZCUBA.

Capítulo 2. Selección de Herramientas y Tecnologías

2.1 Introducción

A nivel mundial existe gran variedad tecnológica que brinda excelentes posibilidades para que diferentes empresas puedan dar soluciones efectivas a sus necesidades. Este capítulo está dirigido a la descripción de las herramientas y tecnologías que serán utilizadas en el desarrollo de los componentes que darán solución al problema científico de la presente investigación.

2.2 Descripción de las tecnologías y herramientas a utilizar

En la Línea de Productos de *Software* Aplicativos SIG para el desarrollo de Sistemas de Información Geográfica se utiliza la plataforma GeneSIG, esta es una herramienta creada por un grupo conformado por especialistas de GEOCUBA⁶, las FAR⁷, y la UCI, con el objetivo de proporcionar servicios de acceso a la información geográfica, para su consulta, análisis y visualización mediante una interfaz de usuario sencilla y de fácil manejo. La misma requiere de tecnologías y herramientas para su utilización, a continuación se describe la plataforma, y luego se describen las tecnologías y herramientas.

2.2.1 Plataforma GeneSIG

La plataforma “GeneSIG” debe su surgimiento y evolución al amplio desarrollo que han tenido los SIG a nivel mundial para la toma de decisiones a bajo costo y con un número apreciable de facilidades. Son muchas las empresas en el mundo que están inmersas en la búsqueda de mecanismos y soluciones que permitan mejorar los sistemas y las técnicas existentes, lo cual ha propiciado el nacimiento de instrumentos de este tipo.

GeneSIG está basado en tecnologías libres. Es un sistema web que permite ser personalizado y adaptado para desarrollar Sistemas de Información Geográfica, siendo capaz de soportar una amplia

⁶Grupo Empresarial de mapa y cartografía.

⁷ Fuerzas Armadas Revolucionarias.

gama de funcionalidades relacionadas con la gestión de datos espaciales por parte de usuarios especializados y nuevos consumidores de servicios de datos geográficos. Tiene además una dualidad funcional, que lo habilita para su utilización como una aplicación SIG, a través de la cual los usuarios pueden consumir, manipular y consultar bases de datos cartográficas digitales, de diversos formatos y orígenes, como por ejemplo: datos vectoriales, datos ráster, datos GPS⁸, bases de datos espaciales en formato PostGIS, Servicios Web de Mapas (WMS) y Servicios de Geometrías (WFS) (Águila, 2011).

Entre los objetivos fundamentales de esta plataforma se encuentran:

- Permitir la representación geoespacial de la información asociada a cualquier negocio que lo requiera.
- Proporcionar servicios de acceso a la información geográfica para su consulta, análisis y visualización mediante una interfaz de usuario sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados en tecnología SIG.
- Integrar la información socioeconómica existente (recursos humanos, activos fijos, entidades de servicios, lugares de interés, etc.) con la información geográfica asociada (Águila, 2011).

Entre las ventajas que ofrece se destacan:

- La portabilidad, pues permite que el cliente opere en varios Sistemas Operativos (SO) sin necesidad de modificar el código que se encuentra en el servidor.
- El procesamiento de los datos manejados en la lógica del negocio es implementado sobre el lenguaje de programación PHP⁹, lo que resulta provechoso a la hora de realizar una migración de SO.
- La implementación de una capa de Servicios Web para compartir información del sistema aumenta notablemente su extensibilidad.

⁸GPS: por las siglas de su traducción en inglés *Global Positioning System*, en español Sistema de Posicionamiento Global.

⁹PHP: *Hypertext Preprocessor*.

- Además, el diseño del negocio es abierto y altamente configurable, dando la posibilidad de extensibilidad de los datos manejados tales como: nuevas capas definidas por interés del usuario sin implicar la reprogramación directa del código (Águila, 2011).

El proceso de personalización de GeneSIG consiste en adecuar las funcionalidades de dicha herramienta a las condiciones necesarias para la creación de un SIG enmarcado en cualquier negocio. Un ejemplo de los sistemas que se han desarrollado sobre la Plataforma lo constituye el SIGMIC: Sistema de Información Geográfica para el Ministerio de la Informática y las Comunicaciones, el SIGUCI: Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas, el SIGONEI: Sistema de Información Geográfica para la Oficina Nacional de Estadística e Información, entre otros.

2.2.2 Metodología de Desarrollo de Software

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental, que ayuda a los desarrolladores a realizar un nuevo *software*. Sin el apoyo de una metodología, nunca se llegaría a satisfacer las necesidades de los clientes para los cuales se trabaja.

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del *software*. Las metodologías de desarrollo se clasifican en ágiles y pesadas (Cockburn, 2006).

Elegir la metodología para un equipo en un determinado proyecto es trascendental para el éxito del producto. Esta se puede clasificar en dos grupos: las metodologías tradicionales, que se basan en una fuerte planificación durante todo el desarrollo y las metodologías ágiles, en las que el desarrollo de *software* es incremental, cooperativo, sencillo y adaptado.

Se consideran metodologías ágiles a aquellas que poseen la capacidad de proveer respuestas rápidas y una gran adaptabilidad a los cambios. Esto provoca menos complicaciones de acuerdo con que se provee la documentación necesaria para dar solución al problema.

Se usa una metodología pesada cuando la definición de los requisitos se realiza a través de entrevistas con los clientes y se hace necesario la generación de una amplia documentación que guíe

al comprador sobre cómo fue desarrollado el producto. Esta metodología es conocida también como tradicional o robusta.

Proceso Unificado de Desarrollo

RUP es un proceso de ingeniería de *software* planteado por Kruchten en el año 1996 cuyo objetivo es producir *software* de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido. Cubre el ciclo de vida y desarrollo del *software*.

El mismo consta de cuatro fases o etapas:

- **Comienzo o Inicio:** se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requisitos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtienen uno o varios productos que han pasado las pruebas. Se ponen estos resultados a consideración de un subconjunto de usuarios.
- **Transición:** El producto ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de esta metodología contiene 9 flujos de trabajos, los cuales se dividen en, 6 de ingeniería y 3 de soporte, en el anexo 8 se encuentra la figura 2 que muestra estos flujos de trabajo.

RUP utiliza el Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) para preparar todos los esquemas de un sistema de *software*.

UML en su versión 2.0, es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema *software*. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelado visual

también ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. En resumen, el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del *software* (Departamento de Sistemas Informáticos y Computación, 2010).

2.2.3 Herramienta de Modelado de *Software*

Las herramientas de Ingeniería de *Software* Asistida por Computadoras, conocidas por sus siglas en inglés (CASE) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software*, reduciendo el costo de la misma desde el punto de vista de tiempo y dinero. Con estas herramientas es posible la modelación y documentación de los artefactos obtenidos. Estas herramientas tienen como característica principal que agilizan los procesos de desarrollo del *software* al intervenir en todos los aspectos de su ciclo de vida, al tratar de automatizar los procesos de generación del producto a partir de los modelos realizados.

Visual Paradigm 8.0

Visual Paradigm for UML (VP) está diseñado para facilitar el diseño de los diagramas y posibilitar generar documentación. VP es una potente herramienta CASE que posibilita generar el código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle *JDeveloper* y *JBuilder*, así como que integra el ciclo completo de desarrollo del *software* desde el análisis, el diseño, la construcción, las pruebas y el despliegue (Paradigm, 2011).

Entre las ventajas de *Visual Paradigm* se encuentran:

- Proporciona a los desarrolladores una plataforma que les permite diseñar un producto rápidamente y con la calidad requerida.
- Posibilita generar código y realizar ingeniería inversa para varios lenguajes de programación, C++, PHP, XML *Schema*.
- Facilita la interoperabilidad con otras herramientas CASE.
- Se integra con múltiples herramientas de desarrollo, como Eclipse, *Jbuilder*, NetBeans y Oracle *Jdeveloper* (Paradigm, 2011).

2.2.4 Sistemas Gestores de Base Datos

Los Sistemas Gestores de Base de Datos (SGBD) son *software* muy específico, encargados de servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Estos sistemas están compuestos por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consultas.

Los SGBD tienen como finalidad, manejar de una forma clara, sencilla y ordenada un grupo de datos que consecutivamente se convertirán en información, para una buena manipulación de los mismos. Deben cumplir con una variedad de objetivos tales como: abstracción de información, independencia, redundancia mínima, consistencia, seguridad, integridad, respaldo y recuperación, control de concurrencia y tiempo de respuesta (Martín, y otros, 2007).

PostgreSQL versión 8.4

Este es un Sistema de Gestión de Bases de Datos Relacionales Orientada a Objetos (ORDBMS), por sus siglas en inglés. Es una herramienta desarrollada por la Universidad de California, Berkeley. Se considera por sus características como uno de los gestores de bases de datos de licencia libre y código abierto más avanzado de la actualidad por contar con más de diez años de evolución. PostgreSQL utiliza el modelo cliente-servidor así como el multiproceso en vez del multihilo, lo que garantiza la estabilidad del sistema. Con este multiproceso en caso de fallos no se afecta el sistema del todo ya que solo se afecta dicho proceso y no los demás (PostgreSQL, 2013).

Entre las ventajas se encuentran:

- Es multiplataforma, puede utilizarse en sistemas operativos como Linux y Windows.
- Provee documentación muy bien organizada, detallada, pública y libre.
- Posee comunidades de desarrollo muy activas, en Cuba existe una comunidad de PostgreSQL.
- Soporte nativo para lenguajes muy utilizados como PHP, C, C++ y Python.

PostgreSQL es un poderoso sistema de base de datos de código abierto que funciona en los principales sistemas operativos. Es compatible con el almacenamiento de objetos binarios, incluyendo

imágenes, sonidos y videos. Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios concurrentes que puede permitir. Para añadir soporte a objetos geográficos se utiliza el módulo de PostGIS en su versión 1.5.

PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos relacional PostgreSQL para su utilización en Sistemas de Información Geográfica. Es *software* libre, con Licencia Pública *General* (GPL por sus siglas en inglés General Public License) y es un importante componente para los Sistemas de Información Geográfica y los proyectos de código abierto con componente espacial (Fornaris, 2011).

2.2.5 Servidores

Servidor de Mapas: *MapServer* versión 5.6

MapServer es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología *Web*. Este puede ser utilizado como una aplicación CGI¹⁰ o a través del acceso a la Interfaz de Programación de Aplicaciones (API) de *MapServer* que proveen las librerías *Mapscript* (Espinosa, 2000).

MapServer posee entre sus principales características una salida cartográfica avanzada que posibilita ejecutar y dibujar elementos según la escala, así como automatizar los componentes del mapa como son la leyenda, la barra de escala y el mapa de referencia. Otros elementos distintivos son que brinda soporte a los lenguajes de *script* y plataformas más reconocidas, también a formatos vectoriales como PostGIS y formatos ráster como JPG, PNG, GIF, entre otros, y es compatible con múltiples sistemas operativos (Mapserver, 2011).

Se decidió utilizar como servidor de mapas *MapServer* porque es de código abierto y por las ventajas que posee ya que ofrece la posibilidad de presentar los datos a otros usuarios empleando *Internet*. Es uno de los pocos productos que ofrecen soluciones a los proyectos que disponen de mapas en múltiples formatos, además es portable, gratuito y se utiliza en GeneSIG.

¹⁰CGI: *Common Gateway Interface*.

Servidor Web: Apache versión 2.2

Apache es un servidor *web* gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Es indiscutiblemente uno de los mayores logros del *software* libre. El servidor Apache se desarrolla dentro del proyecto *HTTP Server* de la *Apache Software Foundation*. Es altamente configurable, presenta bases de datos de autenticación y negociado de contenido.

Es el servidor *web* más utilizado en el mundo y esto es debido a sus características: robustez, rapidez, ser multiplataforma con versiones para Linux, MacOS, Unix, dispone de módulos para ejecutar PHP, Perl, etc.(Cobo, 2005).

2.2.6 Lenguajes de Programación a utilizar

JavaScript

JavaScript fue diseñado para ser un lenguaje de elaboración de *scripts* que pudiera incrustarse en archivos *HyperText Markup Language* (HTML). No es compilado, sino que, en vez de ello, es interpretado por el navegador. Es un lenguaje que fue creado para darle más dinamismo a las páginas *web*. Con este lenguaje, se puede dar más movimiento a una página *web* y lograr una verdadera interactividad con los usuarios(Maza, 2001).

Entre las principales ventajas de JavaScript se encuentran:

- Reducción de la carga del servidor: para realizar funciones simples como validaciones era necesario una respuesta del servidor y con el uso de JavaScript es posible desarrollar funciones del lado del cliente para realizar estas tareas.
- Gran usabilidad: al ser compatible con la mayoría de los navegadores es muy utilizado para desarrollar funciones y ayudar en la visualización de las páginas dinámicas.
- Fácil de aprender, rápido y potente: no es necesario poseer conocimientos de programación para comenzar a utilizar este lenguaje debido a que se puede visualizar directamente en el

navegador. Además, integra las propiedades de los exploradores para crear funciones muy potentes.

- No es necesario un entorno de desarrollo para programarlo, con archivos de tipo texto se logra la implementación de una función en JavaScript.
- Basado en programación orientada a objetos aunque no incorpora la creación de clases ni la herencia.

JavaScript es un lenguaje sencillo y fácil de aprender, el cual puede hacerse cargo de gran cantidad de funcionalidades del lado del cliente, liberando al servidor de esas responsabilidades. Es rápido y potente, ideal para implementar funciones dentro de la página *web*.

Preprocesador de Hipertexto (PHP)

PHP es el acrónimo de *Hypertext Preprocessor* y no es más que un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas *web* dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Se seleccionó PHP v5.3 como lenguaje de programación del lado del servidor porque el *framework* (ver *artefacto de modelo de diseño*) escogido para el desarrollo del sistema está construido sobre este lenguaje. Además PHP ha sido utilizado para desarrollar un servidor de mapas muy utilizado en los SIG, el servidor de mapas *MapServer* es un entorno de desarrollo de código abierto para la creación de aplicaciones SIG con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante tecnologías *web*, además de utilizar las librerías **Mapscript de PHP** que constituyen la vía de comunicación de las aplicaciones SIG con el servidor de mapas *Mapserver* (Espinosa, 2000).

2.2.7 Librería JavaScript

ExtJS versión 3.0

ExtJS en sus inicios era una de las tantas librerías desarrolladas para (YUI) abreviación de Interfaz de Usuario para *Yahoo*. Desarrollada por Jack Sloccum por su gran potencia y ventaja en el desarrollo de aplicaciones *web* en el 2007 sale como una Librería independiente o *framework*. Su principal utilidad se debe a la posibilidad de construir aplicaciones *web* mediante el lenguaje.

Entre las ventajas de ExtJS se encuentran:

- Integración con la mayoría de los navegadores: No es necesario atenerse al navegador que se utiliza para el desarrollo debido a que el código del *framework* ExtJS funciona en la mayoría de los navegadores: Firefox, Internet Explorer, Safari y Opera.
- Balanceada relación entre Cliente-Servidor: Con el uso de ExtJS se distribuye la carga de procesamiento entre el cliente y el servidor, permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- Eficiencia de la red: Con el uso de ExtJS se disminuye el tráfico en la red debido a que se puede especificar cuáles datos se desean enviar al servidor o qué datos necesita del servidor.
- Utilización de componentes predefinidos: Cuenta con un conjunto de componentes y API definidos para la creación de aplicaciones complejas.
- Posee comunicación asíncrona: Con el uso de ExtJS las aplicaciones pueden comunicarse con el servidor sin necesidad de estar sujetas a una acción del usuario, esto permite que el usuario no se aburra esperando por una información que la suministra el servidor. Con esto se actualizan los sectores necesarios (Sloccum, 2008).

2.2.8 Otras Herramientas

ExtDesigner

Es la herramienta que posibilita la creación de prototipos de interfaz de usuario a programadores y personas no expertas en el tema, debido a su fácil uso pues permite colocar un componente con solo

arrastrarlo y soltarlo en la ventana que se desea crear. La presencia de un diseñador gráfico como este promueve la difusión de su empleo al proveer una forma rápida de desarrollar.

PgAdminIII

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia *de código abierto*. Está escrita en C++ usando la librería gráfica multiplataforma *wxWidgets*, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y *Windows*. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como *Pervasive Postgres*, *EnterpriseDB*, *Mammoth Replicator* y *SRA Power Gres* (PgadminIII, 2008).

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar *scripts* programados y soporte para el motor de replicación Slony-I. La conexión al servidor puede hacerse mediante TCP/IP o *Unix Domain Sockets* y puede encriptarse mediante *Secure Socket Layer* (SSL) para mayor seguridad (PgadminIII, 2008).

Net Beans

No es más que un IDE multilenguaje completado y modular, es decir, con soporte para Java Standard Edition, gran cantidad de módulos de terceros (*plugins*), desarrollo *intuitivodrag-and-drop* y completa código. Además es gratis y de código abierto con una gran cantidad de usuarios y desarrolladores. NetBeans es una plataforma para construir aplicaciones completas para el cliente y permite crear ventanas, menús, barras de herramientas y otras acciones fácilmente. Permite desarrollar aplicaciones tanto de Escritorio, Web, *Mobile* o *Enterprise*, utilizando como lenguaje Java, C/C++, Ruby on Rails, PHP, Groovy, Phyton, JavaScript entre otros.

Características

1. Mejoras en el editor de código.

2. Soporte para Ruby, JRuby y Ruby on Rails.
3. Instalación y actualización más simple.
4. Características visuales para el desarrollo web.
6. Creador gráfico de juegos para celulares.
7. Mejoras para UML.
8. Soporte para PHP(Cerda, 2009).

2.3 Conclusiones Parciales

En el presente capítulo se realizó un estudio de las diferentes tecnologías y herramientas para el desarrollo de la aplicación, abarcando sus principales propiedades y usos, lo que ha permitido distinguir cuáles de ellas responden a las necesidades de dicha aplicación, para ello se determinó:

- La utilización de RUP como metodología de desarrollo con el objetivo de guiar todo el proceso y generar una amplia documentación que guíe como fue desarrollado el producto, permitiendo la descripción de artefactos y modelos basados en UML en su versión 2.0, sustentado sobre la herramienta CASE *Visual Paradigm v8.0*.
- La utilización de PostgreSQL como gestor de base de datos para el almacenamiento de la información, además de la utilización de PostGIS, módulo que añade soporte de objetos geográficos a la base de datos relacional para su utilización en el Sistema de Información Geográfica.
- La utilización del lenguaje de programación del lado del servidor PHP en su versión 5.3 y las librerías *Mapscript* de PHP que constituyen la vía de comunicación de las aplicaciones SIG con el servidor de mapas *MapServer*.

Capítulo 3. Presentación de la Solución Propuesta

3.1 Introducción

En el presente capítulo se realiza una descripción de la solución propuesta para esta investigación. Se aborda todo lo referente al modelo de dominio, en el cual se definen los conceptos de las clases pertenecientes al mismo. Se hace un levantamiento de requisitos, especificando detalladamente los funcionales y los no funcionales, además de una descripción de los actores y los casos de uso del sistema.

3.2 Modelo de Dominio

El modelo de dominio representa la realidad de las interacciones entre los objetos del proyecto. Este es un diagrama que está compuesto por los objetos que se identifican como reales que se relacionan con el proyecto que se desea desarrollar y las relaciones existentes entre los mismos. La importancia del modelo de dominio es que ayuda a comprender mediante un diagrama de clases los conceptos más significativos sobre cómo es en realidad el sistema, sus relaciones y sus negocios.

3.2.1 Diagrama de clases del Modelo de Dominio

A continuación se muestra el Modelo de Dominio en la Figura 3.1 y posteriormente se realiza una breve descripción del mismo.

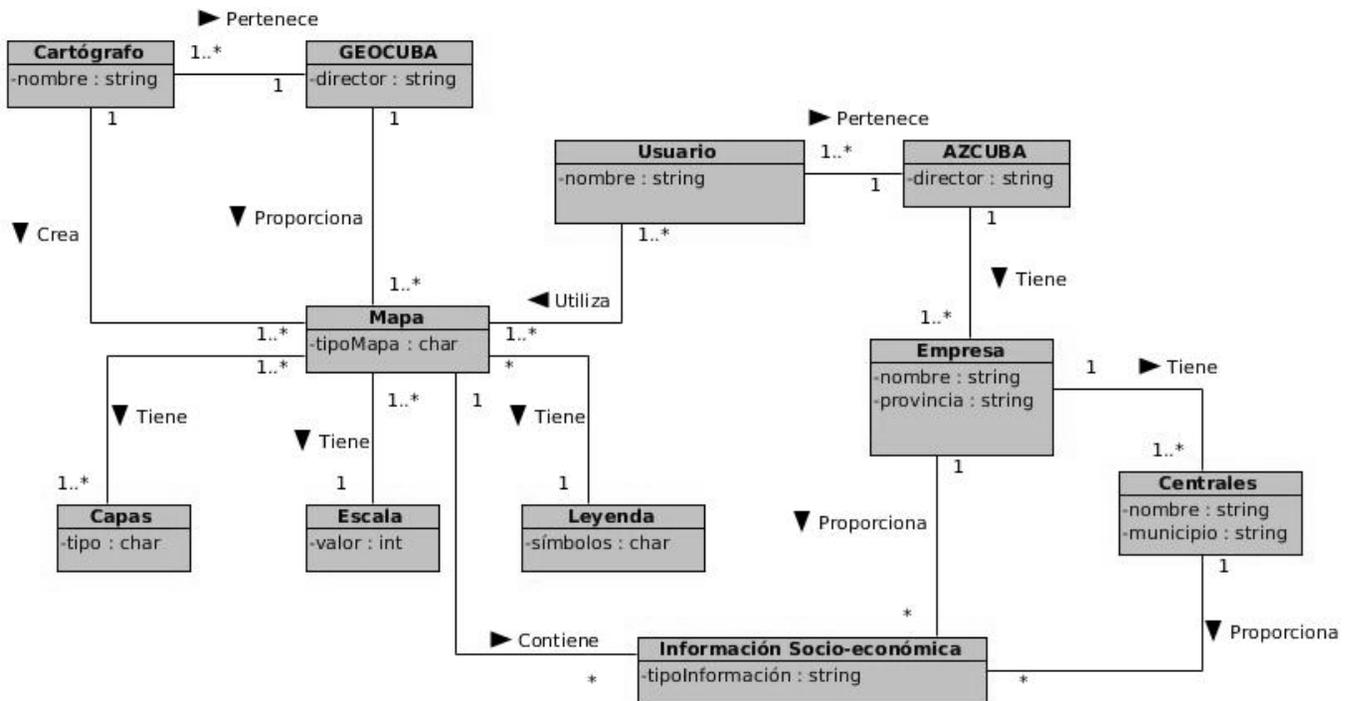


Figura 3.1: Modelo de Dominio.

3.2.2 Descripción del Modelo de Dominio

Los cartógrafos, pertenecientes al grupo GEOCUBA, son los encargados de la elaboración de los mapas, los cuales se componen de al menos una capa, y pueden tener una y una leyenda. Estos a su vez contienen información socio-económica de las empresas provinciales que integran al Grupo Estatal AZCUBA, así como también de los centrales pertenecientes a cada una de estas empresas. El mapa es utilizado por el usuario que representa a los trabajadores de AZCUBA.

3.2.3 Definición de las clases del Modelo de Dominio

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

Capítulo #3 Presentación de la Solución Propuesta

Cartógrafo: es la persona que se dedica profesionalmente a la realización de cartas geográficas, al estudio y elaboración de mapas. Especialista que provee toda la cartografía y datos geográficos utilizados por el SIG.

AZCUBA: Grupo Estatal que sustituye al Ministerio del Azúcar, encargado de la producción de azúcar, derivados y electricidad. Organización que rige todos los procesos de esta índole y dirige su desarrollo orientado a un fin determinado.

Usuario: es la persona que trabaja en AZCUBA y necesita consultar algún tipo de información incluida en un mapa.

Empresa: instalación estatal perteneciente al Grupo Estatal AZCUBA destinada para realizar labores sobre la caña y los derivados a nivel provincial.

Centrales: También denominados como Unidad Empresarial de Base (UEB) Industria, que se subordina directamente a la empresa provincial correspondiente según su ubicación geográfica y que se encarga específicamente de la producción de azúcar.

Información Socio-económica: se le denomina al conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país. En este contexto sería un conjunto de datos ubicados en la Base de Datos del sistema que contiene la información socio-económica de cada una de las empresas azucareras en el país.

Mapa: es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto. Base mediante la cual se obtiene todo el flujo de datos necesario para realizar las funcionalidades del SIG para AZCUBA.

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

Escala: es la relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en

forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes. Es la relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa.

Leyenda: es la explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.

3.3 Especificación de Requisitos

La Especificación de Requisitos de *Software* (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones que tendrán los usuarios con el *software*. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios). Los no funcionales son requisitos que imponen restricciones en el diseño o la implementación (como por ejemplo restricciones en el diseño o estándares de calidad).

3.3.1 Requisitos Funcionales (RF)

Los RF constituyen capacidades o condiciones que el sistema debe cumplir para lograr un objetivo determinado. A partir de los encuentros sostenidos con los clientes se identificaron 25 requisitos que de acuerdo a su similitud o semejanza fueron agrupados en 11 módulos los cuales se muestran a continuación. (Ver el documento de Especificación de Requisitos de *Software* del expediente de proyecto).

- **Autenticación**

RF 1: Autenticar usuario.

RF 2: Cerrar sesión.

- **Administración**

RF 3: Adicionar usuario.

RF 4: Modificar usuario.

RF 5: Eliminar usuario.

- **Navegación**

RF 6: Acercar determinada región del mapa.

RF 7: Alejar determinada región del mapa.

RF 8: Visualizar todo el mapa.

RF 9: Modificar el centro del mapa.

RF 10: Visualizar el mapa anterior.

RF 11: Visualizar el mapa siguiente.

RF 12: Navegar a través del mapa de referencia.

- **Identificación**

RF 13: Realizar identificación puntual.

- **Análisis**

RF 14: Medir distancia.

RF 15: Calcular área y perímetro de una región.

RF 16: Crear mapa temático.

- **Localización**

RF 17: Localizar entidades de AZCUBA.

- **Visualización**

RF 18: Habilitar capas del mapa.

RF 19: Deshabilitar capas del mapa.

RF 20: Modificar la escala del mapa.

- **Impresión**

RF 21: Exportar mapa.

- **Servicio**

RF 22: Exportar los datos.

RF 23: Importar los datos.

- **Gestión**

RF 24: Modificar información.

- **Reporte**

RF 25: Generar reporte.

3.3.2 Requisitos No Funcionales (RNF)

Los RNF representan las propiedades o cualidades que el sistema debe tener. Estos son los requisitos que repercuten en cómo el usuario va a interactuar con el sistema. Es la mejor vía para que dicha aplicación sea confiable, rápida y con una interfaz amigable para las personas que utilizarán el producto.

- **Usabilidad**

RNF 1: La interfaz de usuario del sistema debe ser intuitiva, tener un alto nivel de usabilidad permitiendo que el personal perteneciente a AZCUBA tenga los permisos necesarios, un conocimiento básico en el manejo de computadoras y de información de la producción azucarera para que pueda utilizarlo sin problema. Además el diseño del sistema debe tener un etiquetado que resulte familiar al usuario que interactuará con él.

RNF 2: Las funcionalidades principales del sistema estarán orientadas a íconos para un mayor reconocimiento por parte del usuario.

- **Fiabilidad**

RNF 3: La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

RNF 4: El tiempo medio de reparación en caso de un fallo será de 7 días.

- **Eficiencia**

RNF 5: El tiempo de respuesta y la velocidad de procesamiento estarán dados por la cantidad de información a procesar. Entre mayor cantidad de información mayor será el tiempo de procesamiento, de actualización y recuperación.

- **Soporte**

RNF 6: La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

- **Restricciones de Diseño**

RNF 7: Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

RNF 8: El producto de *software* final debe diseñarse sobre una arquitectura cliente-servidor.

RNF 9: Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará sobre la Plataforma GeneSIG y que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

- **Hardware**

RNF 10:

Para las PC clientes:

- Se requiere tengan tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- El Servidor de base de datos debe tener como mínimo 2 GB de RAM y 40 GB de disco duro.
- Procesador 3 GHz como mínimo.

- **Software**

RNF 11:

Para las PC clientes:

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

Para los servidores:

- Sistema operativo GNU/Linux Ubuntu Server 11.04 o superior.
- Servidor *Web* Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión *pgsql* incluida.
- PostgreSQL 8.4 o superior como Sistema Gestor de Base de Datos.
- PostGIS 1.5.2 o superior como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.6 o superior, con extensión PHP Mapscript.

- **Interfaces de Comunicación**

RNF 12: El producto garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, así como elementos para cambiar las vistas, las escalas y las capas que serán visibles en la interacción.

RNF 13: La resolución de pantalla que debe ser usada es: 1024 x 768 píxeles.

- **Requisitos Legales, de Derecho de Autor y otros**

RNF 14: El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

RNF 15: La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

- **Estándares aplicables**

RNF 16: El sistema será desarrollado bajo estándares *OpenGIS* como aseguramiento de la parte científica y en el desarrollo se codificará y modelará siguiendo los patrones de las normativas *International Organization for Standardization* (ISO), tanto de codificación como de diseño de bases de datos.

Una vez que se han definido los requisitos funcionales y no funcionales del *software* a desarrollar, es necesario determinar los actores y casos de uso que tendrá el sistema.

3.4 Descripción del sistema propuesto

3.4.1 Descripción de los actores

Los actores del sistema constituyen agentes foráneos, es decir en un principio de desarrollo no se cuenta con ellos, son los que interactuarán con la aplicación solicitando y recibiendo información en aras de obtener un resultado. Seguidamente se muestra la Tabla 3.1 donde aparecen los actores que intervienen en el sistema y una breve descripción de cada uno.

Tabla 3.1: Actores del Sistema.

Actor	Descripción
Empleado	Trabajador del Grupo Estatal AZCUBA que utiliza las funcionalidades de la aplicación sin modificar información alguna sobre la misma.
Especialista	Personal autorizado, encargado de mantener actualizada la información manejada por el sistema.
Administrador	Personal autorizado, responsable de la gestión de los usuarios que van a poder acceder a la aplicación.

3.4.2 Diagrama de Casos de Uso del Sistema (CUS)

En el Diagrama de Casos de Uso del Sistema que se muestra en la Figura 3.2 se evidencia la relación existente entre los actores que interactúan con los casos de uso, así como también las relaciones entre estos.

Capítulo #3 Presentación de la Solución Propuesta

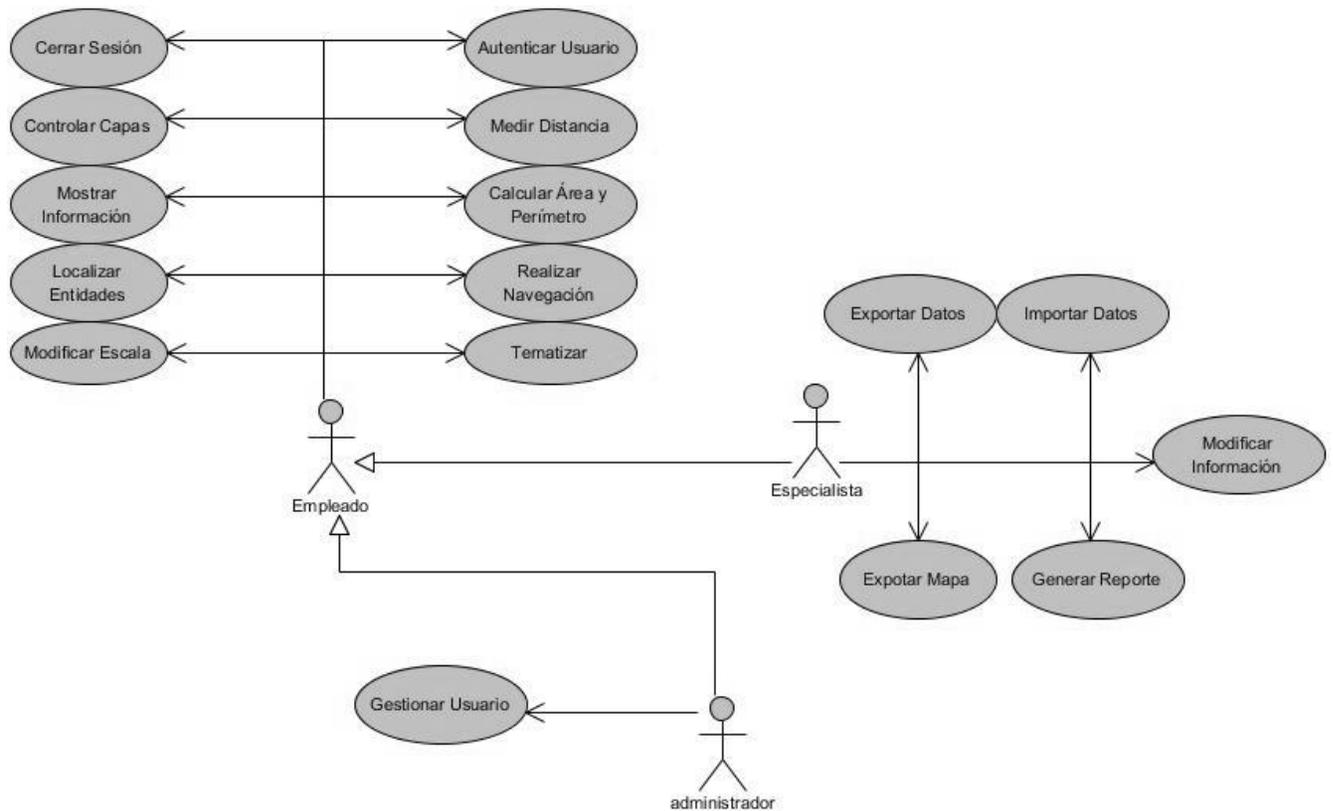


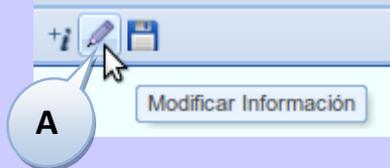
Figura 3.2: Diagrama de Casos de Uso del Sistema.

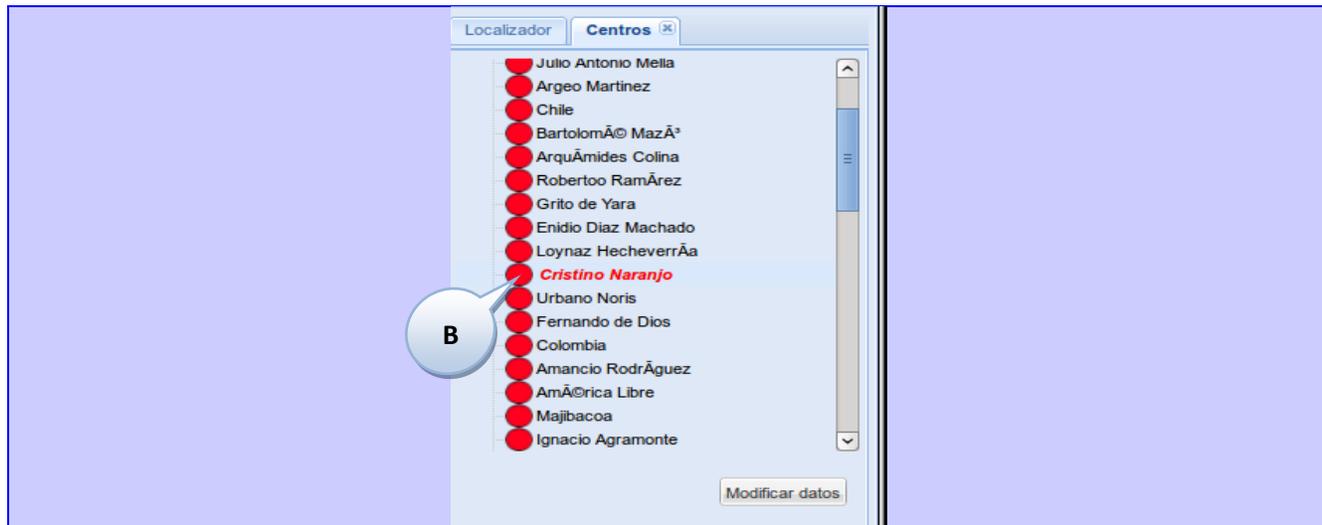
3.4.3 Descripción textual de los Casos de Uso del Sistema.

Tabla 3.2: Descripción Textual del CU Modificar Información.

Caso de Uso:	Modificar Información.
Actores:	Especialista
Propósito:	Este caso de uso se realiza con el objetivo de modificar la información de los centrales azucareros.
Resumen:	Este caso de uso se inicia cuando el especialista desea modificar la información de los centrales azucareros y termina cuando el sistema muestra un mensaje indicando que todos los datos fueron modificados correctamente.

Capítulo #3 Presentación de la Solución Propuesta

Precondiciones:	El especialista tiene que estar autenticado en la aplicación.
Referencias:	RF 24.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el especialista selecciona la opción Modificar Información (A) que aparece en la barra de herramientas. Ver interfaz 1.	2. El sistema muestra en el panel derecho todos los centrales azucareros para que el especialista seleccione el que desea modificar. Ver Interfaz 2 (B).
3. El especialista selecciona con un clic el central que desea modificar.	4. El sistema activa el botón "Modificar datos".
5. El especialista presiona el botón "Modificar datos".	6. El sistema muestra el formulario correspondiente para que el usuario introduzca los datos.
7. El especialista introduce los datos correspondientes a cada uno de los elementos y presiona el botón "Aceptar".	8. El sistema muestra un mensaje de confirmación.
9. El especialista confirma la modificación.	10. El caso de uso termina cuando el sistema guarda los datos.
Prototipo de Interfaz	
Interfaz 1	
	
Interfaz 2	



Flujos alternos

Acción del Actor	Respuesta del Sistema
3. El especialista cierra el panel "Centro"	4. El sistema oculta el panel derecho.
7. El especialista cierra el panel "Modificar centrales."	8. El sistema oculta el panel "Modifica centrales".
9. El especialista no confirma la modificación del central.	10. El sistema mantiene los datos que hasta ese momento tenía.
9. El especialista cierra el mensaje de confirmación.	10. El sistema mantiene los datos que hasta ese momento tenía.
Poscondiciones:	El sistema modifica la información asociada a los centrales.

Las descripciones correspondientes a los restantes casos de uso se encuentran en el artefacto Modelo del Sistema del expediente del proyecto y también en el Anexo 3 de este documento.

3.5 Conclusiones Parciales

En este capítulo se analizaron las características principales de la aplicación, así como las restricciones bajo las cuales debe funcionar, por lo que se puede arribar a las siguientes conclusiones.

- Se confeccionó el modelo de dominio que permitió un mejor entendimiento del problema.

Capítulo #3 Presentación de la Solución Propuesta

- La captura de los requisitos funcionales y no funcionales permitió definir las funcionalidades del sistema, además de enmarcar las restricciones de la aplicación.
- A partir del levantamiento de requisitos se diseñaron los casos de uso y se realizó la descripción de los mismos, permitiendo que se identificaran los casos críticos y se entendiera en profundidad el resultado final. Una vez realizado esto se tienen las bases para comenzar a construir el producto final.

Capítulo 4. Construcción de la Solución

4.1 Introducción

En este capítulo se presentan los elementos fundamentales relacionados con el diseño del sistema. Se realizan las descripciones de los diferentes patrones arquitectónicos y de diseño, así como la confección de los diagramas de clases del diseño para cada caso de uso identificado en el Modelo del Sistema. Se realiza el diseño de la base de datos, así como la representación de las tablas con las que trabajará el sistema. Como parte del modelo de implementación se elabora el diagrama de componentes del sistema y el diagrama de despliegue. Además, se realizan los diseños de casos de prueba que permitirán validar la solución propuesta.

4.2 Patrones de Arquitectura

Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Este define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones (Pressman, 2006).

Los patrones se categorizan según la escala o nivel de abstracción, sin embargo, cada una de las categorías de patrones define un mismo nivel de abstracción o escala de aplicabilidad (Pressman, 2006).

Para construir cualquier sistema es importante tener en cuenta los patrones de arquitectura existentes para poder determinar cuál o cuáles son los más convenientes utilizar en dependencia de las características que debe poseer la aplicación a desarrollar.

Los patrones de arquitectura de *software* son patrones de diseño que constituyen una vía en la solución de problemas de arquitectura de *software*. Los mismos poseen un nivel de abstracción

mucho mayor que los patrones de diseño. Además, brindan una descripción de los elementos y el tipo de relación que tienen, así como las restricciones para su uso (Buschmann, y otros, 1996).

Teniendo en cuenta que se empleará como herramienta base para el desarrollo la Plataforma GeneSIG y que esta ha sido desarrollada haciendo uso del *framework* CartoWeb es conveniente mantener la misma arquitectura para evitar futuras incompatibilidades.

El *framework* CartoWeb posee una arquitectura orientada a objetos y basada en componentes. Esto significa que está compuesto por *plugins*, los cuales están agrupados en dos carpetas, una denominada “*coreplugins*” y la otra “*plugins*”. La primera contiene aquellos *plugins* que son de obligatoria presencia para que pueda funcionar correctamente la Plataforma y la segunda está destinada a los *plugins* que serán creados por los desarrolladores de acuerdo a las funcionalidades que se deseen realizar. Esta arquitectura permite que GeneSIG sea flexible y fácil de personalizar para crear el SIG a desarrollar.

Para lograr un mejor entendimiento de los patrones arquitectónicos antes mencionados, a continuación se hace una breve descripción de cada uno de estos.

- **Patrón de arquitectura orientada a objetos**

El patrón de arquitectura orientada a objetos define el sistema como un conjunto de objetos que cooperan entre sí en lugar de un conjunto de procedimientos. Los componentes del estilo se basan en principios orientados a objetos: encapsulamiento, herencia y polimorfismo. Son las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación. Las interfaces están separadas de las implementaciones y en cuanto a las restricciones, puede admitirse o no que una interfaz pueda ser implementada por múltiples clases.

- **Patrón de arquitectura basado en componentes**

Las características principales de este patrón son la modularidad, la reusabilidad y la compatibilidad. El mismo describe un acercamiento al diseño de sistemas como un conjunto de componentes que exponen interfaces bien definidas y que colaboran entre sí para resolver problemas. Los componentes son diseñados de forma que puedan ser reutilizados en distintos escenarios en disímiles aplicaciones, aunque algunos componentes son realizados para una tarea específica.

Luego de haber analizado la arquitectura de la plataforma GeneSIG y haber decidido mantener la misma para el sistema a desarrollar, se hace necesario entonces definir los patrones de diseño que se pueden emplear en la construcción del SIG.

4.3 Modelo de Diseño

El propósito del diseño es especificar una solución que trabaje y pueda ser convertida en código fuente y construir una arquitectura simple y fácilmente extensible. Específicamente 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 con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Patrones de Diseño

Un patrón de diseño constituye una solución estándar para un problema común de programación en el desarrollo del *software*. Además es una técnica muy eficaz para flexibilizar el código haciéndolo satisfacer ciertos criterios, así como permite una manera más práctica de describir ciertos aspectos de la organización de un programa (Gamma, 1995).

- **Patrones de Diseño GRASP**

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) son patrones generales de *software* para la asignación de responsabilidades. Indica el nivel de responsabilidad de la creación de un objeto o la implementación de un método de manera que se conceda toda la información necesaria para crear el sistema (Gutierrez, 2006).

Entre los más significativos se encuentran los patrones siguientes: Experto, Creador, Alta Cohesión y Bajo Acoplamiento, los cuales se detallan a continuación:

Experto: debe considerarse como el principio fundamental a tener en cuenta siempre y cuando se esté asignando una responsabilidad a una clase. La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Una clase contiene toda la información necesaria para realizar la labor que tiene encomendada.

Creador: ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, o maneja varias instancias de la clase.

Alta Cohesión: Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. La información que almacena una clase debe ser coherente y estar en la mayor medida relacionada con la clase. Estas clases mejoran la claridad y la facilidad con que se entiende el diseño.

Bajo Acoplamiento: Debe haber pocas dependencias entre las clases. De manera que en caso de producirse una modificación en alguna de estas, se tenga la mínima repercusión posible en el resto de las clases. Con un bajo acoplamiento las clases de acceso a datos son casi independientes de las de abstracción a datos.

En el diseño del sistema se utilizan los patrones de asignación de responsabilidades, los cuales se ven reflejados en el modelo de diseño. A continuación se hace un resumen del uso de los patrones que fueron utilizados.

En el modelo de diseño se muestra que no existen dependencias fuertes entre las clases, permitiendo así que al producirse una modificación en alguna de estas, se tenga la mínima repercusión posible en el resto de las clases y pueda ser reutilizado el código, evidenciándose el uso del patrón **bajo acoplamiento**.

Además existen las clases clientes y las clases servidoras las cuales se encargan de realizar cada

una su función, esto permite que las clases no estén saturadas y que se garantice a través de la asignación de responsabilidades, que el sistema presente **alta cohesión** en sus clases.

Cuando llega una petición del cliente este construye un objeto *request* y cuando el sistema gestiona la petición construye un objeto *result*, cada clase es la encargada de crear nuevos objetos y es creada por la clase que tiene mayor conocimiento, esto fue posible haciendo uso del patrón **creador** y del **experto**.

- **Patrones de Diseño GOF**

Además del uso de los patrones GRAPS se tuvo en cuenta los patrones GOF acrónimo de “*Gang of Four*”, los cuales se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

De los patrones GOF existentes, se utilizan en el diseño los siguientes: *Singleton* y *Command*, de los cuales se realiza una breve descripción a continuación:

Singleton (instancia única): en el diseño de clases es necesario aplicar la solución del patrón *Singleton* que no es más que garantizar el acceso único a una clase mediante una única instancia.

De esta forma se controla el acceso a las clases. Se utiliza para modificar el *framework* CartoWeb, el objetivo del mismo es crear el objeto “mapa” para que no se cree cada vez que se hace un envío en la aplicación.

Command (acción): encapsula una petición en un objeto, pudiendo realizar operaciones como crear una cola de objetos, gestionarla y poder deshacer las operaciones. Entre las ventajas que posee se encuentran la simplificación de llamadas, estructuración del sistema y que permite realizar una cola de operaciones.

Se utiliza en el proceso de petición mediante la Interfaz Gráfica de Usuario (GUI) al sistema de una información cualquiera por un cliente. Esto se evidencia en la clase *AjaxHelper* que es la encargada de manejar las peticiones mediante la interfaz. Uno de los aspectos más importantes en el sistema son las GUI, ya que el usuario interactúa constantemente con ellas y por eso se aplica la solución

propuesta por este patrón de diseño.

Diagrama de Clases del Diseño

Una clase del diseño es una abstracción de una clase o construcción similar en la implementación del sistema. El lenguaje que se utiliza en dichas clases es el mismo que se emplea para la implementación del sistema; se especifican los atributos y las operaciones; se pueden realizar interfaces si tienen sentido para la programación y los métodos tienen correspondencia con los métodos de la programación.

A continuación se muestra en la Figura 4.1 el Diagrama de clases del diseño del CU Mostrar información.

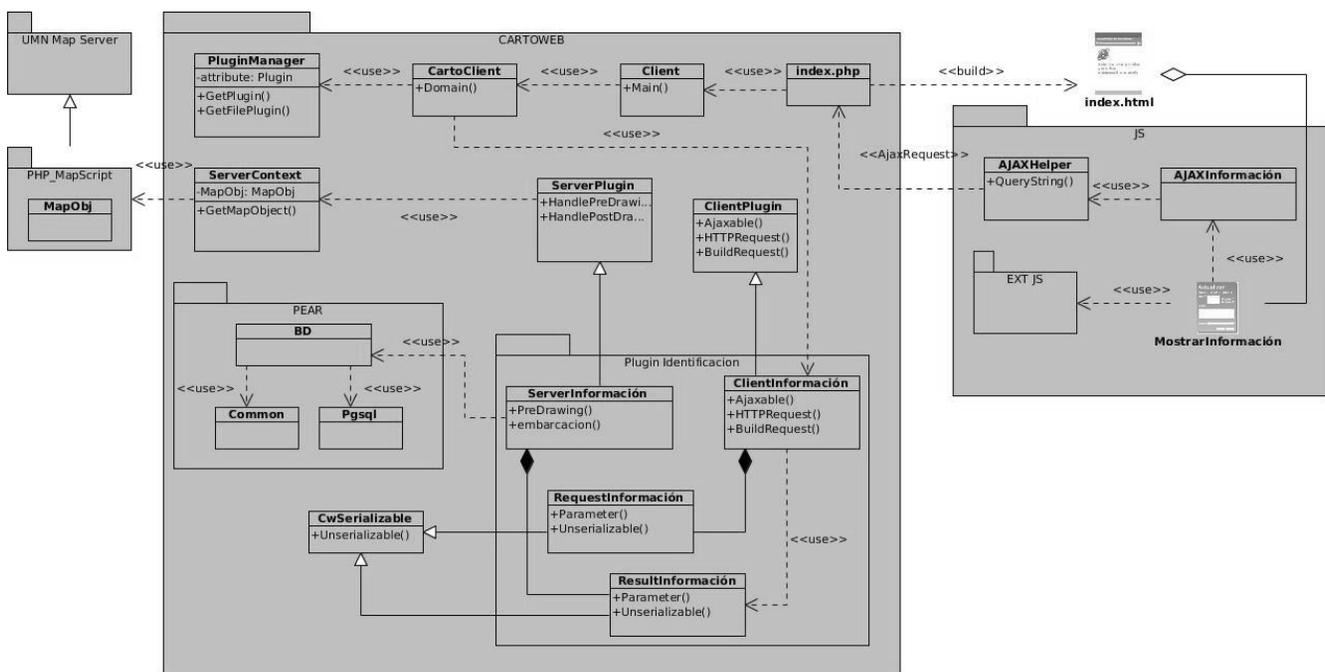


Figura 4.1: Diagrama de clases del diseño del CU Mostrar Información.

En este diagrama se evidencian las relaciones entre las clases que intervienen en el CU Modificar Información.

Descripción de las clases

Cada uno de los diagramas de Clases de Diseño que emplean el *framework* CartoWeb poseen clases que son comunes para la mayoría de ellos, tanto las que son JavaScript como las que son PHP, estas clases serán descritas a continuación.

Clases del diseño asociadas al *framework* CartoWeb

- ***index.php***: tiene como propósito controlar la realización del CU en sí, recibe las peticiones realizadas por el cliente, gestiona las mismas y manda a construir la *ClientPage*.
- ***Client***: contiene todos los archivos específicos de PHP del lado de *CartoClient* y permite la interacción entre la *index.php* y la *CartoClient*.
- ***CartoClient***: integra y recoge todos los datos y funciones realizadas por cada una de las *.js* que intervienen en el caso de uso, y se definen una serie de variables globales que van a ser utilizadas por la aplicación.
- ***PluginManager***: clase que se utiliza para gestionar la base de *plugins*.
- ***ClientPlugin***: contiene las interfaces necesarias para los *plugins* del lado del cliente.
- ***ServerPlugin***: esta clase proporciona la base de herramientas para el desarrollo de *plugins*.
- ***ServerContex***: es la contenedora de la información común que ha de ser utilizada por la parte cliente y la servidora, empleando la información seleccionada como un objeto para un fácil manejo de los datos.
- ***MapObj***: es donde se definen los métodos, funciones, además del lenguaje para el intercambio de datos con el servidor de mapa (MapServer).
- ***BD***: es la clase encargada de establecer la conexión con el servidor de base de datos para procesar los objetos a editar.
- ***Common***: es la encargada de administrar las conexiones a la base de datos para ejecutar las consultas a la misma satisfactoriamente, esto incluye tratamiento de los datos.
- ***Pgsqli***: gestiona desde PHP las funciones de postgresQL.
- ***CwSerializable***: se encarga de serializar todas aquellas clases que pueden ser serializadas, permitiendo la comunicación entre el *Client* y el *Server* del *plugin*.
- ***AJAXHelper***: tiene como propósito enviar las respuestas de los *plugins* "AJAX", para

alimentar a los *plugins* que responden a las peticiones del usuario.

- **Index.html:** es la encargada de mostrar en el mapa la región localizada.
- **AJAXinformación:** es la encargada de gestionar el pedido y respuesta a las peticiones del usuario por ajax.
- **MostrarInformacion:** tiene como objetivo mostrar la información de los centrales y empresas de AZCUBA.
- **Serverinformacion:** es la clase servidora la cual tiene como principal función la conexión con la base de datos para efectuar los cambios requeridos y enviar las respuestas necesarias al Clientinformacion.
- **Clientinformacione:** recoge y selecciona de las .js contenidas en el paquete JS, toda la información correspondiente a los datos a precisar, entrados a través de los formularios, y los envía al Serverinformación.
- **Requestinformaciónr:** es una clase común encargada de transportar los datos recogidos en Clientinformación desde la interfaz y transportarlos a la clase Serverinformación (con propósito general).
- **Resultinformación:** es una clase común encargada de transportar los datos generados en Serverinformación a la clase Clientinformacion.

Los diagramas de clases del diseño correspondientes a los restantes casos de uso se podrán consultar en el Anexo 4. Además de los diagramas de clases del diseño es importante también conocer antes de diseñar las Interfaces de Usuario, los principios principales a tener en cuenta para lograr un diseño con calidad y que satisfaga las expectativas de los clientes.

4.4 Principios de Diseño de Interfaz de Usuario

Para lograr una interfaz de usuario amigable, atractivo y funcional para el usuario final, es necesario definir principios de diseño de interfaz de usuario, pues esta es una actividad de gran importancia dentro del diseño de un *software*. Siempre teniendo en cuenta que el éxito de la empresa depende en gran medida de la aceptación por parte de los usuarios.

Principios de Diseño de interfaz

A continuación se definen los principios a tener en cuenta en el desarrollo de la interfaz de usuario de la aplicación:

- Las funcionalidades deberán estar al alcance de un *clic* y representadas por íconos asociados a la acción que se realiza, de manera que cualquier persona con un mínimo dominio de la informática pueda hacer uso del sistema.
- Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- Los mensajes mostrados al usuario deben ser concisos y de fácil comprensión.
- Los menús y etiquetas de botones deben comenzar con la palabra más importante.
- Los eventos más importantes del sistema deben ser mostrados en una barra de estado.

Teniendo en cuenta estos principios para el diseño de las interfaces de usuario se garantiza elevar la calidad de las mismas y lograr una mayor satisfacción por parte de los usuarios del sistema. Una vez que se haya concluido el diseño de las interfaces, se procede con el diseño de la Base de Datos.

4.5 Diseño de la Base de Datos

Es en el modelo de datos donde se va a describir la representación física de las tablas que conforman dicha base de datos, con sus llaves y descripciones de campos. A continuación se muestra en la figura 4.2 el modelo relacional.

Capítulo #4 Construcción de la Solución

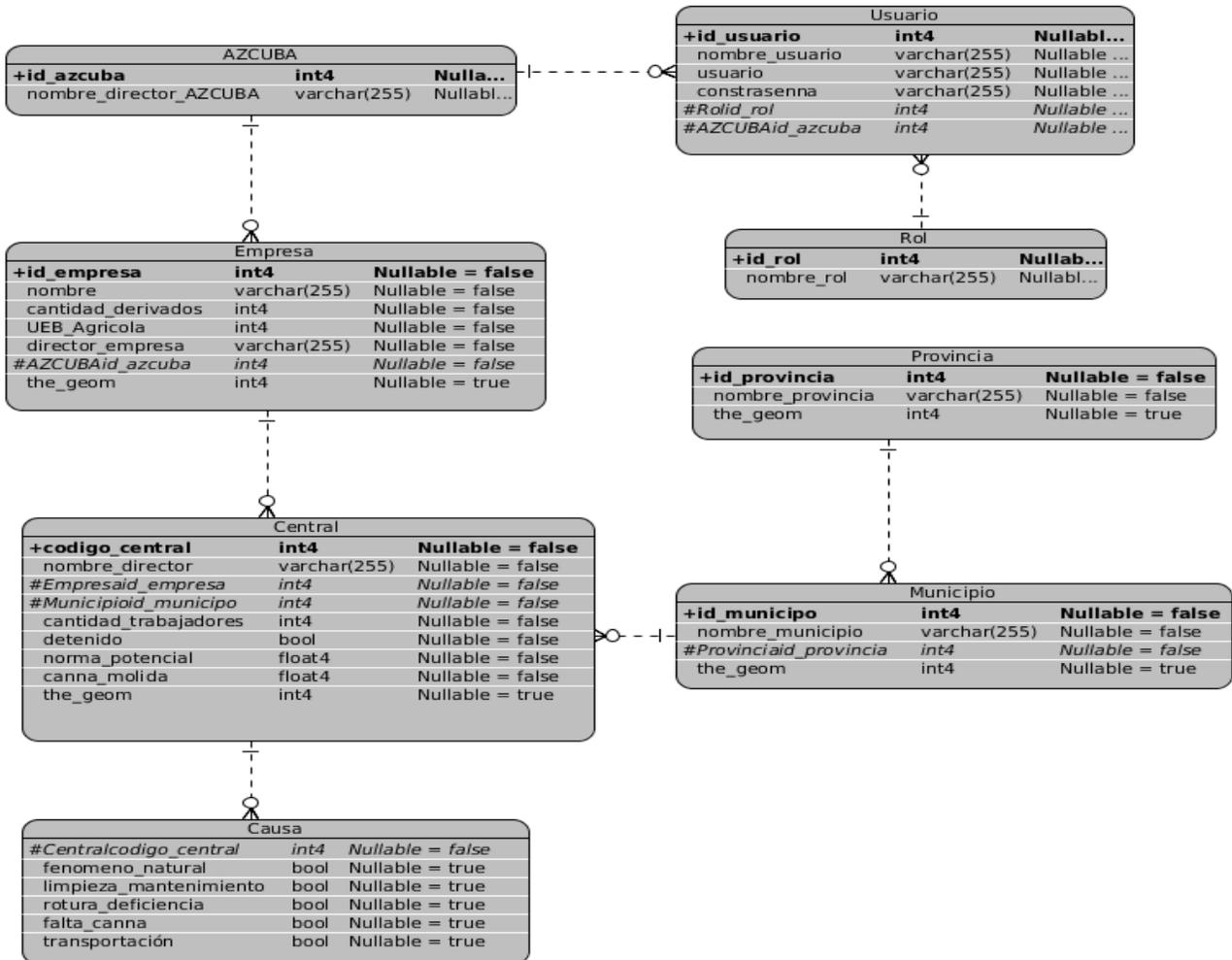


Figura 4.2: Modelo Relacional.

Como se puede observar en el diagrama, se disponen de 8 tablas: Una vez que se cuenta con el diseño de la Base de Datos, se procede entonces a realizar el Modelo de Implementación, el cual permitirá tener una representación exacta de los componentes que intervienen así como la relación que se establece entre estos.

4.6 Modelo de Implementación

El modelo de implementación está compuesto por una colección de componentes, subsistemas de implementación, y paquetes utilizados para agrupar elementos del modelo. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Los diagramas de componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes.

Diagrama de componentes

Los diagramas de componentes fueron confeccionados, agrupando los casos de uso por funcionalidad, de la misma manera que fueron agrupados para confeccionar los diagramas de clases del diseño. A continuación se muestra el Diagrama de Componentes del CU Crear Mapa Temático.

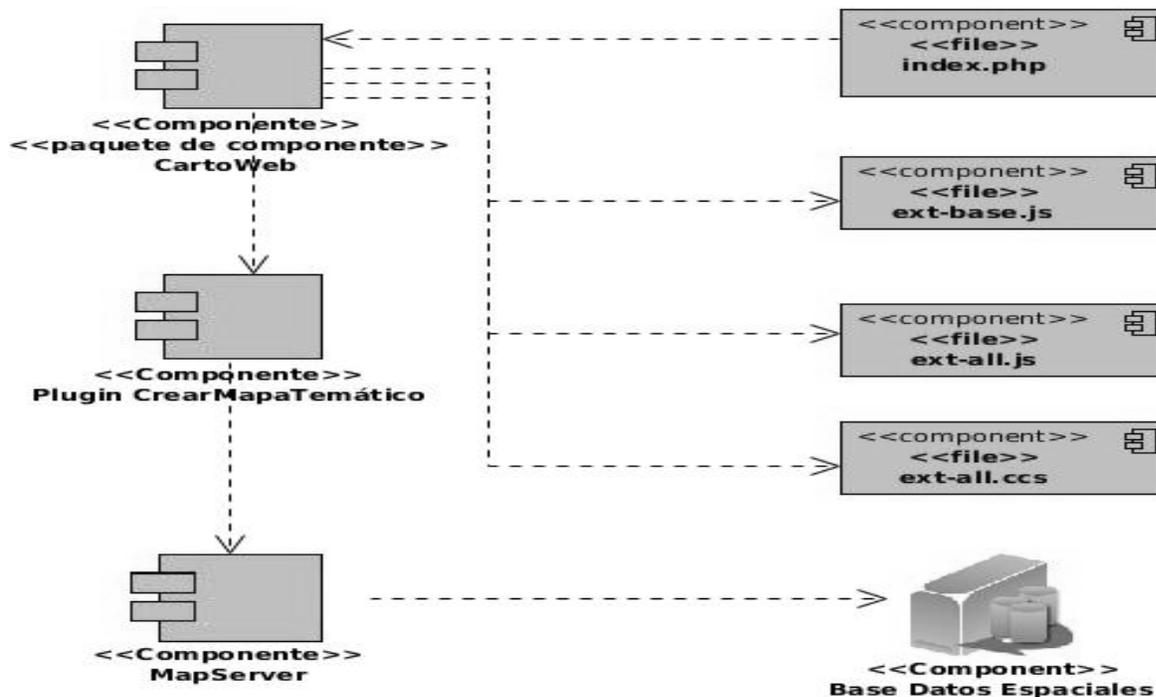


Figura 4.3: Diagrama de componentes.

Los restantes Diagramas de Componentes se encuentran en los anexos. Luego de conocer cómo quedó estructurado el Modelo de Implementación, se hace necesario conocer también la estructura del Modelo de Despliegue.

4.7 Modelo de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema. Es un conjunto de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos y procesos.

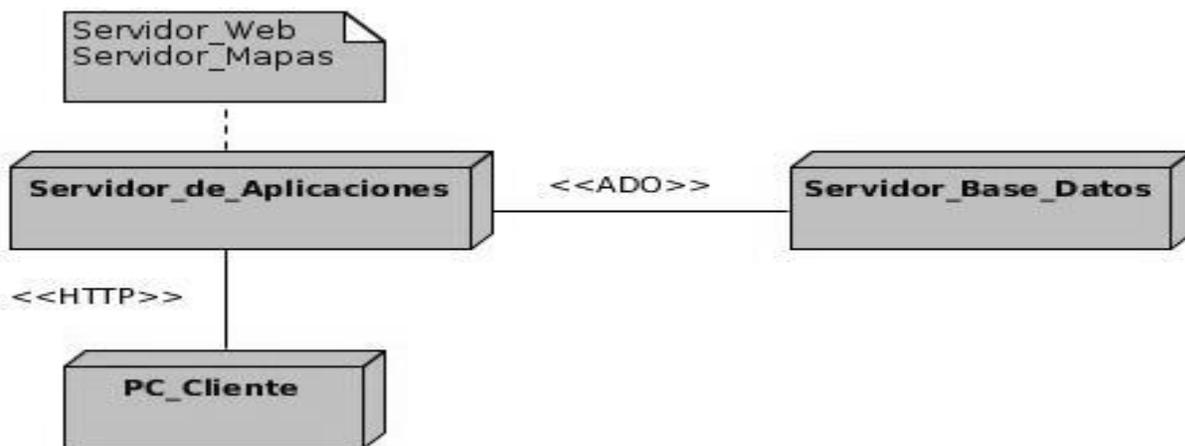


Figura 4.4: Modelo de Despliegue.

4.8 Diseño de Casos de Prueba

Las pruebas de *software* son de vital importancia ya que proporciona un alto grado de confianza y seguridad en el producto, a la vez que permite hacer estimaciones realistas sobre el comportamiento del producto una vez que se encuentre en funcionamiento. Realizar las pruebas de *software* a un producto es un requisito que debe ser cumplido con rigor.

Las pruebas son una actividad en la cual el sistema o los componentes son ejecutados bajo ciertas condiciones o requerimientos especificados, donde los resultados son observados y registrados para realizar una evaluación de algún aspecto del sistema o del componente.

Objetivos de las pruebas

- Las pruebas buscan encontrar los posibles fallos en la implementación, en la usabilidad y en la calidad de un programa determinada para validar su correcto funcionamiento. Entre los objetivos de la realización de las pruebas se tiene:
- Detectar defectos en la aplicación.
- Verificar que todos los requisitos se han implementado satisfactoriamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el producto al usuario o cliente final.
- Diseñar los casos de prueba que permitan identificar diferentes clases de errores utilizando la menor cantidad de tiempo y esfuerzo.

Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas Pruebas de Comportamiento, se centran en los requisitos funcionales del *software*. La prueba de caja negra intenta encontrar errores de las siguientes categorías:

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

Estas pruebas se refieren a las pruebas que se llevan a cabo sobre la interfaz del *software*. Los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se

acepta de forma adecuada y que se produce un resultado correcto. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del *software* (Pressman, 2002).

Para el desarrollo de las pruebas de caja negra existen varias técnicas, entre las que se encuentran:

Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del sistema.

Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del sistema para manejar datos que se encuentran en los límites aceptables.

Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

En la presente investigación se utiliza la técnica de Partición de Equivalencia ya que esta te permite reducir los casos de prueba de forma tal que se realicen los casos de prueba necesarios para probar las clases de datos válidos, inválidos, y ver si el sistema funciona correctamente.

Pruebas para la Solución Propuesta

En la presente investigación se utiliza para el diseño de casos de prueba las pruebas de caja negra. A continuación se muestra el diseño de caso de prueba (DCP) del caso de uso Gestionar Usuario, los restantes DCP correspondientes a cada caso de uso se encuentran los anexos.

Descripción General

El caso de uso se inicia cuando el administrador desea realizar la gestión de los usuarios. Esta funcionalidad permite agregar nuevos usuarios, así como modificar o eliminar los ya existentes.

Condiciones de Ejecución

El administrador debe estar autenticado.

Capítulo #4 Construcción de la Solución

Tabla 4.1: Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Gestionar usuario	EC 1.1: Gestionar usuario satisfactoriamente.	El sistema realiza la acción según la operación de gestión (Adicionar, Modificar o Eliminar), seleccionada por el usuario.	<ul style="list-style-type: none"> • Seleccionar en la barra de herramientas la opción “Gestionar usuarios”. • Seleccionar la sección correspondiente a la acción que desea realizar.
	EC 1.2: Cerrar panel “Usuarios”.	El sistema oculta el panel usuario.	<ul style="list-style-type: none"> • Seleccionar en la barra de herramientas la opción “Gestionar usuarios”. • Cerrar el panel “Usuarios” mediante la opción correspondiente (X).
SC2: Adicionar usuario	EC 2.1: Adicionar usuario satisfactoriamente.	El sistema adiciona el usuario, mostrando un mensaje que confirma dicha acción.	<ul style="list-style-type: none"> • Seleccionar en la barra de herramientas la opción “Gestionar usuarios”. • Seleccionar la sección “Adicionar”.

Capítulo #4 Construcción de la Solución

			<ul style="list-style-type: none">• Llenar los datos correspondientes.• Seleccionar la opción "Aceptar".
EC 2.2: Se dejan campos en blanco.	El sistema muestra el mensaje "Hay campos con valores inválidos".		<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción "Gestionar usuarios".• Seleccionar la sección "Adicionar".• Llenar los datos correspondientes.• Seleccionar la opción "Aceptar".
EC 2.3: No coinciden la contraseña y la confirmación de la misma.	El sistema muestra el mensaje: "Los campos contraseña y su confirmación deben ser iguales".		<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción "Gestionar usuarios".• Seleccionar la sección "Adicionar".• Llenar los datos correspondientes.• Seleccionar la opción "Aceptar".

Capítulo #4 Construcción de la Solución

SC3: Modificar usuario	EC 3.1: Modificar usuario satisfactoriamente.	El sistema modifica el usuario, mostrando un mensaje que confirma dicha acción.	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Modificar”.• Seleccionar el usuario que se desea modificar.• Modificar los datos correspondientes.• Seleccionar la opción “Aceptar”.
	EC 3.2: Se dejan campos en blanco.	El sistema muestra el mensaje “Hay campos con valores inválidos”.	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Modificar”.• Seleccionar el usuario que se desea modificar.• Modificar los datos correspondientes.• Seleccionar la opción “Aceptar”.

Capítulo #4 Construcción de la Solución

	EC 3.3: No coinciden la contraseña y la confirmación de la misma.	El sistema muestra el mensaje: “Los campos Nueva Contraseña y Repetir Contraseña deben ser iguales”	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Modificar”.• Seleccionar el usuario que se desea modificar.• Modificar los datos correspondientes.• Seleccionar la opción “Aceptar”.
SC4: Eliminar usuario	EC 4.1: Eliminar usuario satisfactoriamente.	El sistema elimina el usuario, mostrando un mensaje que confirma dicha acción.	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Eliminar”.• Seleccionar el usuario que se desea eliminar.• Seleccionar la opción “Aceptar”.• Confirmar la eliminación del usuario (seleccionar “sí” en el mensaje de confirmación).

Capítulo #4 Construcción de la Solución

	EC 4.2:No se confirma la eliminación del usuario.	El sistema oculta el mensaje de confirmación y mantiene el usuario registrado en el sistema.	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Eliminar”.• Seleccionar el usuario que se desea eliminar.• Seleccionar la opción “Aceptar”.• Confirmar la eliminación del usuario (seleccionar “no” en el mensaje de confirmación).
	EC 4.3:Cerrar mensaje de confirmación.	El sistema oculta el mensaje de confirmación y mantiene el usuario registrado en el sistema.	<ul style="list-style-type: none">• Seleccionar en la barra de herramientas la opción “Gestionar usuarios”.• Seleccionar la sección “Eliminar”.• Seleccionar el usuario que se desea eliminar.• Seleccionar la opción “Aceptar”.• Cerrar el mensaje de confirmación mediante

Capítulo #4 Construcción de la Solución

			la opción correspondiente (X).
--	--	--	--------------------------------

Capítulo #4 Construcción de la Solución

Tabla 4.2: Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Usuario	<ul style="list-style-type: none">• Campo de texto (En la opción “Adicionar”).• Lista desplegable (En las opciones “Modificar y Eliminar”).	No	Este campo no debe estar en blanco.
2	Nombre	Campo de texto	No	Este campo no debe estar en blanco.
3	Contraseña	Campo de texto	No	Este campo no debe estar en blanco.
4	Repetir contraseña	Campo de texto	No	Este campo no debe estar en blanco.
5	Rol	Lista desplegable	No	Se debe seleccionar un rol.
6	Provincia a la que tiene acceso	Lista desplegable	No	Se debe seleccionar una o todas las provincias.

Capítulo #4 Construcción de la Solución

Tabla 4.3: Matriz de datos

ID del escenario	Escenario	V1	V2	V3	V4	V5	V6	Respuesta del Sistema	Resultado de la Prueba
EC 2.1	Adicionar usuario satisfactoria mente.	V (lledea)	V (lily)	V (pass)	V (pass)	V (administrador)	V (Todas)	El sistema adiciona el usuario, mostrando un mensaje que confirma dicha acción.	Satisfactorio
EC 2.2	Se dejan campos en blanco.	I ()	I ()	I ()	I ()	I ()	I ()	El sistema muestra el mensaje "Hay campos con valores inválidos".	Satisfactorio
		V ()	V (lily)	V (pass)	V (pass)	V (administrador)	V (Todas)		
		V (lledea)	I ()	V (pass)	V (pass)	V (administrador)	V (Todas)		

Capítulo #4 Construcción de la Solución

ID del escenario	Escenario	V1	V2	V3	V4	V5	V6	Respuesta del Sistema	Resultado de la Prueba
		V (lledea)	V (lily)	I ()	V (pass)	V (administrador)	V (Todas)		
		V (lledea)	V (lily)	V (pass)	I ()	V (administrador)	V (Todas)		
		V (lledea)	V (lily)	V (pass)	I ()	V (administrador)	I ()		
EC 2.3:	No coinciden la contraseña y la confirmación de la misma.	V (lledea)	V (lily)	V (pass)	I (pas)	V (administrador)	V (Todas)	El sistema muestra el mensaje: “Los campos contraseña y su confirmación deben ser iguales”.	Satisfactorio

En la primera iteración de pruebas realizadas al sistema se encontraron no conformidades que fueron posteriormente corregidas, luego se procedió a realizar una segunda iteración la cual arrojó mejores resultados pero todavía existían no conformidades, en una tercera iteración realizada, la aplicación arrojó resultados satisfactorios por lo que no fue necesario una cuarta iteración. En el anexo 9 se encuentra la figura 3 que muestra una gráfica con una comparación de los resultados arribados en la 1era, 2da y 3era iteración de pruebas.

4.9 Conclusiones Parciales

En el presente capítulo se llevó a cabo la modelación de los diferentes artefactos correspondientes a los flujos de trabajo diseño, implementación y prueba. Por lo que se puede concluir que:

- Se definieron los patrones arquitectónicos y de diseño a aplicar estructurando correctamente el *software* y refinando las relaciones entre sus componentes los mismos permitieron establecer las pautas para una correcta implementación.
- La elaboración de los diagramas de clase del diseño que, unido al diagrama de componentes y despliegue guiaron la implementación del sistema.
- Por su parte, las pruebas de caja negra con su técnica partición equivalente posibilitaron comprobar que el sistema cumple con los requisitos definidos y que posee un nivel adecuado de calidad.

Conclusiones

Una vez culminada la investigación es posible afirmar que se dio cumplimiento a cada uno de los objetivos trazados, por lo que se concluye que:

- La elección de RUP como metodología de desarrollo permitió documentar el ciclo de vida de la aplicación a través de los artefactos generados que brindan la posibilidad de realizar nuevas versiones o añadirle nuevas funcionalidades.
- La utilización de las tecnologías y herramientas seleccionadas para el desarrollo del sistema ofrecieron el soporte necesario para lograr un producto que cumpla con los requisitos propuestos y que sea totalmente libre.
- La realización de las pruebas mediante la técnica partición equivalente permitió detectar los posibles errores, para así dar cumplimiento de los requisitos funcionales y no funcionales establecidos y obtener un producto con la calidad adecuada.
- Con el desarrollo del Sistema de Información Geográfica se encamina el proceso de personalización de los productos desarrollados por el proyecto al mejoramiento de la toma de decisiones.

Recomendaciones

Concluida la investigación y basándose en las experiencias acumuladas a lo largo del desarrollo de la misma, se proponen las siguientes recomendaciones:

- Proponer y aplicar el sistema desarrollado en el Grupo Estatal AZCUBA.
- Que se añada al módulo de localización una funcionalidad que sea localizar campos de caña asociados a los centrales azucareros y realizar una representación de los mismos en el mapa.
- Que se implemente y se añada al sistema una ayuda para la correcta y fácil utilización del mismo por parte de los usuarios.
- Que se realice la validación del sistema mediante el uso de otros tipos de pruebas.

Referencia Bibliográfica

Águila, Adrián Gracia. 2011.*Plataforma Tecnología Soberana, para el almacenamiento, representación y análisis de información espacial en interés para la defensa.* Moa : s.n., 2011.

Aplicación de un sistema geoespacial de soporte a la toma de decisiones para un desarrollo sostenible y ecológico en la agricultura cañera cubana. **Perez, Arlenis. 2005.** 3, 2005, Vol. 8.

Buschmann, Frank, Meunier, Regine y Rohnert, Hans. 1996.*Pattern-Oriented Software Architecture: A System of.* 1996.

Cartografia.cl. 2005. Cartografia.cl. [En línea] 2005. [Citado el: 4 de diciembre de 2012.] http://www.cartografia.cl/beta/index.php?option=com_content&view=article&id=668:ventajas-de-la-cosecha-con-ayuda-de-sig&catid=63:noticias&Itemid=18.

Cobo, Ángel. 2005.*PHP y MySQL: Tecnología para el desarrollo de aplicaciones web.* 2005.

Cockburn, Alistair. 2006.*Agile Software Development.* 2006.

Cuba, Gaceta Oficial de la República de. 2011.*Gaceta Oficial de la República de Cuba.* La Habana : s.n., 2011. 37.

Departamento de Sistemas Informáticos y Computación. 2010.*Rational Unified Process (RUP).* Valencia : s.n., 2010.

Esparza, René, Reséndiz, Rita y Embriz, Arnulfo. 2000.*Catálogo de mapas, planos, croquis e ilustraciones históricas de Restitución y Dotación de tierras y Ampliación de ejidos del archivo general agrario.* Mexico : s.n., 2000.

Espinosa, Antonio Membrides. 2000.*Fundamentos del Mapserver, Mapscript, PostGIS y su integración con el Cartoweb.* Ciudad de la Habana : s.n., 2000.

Fornaris, Leiber. 2011.*Sistema de Información Geográfica para la representación de objetivos hidrográficos de precipitaciones.* 2011.

Gamma, Erich. 1995.*Design Patterns: Elements of Reusable Object-Oriented Software.* 1995.

Gutierrez, Jorge. 2006.*Patrones Grasp.* 2006.

Guzmán, Adriana Patricia. 2006.*Comunicación Empresarial.* 2006.

imta. 2005. imta. [En línea] 2005. [Citado el: 6 de diciembre de 2012.] http://www.imta.gob.mx/cotennser/index.php?option=com_content&view=article&id=124:sistema-experto-para-aumentar-productividad-de-cana-de-azucar-&catid=41:noticias-y-articulos&Itemid=74.

Llopis, Juan Peña. 2006. *Sistemas de información geográfica aplicados a la gestión del territorio*. s.l. : Club Universitario, 2006.

Mapserver, Sitio Oficial. 2011. MapServer open source web mapping. *MapServer open source web mapping*. [En línea] 2011. [Citado el: 8 de enero de 2013.] <http://mapserver.org/trunk/es/about.html#about>.

Maza, Miguel Ángel Sánchez. 2001. *JavaScript*. 2001.

Navarro, Antoni Pérez. 2011. *Introducción a los sistemas de información geográfica y geotelemática*. s.l. : UOC, 2011.

Olaya, Víctor. 2010. *Sistemas de Información Geográfica*. Girona : s.n., 2010.

Paradigm, Visual. 2011. Visual Paradigm. [En línea] 2011. <http://www.visual-paradigm.com...>

Perez, Arlenis. 2009. Aplicación de un sistema geoespacial de soporte a la toma de decisiones. [En línea] 2009.

PgadminIII. 2008. PgadminIII. [En línea] 10 de 3 de 2008. [Citado el: 21 de 2 de 2013.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.

Pressman. 2002. *Ingeniería del Software, un enfoque práctico*. 2002.

Pressman, Roger. 2006. *Ingeniería de Software. Un enfoque práctico. 5ta Edición*. México : s.n., 2006.

—. 2006. *Ingeniería de Software: Un enfoque práctico*. 2006.

Quintero, Ing. Duvier Olaya. 2010. slideshare. [En línea] 14 de junio de 2010. [Citado el: 4 de diciembre de 2012.] <http://www.slideshare.net/AlvaroPuentesMolina/03-los-datos-geograficos>.

serviciosTIC. 2006. serviciosTIC. [En línea] 2006. [Citado el: 4 de diciembre de 2012.] <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.

Sloccum, Jack. 2008. *Introducción a ExtJs*. 2008.

Urrutia, Javier. 2006. *Cartografía Orientación y GPS*. s.l. : ETOR-OSTOA S.L, 2006.

—. 2006. *Cartografía, orientación y GPS*. 2006.

Bibliografía

- Águila, Adrián Gracia. 2011.***Plataforma Tecnología Soberana, para el almacenamiento, representación y análisis de información espacial en interés para la defensa.* Moa : s.n., 2011.
- Alonso Omo, Fernando y otros. 2005.***Introducción a la ingeniería del software.* 2005.
Aplicación de un sistema geoespacial de soporte a la toma de decisiones para un desarrollo sostenible y ecológico en la agricultura cañera cubana. **Perez, Arlenis. 2005.** 3, 2005, Vol. 8.
- Ayala, Claudia P. 2005.***Análisis Comparativo de Lenguajes de Modelado Orientados a Objetivos basados en i*.* Barcelona : s.n., 2005.
- Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 1998.***The Unified Modeling Language Reference Manual 1998.* 1998.
- Bosque, Sendra J. 1992.***Sistemas de Información Geográfica.* Madrid : s.n., 1992.
- Bravo, Javier Domínguez. 2000.***Breve Introducción a la Cartografía y a los Sistemas de Información Geográfica (SIG).* Madrid : CIEMAT, 2000.
- Buschmann, Frank, Meunier, Regine y Rohnert, Hans. 1996.***Pattern-Oriented Software Architecture: A System of.* 1996.
- Camacho, Erika y Nuñez, Fabio Cardeso and Gabriel. 2004.***Arquitecturas de Software, Guías de Estudio.* 2004.
- Capderrich Falgueras, Benet. 2003.***Ingeniería de software.* 2003.
- Cartografia.cl. 2005.** Cartografia.cl. [En línea] 2005. [Citado el: 4 de diciembre de 2012.] http://www.cartografia.cl/beta/index.php?option=com_content&view=article&id=668:ventajas-de-la-cosecha-con-ayuda-de-sig&catid=63:noticias&Itemid=18.
- Cerda, Felipe. 2009.***NetBeans 6.5, el único IDE que necesitas.* 2009.
- Chorley, Richard J. 1987.** Nuevas tendencias en geografía . [En línea] 1987.
- Cobo, Ángel. 2005.***PHP y MySQL: Tecnología para el desarrollo de aplicaciones web.* 2005.
- Cockburn, Alistair. 2006.***Agile Software Development.* 2006.
- Comas, David y Ruiz, Ernest. 1993.***Fundamentos de los sistemas de información geográfica .* 1993.
- Cuba, Gaceta Oficial de la República de. 2011.***Gaceta Oficial de la República de Cuba.* La Habana : s.n., 2011. 37.
- Departamento de Ingeniería de Software. 2007-2008.***Introducción al proceso de desarrollo de software.* Habana : s.n., 2007-2008.
- Departamento de Sistemas Informáticos y Computación. 2010.***Rational Unified Process (RUP).* Valencia : s.n., 2010.
- Díaz Narváez, Víctor. 2009.***Metodología de la investigación científica y bioestadística: para médicos.* 2009.
- Díaz, Alvaro A. Penroz. 2005.***Graphical User Interface (GUI) para el programa servidor de mapas MapServer 4.6.1.* 2005.
- Entorno Virtual de Aprendizaje. 2007-2008.** Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] 2007-2008. <http://eva.uci.cu/course/view.php?id=161>.
- Entorno Visual de Aprendizaje . 2010.** Eva. [En línea] 7 de 12 de 2010. [Citado el: 13 de 4 de 1013.] Eva.uci.cu.
- Esparza, Rene, Resendiz, Rita y Embriz, Arnulfo. 2000.***Catálogo de Mapas, Planos, Croquis e Ilustraciones Históricas de Restitución ...* 2000.

- Esparza, René, Reséndiz, Rita y Embriz, Arnulfo. 2000.***Catálogo de mapas, planos, croquis e ilustraciones históricas de Restitución y Dotación de tierras y Ampliación de ejidos del archivo general agrario.* Mexico : s.n., 2000.
- Espinosa, Antonio Membrides. 2000.***Fundamentos del Mapserver, Mapscript, PostGIS y su integración con el Cartoweb.* Ciudad de la Habana : s.n., 2000.
- . **2008.***Plataforma GeneSIG.* Habana : s.n., 2008.
- ESRI. 1995.** ArcGis Resources. [En línea] 1995. [Citado el: 1 de 2 de 2013.] <http://resources.arcgis.com>.
- Fornaris, Leiber. 2011.***Sistema de Información Geográfica para la representación de objetivos hidrográficos de precipitaciones.* 2011.
- Gallego, Juan Pablo Gómez. 2007.***Fundamentos de la Metodología RUP.* s.l. : UNIVERSIDAD TECNOLÓGICA DE PEREIRA, 2007.
- Gamma, Erich. 1995.***Design Patterns: Elements of Reusable Object-Oriented Software.* 1995.
- Gutierrez, Jorge. 2006.***Patrones Grasp.* 2006.
- Guzmán, Adriana Patricia. 2006.***Comunicación Empresarial.* 2006.
- imta. 2005.** imta. [En línea] 2005. [Citado el: 6 de diciembre de 2012.] http://www.imta.gob.mx/cotennser/index.php?option=com_content&view=article&id=124:sistema-experto-para-aumentar-productividad-de-cana-de-azucar-&catid=41:noticias-y-articulos&Itemid=74.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.***El Proceso Unificado de Desarrollo de Software.* España : Pearson Educación, 2000.
- Jacobson, P.Jonsson, M. Christerson and G.Overgaard. 1992.***Ingeniería de Software orientada a Objetos- Un acercamiento a través de los casos de uso.* 1992.
- Kiccilof, Carlos Reynoso and Nicolás. 2004.***Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* 2004.
- Kruchten, Philippe. 2004.***The Rational Unified Process: An Introduction .* 2004.
- Laman, Craig. 1999.***UML y Patrones. Introducción a1 análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999.
- Larman, Craig. 2002.***UML y Patrones una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2002.
- . *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* Canadá : s.n.
- Llopis, Juan Peña. 2006.***Sistemas de información geográfica aplicados a la gestión del territorio.* s.l. : Club Universitario, 2006.
- Mapserver, Sitio Oficial. 2011.** MapServer open source web mapping. *MapServer open source web mapping.* [En línea] 2011. [Citado el: 8 de enero de 2013.] <http://mapserver.org/trunk/es/about.html#about>.
- Maza, Miguel Ángel Sánchez. 2001.***JavaScript.* 2001.
- Navarro, Antoni Pérez. 2011.***Introducción a los sistemas de información geográfica y geotelemática.* s.l. : UOC, 2011.
- NCGIA. 1990.** National Center for Geographic Information and Analysis. [En línea] 1990. [Citado el: 1 de 02 de 2013.] <http://www.ncgia.ucsb.edu/>.
- Olaya, Víctor. 2010.***Sistema de Información Geográfica.* 2010.
- . **2010.***Sistemas de Información Geográfica.* Girona : s.n., 2010.
- Ortiz Ocaña, Alexander. 2006.***Temas pedagógicos, didácticos y metodológicos.* 2006.
- Paradigm, Visual. 2011.** Visual Paradigm. [En línea] 2011. <http://www.visual-paradigm.com...>
- Pavón, Eduardo León. 2000.***Tutorial Visual Paradigm for UML.* España : s.n., 2000.

- Perez, Arlenis. 2009.** Aplicación de un sistema geoespacial de soporte a la toma de decisión. [En línea] 2009.
- PgadminIII. 2008.** PgadminIII. [En línea] 10 de 3 de 2008. [Citado el: 21 de 2 de 2013.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.
- PostgreSQL. 2013.** PostgreSQL Sitio oficial del Servidor de Base de datos. [En línea] 2013. [Citado el: 12 de noviembre de 2012.] http://www.dirphp.com/dir/Software_y_servidores_PHP/PostgreSQL_Sitio_oficial_del_Servidor_de_Base_de_datos_51.html.
- Pressman. 2002.** *Ingeniería del Software, un enfoque práctico*. 2002.
- Pressman, Roger. 2005.** *Ingeniería de Software*. 2005.
- . 2004. *Ingeniería de Software. Un enfoque práctico*. 2004. Quinta Edición.
- . 2006. *Ingeniería de Software: Un enfoque práctico*. 2006.
- Pressman, Roger S. 2002.** *Ingeniería de Software, un enfoque práctico*. 2002.
- Pressman, Roger. 2005.** *Software Engineering : A practitioner's Approach*. 2005.
- QGIS. 2007.** Quantum GIS. [En línea] 2007. [Citado el: 10 de 02 de 2013.] <http://www.qgis.org/en/documentation/manuals.html>.
- Quintero, Ing. Duvier Olaya. 2010.** slideshare. [En línea] 14 de junio de 2010. [Citado el: 4 de diciembre de 2012.] <http://www.slideshare.net/AlvaroPuentesMolina/03-los-datos-geograficos>.
- Reynoso, Carlos y Kiccillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de*. Buenos Aires : Universidad de Buenos Aires, 2004. V 1.0.
- Rumbaugh, J, Booch, G y Jacobson, I. 1999.** *El lenguaje unificado de modelado* . 1999.
- Rumbaugh, James. 2004.** *El proceso unificado de desarrollo de software*. La Habana : Félix Varela, 2004.
- Sanches Menendez, Fernando J. 2004.** *Geodesia Y Cartografía: Los Conceptos Y su Aplicacion Práctica*. 2004.
- Sanchez, María A. Mendoza. 2002.** *Metodologías De Desarrollo De Software*. 2002.
- serviciosTIC. 2006.** serviciosTIC. [En línea] 2006. [Citado el: 4 de diciembre de 2012.] <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.
- Sierra, Virginia y Alvarez, Carlos.** *Metodología de la Investigación Científica* .
- Sloccum, Jack. 2008.** *Introduccion a ExtJs*. 2008.
- Solís, Roberth Figueroa y camilo J.** *Metodologías Tradicionales VS. Metodologías Ágiles*. s.l. : Universidad Técnica Particular de Loja.
- Sommerville, Ian. 2005.** *Ingeniería de software* . 2005.
- . 1995. *Ingeniería del software* . 1995.
- Stair, Ralph M y Reynolds, George W. 1990.** *Sistemas de información geográfica : Introducción* . 1990.
- Urrutia, Javier. 2006.** *Cartografía Orientación y GPS*. s.l. : ETOR-OSTOA S.L, 2006.
- . 2006. *Cartografía, orientación y GPS* . 2006.
- Weitzenfeld, Alfredo. 2003.** *Ingeniería de software orientada a objetos con UML, Java e Internet* . 2003.
- Zayas, Carlos Alvarez de. 1995.** *Metodología de la Investigación Científica*. Santiago de Cuba : Centro de Estudios de Educación Superior, 1995. pág. 80.