



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



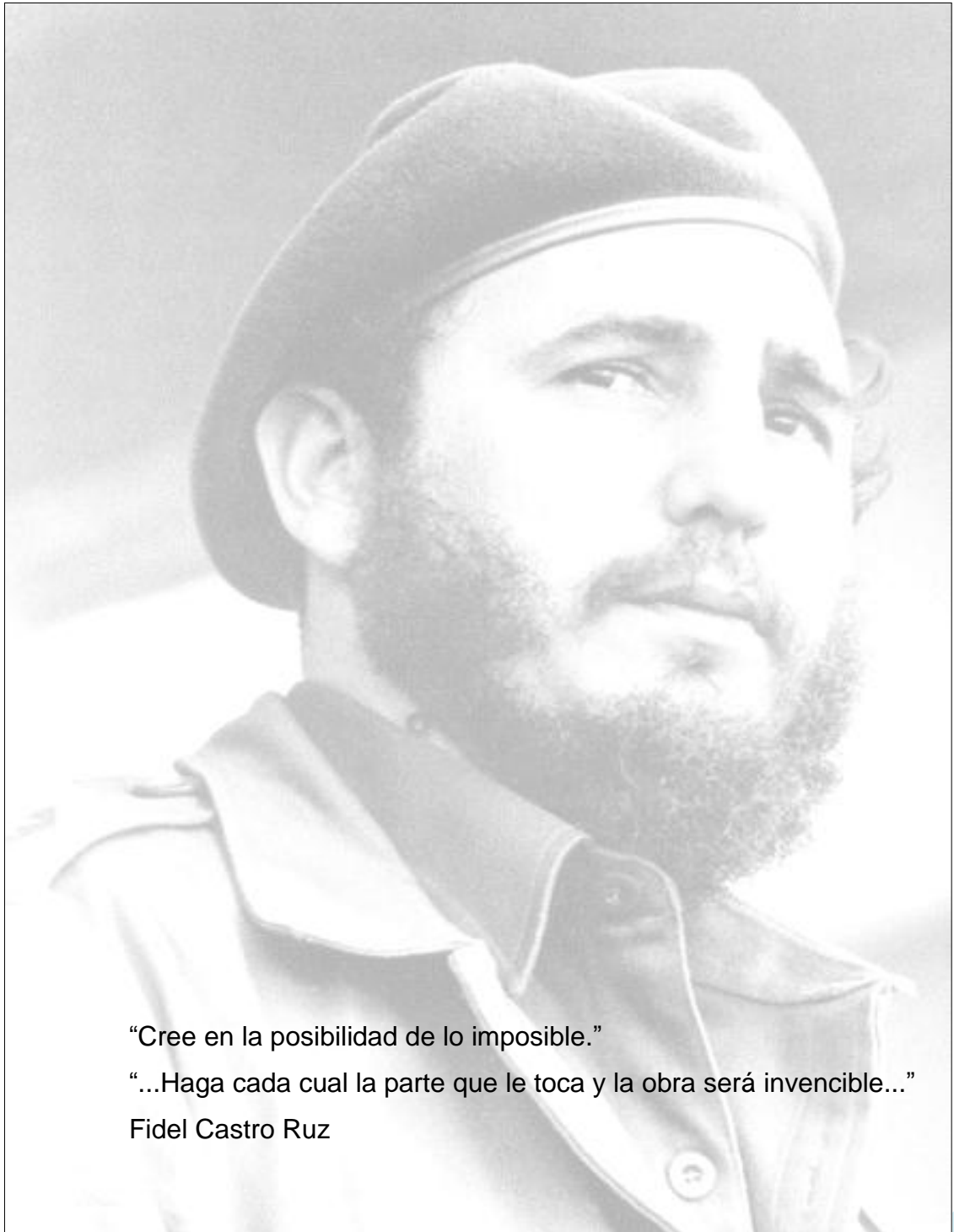
**Título: Servidor para el Módulo de la
Dirección de Ciencia, Tecnología y
Medio Ambiente (CITMA), de la
Administración Provincial
de Artemisa (APA).**

**Autor: Gabriela García Rivero
Tutores: Ing. Raisa Ortega Báez
Ing. Dayana Canova Ramírez**

Artemisa, Julio 2012



Frases...



“Cree en la posibilidad de lo imposible.”

“...Haga cada cual la parte que le toca y la obra será invencible...”

Fidel Castro Ruz

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo Facultad Regional “Mártires de Artemisa” de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Gabriela García Rivero Ing. Raisa Ortega Báez

Firma del Autor Firma del Tutor

Ing. Dayana Canova Ramírez

Firma del Co-Tutor

Dedicatoria

A mi abuela

por haber sido incondicional en todos los años de mi vida,

por haber sido tan incansable,

A mis amigos

que todos de una forma u otra me han apoyado en mi búsqueda

de ser una mejor persona,

A mis padres

que han hecho lo que han podido por mí.

Agradecimientos

A mi abuela, que siempre tuvo las palabras correctas para levantarme los ánimos, para ayudarme a encontrar la solución a mis problemas, para darme la chispa que encendía mi mente.

A mi abuelo Jesús, que siempre está presente en mis decisiones y logros, que lo extraño mucho.

A mi abuela Verónica, que me enseñó a conocer mejor mis orígenes y a ponerle empeño a mi carrera.

A mi mamá, quien ha apoyado todas mis decisiones y me ha dado cuanto ha podido.

A mi papá, a quien espeto.

A mis hermanas, que cada vez se crecen ante mis ojos, para mi seguirán siendo mis hermanitas.

A mis tíos, a mis primos, a mis tíos abuelos queridos, que aunque están lejos siempre aparecen con algo útil y llenan la casa de luz y alegría.

A Osiel y Olivia, gente de esas que aparecen siempre a levantarte cuando el mundo se cae a tu alrededor.

A mi novio Amed, que él sabe que lo quiero mucho y que en mucho de lo que he hecho en esta tesis él me ha apoyado y dado su ojo crítico.

A su familia, numerosa, carismática y unida, que sin haber sido avisada ya me consideraban parte de ella.

A mis amigas, mis brujitas, las dos Jessicas, Anabel, Lisandra y las jimaguas Aylín y Aymé.

A mis amigos, mi hermanito mayor Michel, Bruno, Willito, Alejandro, Yasser, Yaniel y Nelson.

A Yosmany, mi hermanito, mi mejor amigo, le deseo la mejor de las suertes en la vida. Para ti también es esta tesis.

A la gente de la Facultad Regional “Mártires de Artemisa” y de la UCI, que desde primero vienen dándome que hacer y dejándome ser incansable con ellos, dándoles consulta sin ser alumna ayudante, buscando el modo de estar mejor en esta facultad, jugando voleibol con ellos, divirtiéndonos, haciendo de niñera en ocasiones, ha sido mucho lo que hemos vivido juntos.

A los amigos que empezaron este viaje y ahora no se encuentran aquí, yo sé que están mejor en dondequiera que estén.

A mis compañeros de aula, a las chicas de mi cuarto, ha sido tanta la “quemadera” que sé que ninguna salió igual que como entró, algo aprendimos, hasta la que menos.

A mis rivales, que me han ayudado a formarme con mayor firmeza para lo que viene.

A Tania la Bibliotecaria.

A Yuray la profesora de Física, que mucho me ha escuchado y ayudado incontables veces, casi una mamá ha sido mientras estuve en la facultad.

A José Ramón y Juana, mis vicedecanos, tan atentos y amables,

no olvidaré lo que me ayudaron.

A mis profesores, a los que ya no están entre nosotros, a los que quedaron para vernos graduarnos, de mi barrio y de la facultad.

A mis tutoras, que tanto les he caído atrás y ellas a mí, muchos fueron los debates, pero para algo bueno ha sido.

A mis profesores del politécnico, ellos saben que yo no me iba a rendir tan fácil y sé que ellos están orgullosos de mí.

A todos mis vecinos que siempre han preguntado por mí y por mis estudios, siempre apoyándome para que no me rinda en este camino que elegí.

A la gente pequeña que llena mi mundo y que sé que aunque no estén todos los días ni todo el tiempo ayudaron a que terminara esta tesis, a que la adornara, a que la valorara y abrazara lo que han sido los años más difíciles de mi vida y también los más completos, los más bellos.

Y a todos los de una forma u otra me empujaron, soportaron, ayudaron o simplemente estuvieron ahí cuando los necesité, muchas gracias...

Resumen

A raíz de los cambios realizados en la estructura política-económica de Cuba es que surge el gobierno de Mayabeque y Artemisa, el último cuenta con una entidad que se encarga de la recopilación de información para su posterior utilización en la toma de decisiones de la naciente provincia. El presente trabajo consiste en el desarrollo de un sistema de gestión *web* que persigue el propósito de reemplazar el manejo manual y desorganizado de la información, logrando introducir el *software web* dentro de este mercado, con una alternativa libre.

El documento recoge, en un primer momento, el estudio del estado del arte y la investigación realizada. Luego se detalla el proceso de desarrollo de *software* seguido para crear la solución y los aspectos técnicos envueltos en la misma.

Palabras clave: CITMA, páginas *web*, sitio *web*, alternativa *web*, sistema de gestión de la información.

Índice General

<i>Declaración de Autoría.....</i>	<i>999</i>
<i>Dedicatoria.....</i>	<i>99</i>
<i>Agradecimientos.....</i>	<i>9</i>
<i>Resumen.....</i>	<i>9999</i>
<i>Índice General.....</i>	<i>9X</i>
<i>Introducción.....</i>	<i>2</i>
<i>Capítulo 1. Fundamentación Teórica.....</i>	<i>10</i>
<i>1.1. Principales conceptos y sus antecedentes.....</i>	<i>10</i>
<i>1.2. Análisis de los Sistemas de Gestión de la Información existentes.</i> <i>.....</i>	<i>16</i>
<i>1.2.1. Sistemas de Gestión de la Información en el sector del Medio</i> <i>Ambiente en el mundo.....</i>	<i>16</i>
<i>1.2.2. Aplicaciones de los SGI en el sector del Medio Ambiente en</i> <i>Cuba.....</i>	<i>16</i>
<i>1.2.3. Conclusiones del análisis de los sistemas.....</i>	<i>17</i>
<i>1.3. Estudio de las herramientas y lenguajes de desarrollo web.</i>	<i>18</i>
<i>1.3.1. Lenguajes usados para el desarrollo.....</i>	<i>19</i>
<i>1.3.2. Protocolos de comunicación.....</i>	<i>24</i>
<i>1.3.3. Frameworks de desarrollo.....</i>	<i>24</i>
<i>1.3.4. Herramientas usadas para el desarrollo.....</i>	<i>26</i>
<i>1.3.5. Otras herramientas.....</i>	<i>31</i>
<i>1.3.6. Metodología de Ingeniería de Software utilizada.....</i>	<i>31</i>
<i>1.3.7. Arquitectura de Software.....</i>	<i>35</i>

1.4. Conclusiones del capítulo.....	37
Capítulo 2: Características del Sistema.....	38
2.1. Descripción general de los procesos involucrados en el negocio...	38
2.2. Objeto de Automatización.....	39
2.3. Propuesta del sistema.....	40
2.2.1. Planificación del proyecto por Roles.....	40
2.2.2. Modelo de Dominio.....	41
2.2.3. Lista de Reserva del Producto.....	43
2.2.4. Historia de Usuario.....	45
2.2.5. Tareas de Ingeniería.....	49
2.2.6. Plan de Releases.....	51
2.2.7. Diagrama de Paquetes.....	52
2.2.8. Diagrama de Componentes.....	54
Capítulo 3: Adquisición y resultados del Sistema.....	58
3.1. Pruebas Unitarias.....	58
3.1.1. Pruebas de Caja Blanca.....	59
3.2. Resultados obtenidos.....	62
3.3. Funcionalidades obtenidas.....	63
3.4. Aporte social y económico.....	63
Conclusiones Generales.....	65
Recomendaciones.....	66
Referencias.....	67

Índice de Figuras

Fig. 1. Tabla comparativa entre los IDEs Netbeans y Eclipse....	27
--	-----------

<i>Figura 2. Arquitectura N-Capas.....</i>	<i>35</i>
<i>Figura 3. Ejemplos de Flujo de Control Lógico.....</i>	<i>60</i>
<i>Figura 4. Ejemplo de Diagramas de Flujo.....</i>	<i>61</i>

Índice de Tablas

<i>Tabla 2.1. Roles encargados de ejecutar la propuesta de SGI.</i>	<i>41</i>
<i>Tabla 2.2. Lista de Reserva del Producto. Ver Anexo 2.....</i>	<i>45</i>
<i>Tabla 2.3. Historia de Usuario. Reporte de Programas y Proyectos en Ejecución. Ver Anexo 3.....</i>	<i>49</i>
<i>Tabla 2.4. Tareas de Ingeniería. Reporte de Programas y Proyectos en Ejecución. Ver Anexo 4.....</i>	<i>51</i>
<i>Tabla 2.5. Plan de Releases.....</i>	<i>52</i>

Índice de Diagramas

<i>Diagrama 1. Modelo de Dominio de la propuesta de SGI.</i>	<i>42</i>
<i>Diagrama 2. Diagrama de Paquetes de la propuesta de SGI.....</i>	<i>53</i>
<i>Diagrama 3. Diagrama de Componentes de la propuesta de SGI.....</i>	<i>56</i>

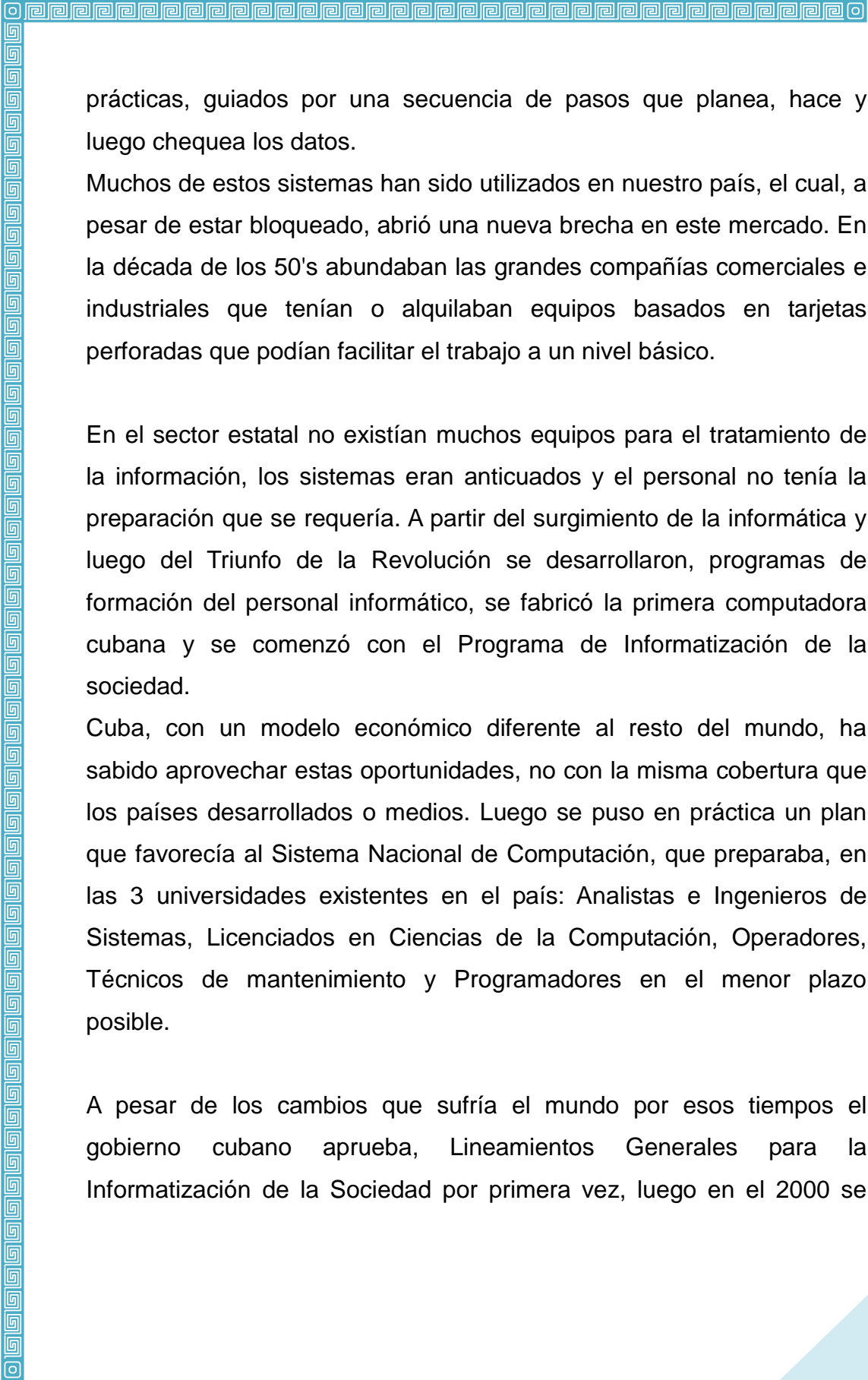
Introducción

En la sociedad desde tiempos inmemoriales se trató con datos, fechas y acontecimientos que requerían ser almacenados y publicados y que al pasar el tiempo pudieran ser nuevamente referenciados o consultados. Con el paso de los años se buscaron nuevos métodos y sistemas capaces de controlar mejor la información.

Aparecen nuevos usos para la información como: mejorar o aumentar el conocimiento humano del usuario, o dicho de otra forma disminuir la ignorancia existente, brindar apoyo a la toma de decisiones como base para la formulación de soluciones, extender una serie de reglas de evaluación y decisión para fines de control; todo esto debido a que la información podía ser obtenida y modificada con mayor facilidad.

Para el tratamiento de dicha información se requerían horas de escritura y organización manual. Sin embargo con el surgimiento de la Informática y sus tecnologías se desarrollaron las Tecnologías de la Información y las Comunicaciones (TIC), los *software* para la gestión de los datos, los cursos y carreras que aseguraban especialistas en este nuevo campo.

Surgen además, los Sistemas de Gestión de la Seguridad de la Información (SGSI) o “Information Security Management System (ISMS)” de la traducción al inglés. Herramientas que han devenido en un gran avance en sistemas informáticos modernos. Dichos sistemas deben estar concebidos sobre la base de los cambios que sufre la informática actual, de ahí que algunos sean basados en buenas



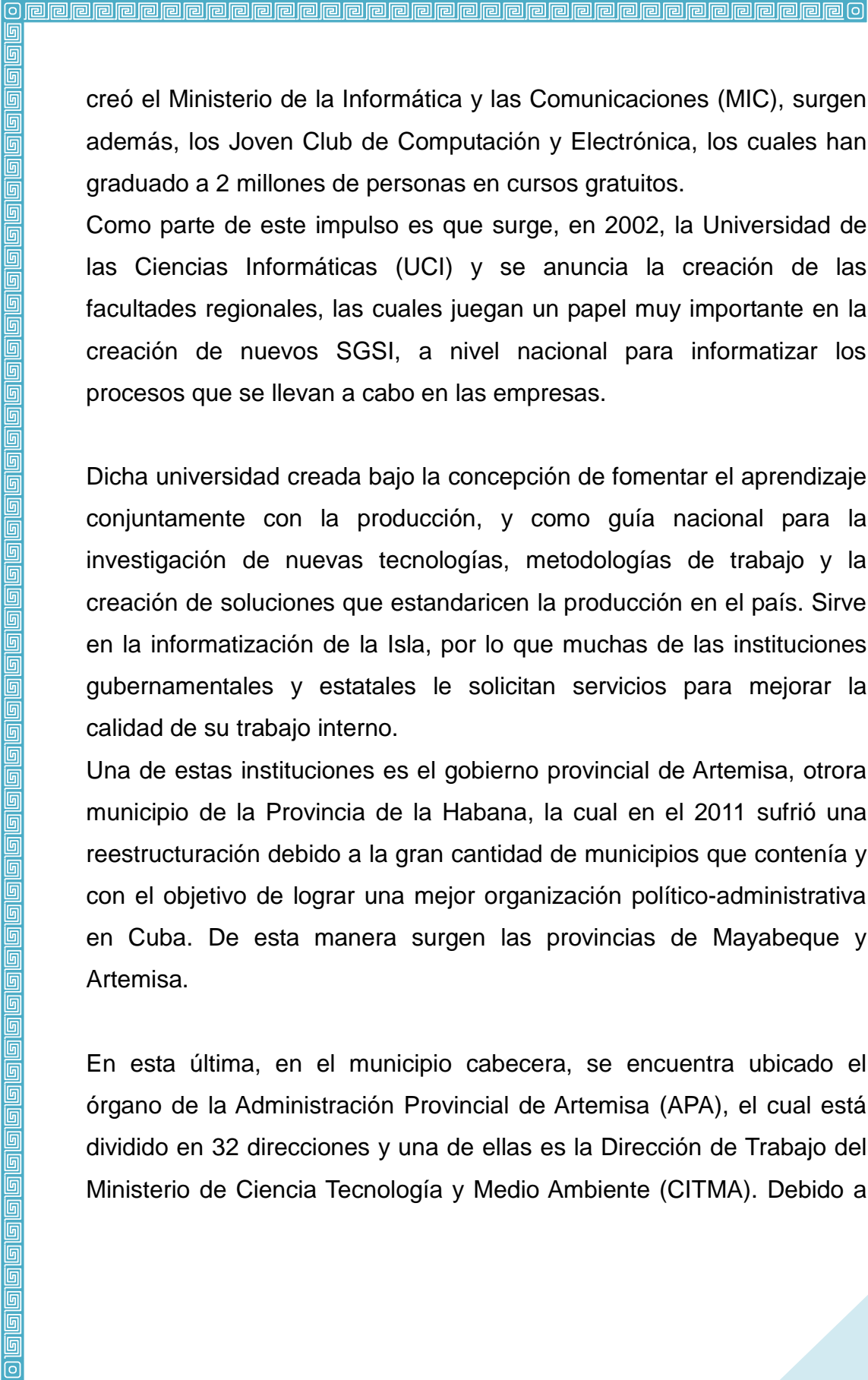
prácticas, guiados por una secuencia de pasos que planea, hace y luego chequea los datos.

Muchos de estos sistemas han sido utilizados en nuestro país, el cual, a pesar de estar bloqueado, abrió una nueva brecha en este mercado. En la década de los 50's abundaban las grandes compañías comerciales e industriales que tenían o alquilaban equipos basados en tarjetas perforadas que podían facilitar el trabajo a un nivel básico.

En el sector estatal no existían muchos equipos para el tratamiento de la información, los sistemas eran anticuados y el personal no tenía la preparación que se requería. A partir del surgimiento de la informática y luego del Triunfo de la Revolución se desarrollaron, programas de formación del personal informático, se fabricó la primera computadora cubana y se comenzó con el Programa de Informatización de la sociedad.

Cuba, con un modelo económico diferente al resto del mundo, ha sabido aprovechar estas oportunidades, no con la misma cobertura que los países desarrollados o medios. Luego se puso en práctica un plan que favorecía al Sistema Nacional de Computación, que preparaba, en las 3 universidades existentes en el país: Analistas e Ingenieros de Sistemas, Licenciados en Ciencias de la Computación, Operadores, Técnicos de mantenimiento y Programadores en el menor plazo posible.

A pesar de los cambios que sufría el mundo por esos tiempos el gobierno cubano aprueba, Lineamientos Generales para la Informatización de la Sociedad por primera vez, luego en el 2000 se



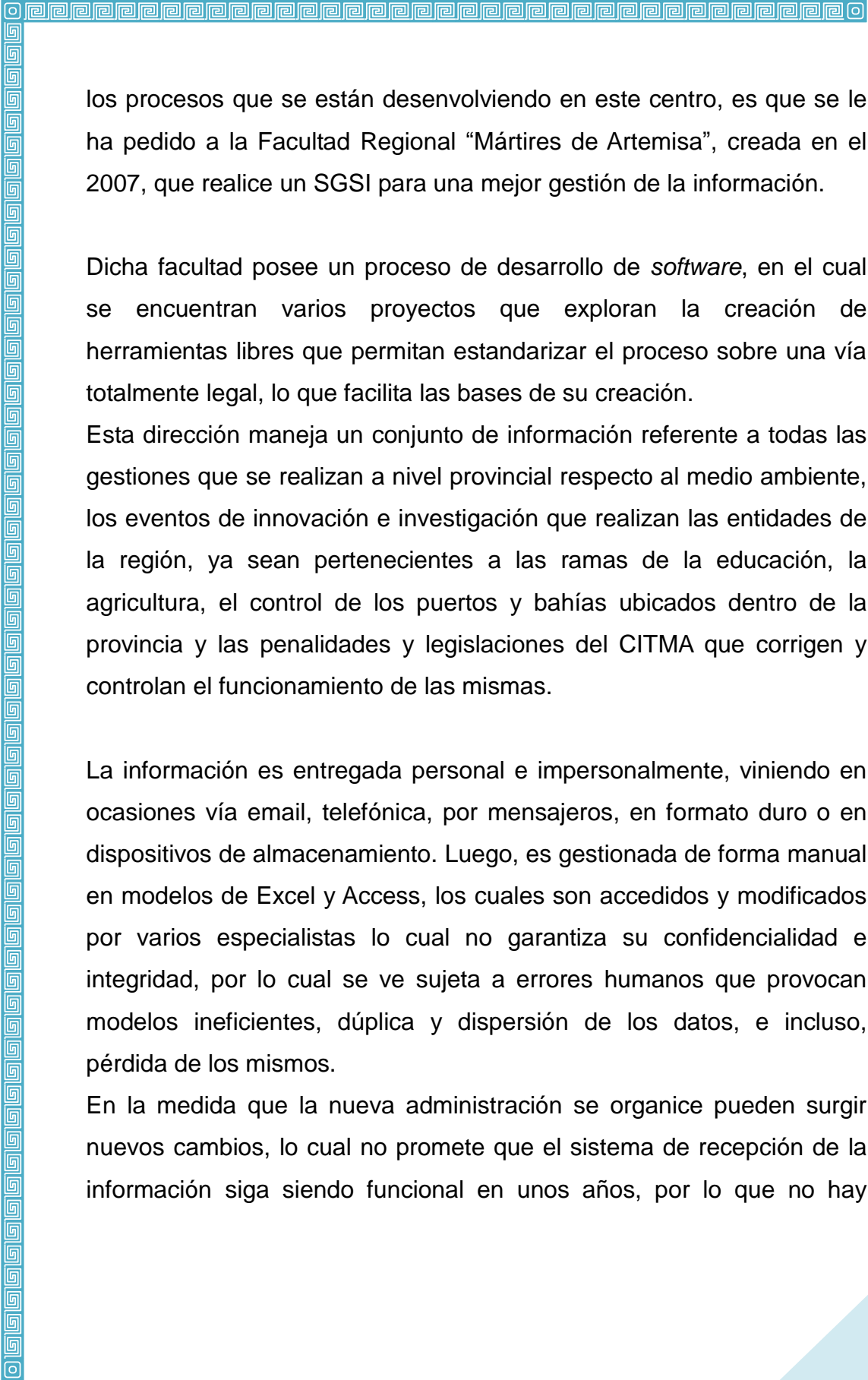
creó el Ministerio de la Informática y las Comunicaciones (MIC), surgen además, los Joven Club de Computación y Electrónica, los cuales han graduado a 2 millones de personas en cursos gratuitos.

Como parte de este impulso es que surge, en 2002, la Universidad de las Ciencias Informáticas (UCI) y se anuncia la creación de las facultades regionales, las cuales juegan un papel muy importante en la creación de nuevos SGSI, a nivel nacional para informatizar los procesos que se llevan a cabo en las empresas.

Dicha universidad creada bajo la concepción de fomentar el aprendizaje conjuntamente con la producción, y como guía nacional para la investigación de nuevas tecnologías, metodologías de trabajo y la creación de soluciones que estandaricen la producción en el país. Sirve en la informatización de la Isla, por lo que muchas de las instituciones gubernamentales y estatales le solicitan servicios para mejorar la calidad de su trabajo interno.

Una de estas instituciones es el gobierno provincial de Artemisa, otrora municipio de la Provincia de la Habana, la cual en el 2011 sufrió una reestructuración debido a la gran cantidad de municipios que contenía y con el objetivo de lograr una mejor organización político-administrativa en Cuba. De esta manera surgen las provincias de Mayabeque y Artemisa.

En esta última, en el municipio cabecera, se encuentra ubicado el órgano de la Administración Provincial de Artemisa (APA), el cual está dividido en 32 direcciones y una de ellas es la Dirección de Trabajo del Ministerio de Ciencia Tecnología y Medio Ambiente (CITMA). Debido a



los procesos que se están desarrollando en este centro, es que se le ha pedido a la Facultad Regional “Mártires de Artemisa”, creada en el 2007, que realice un SGSI para una mejor gestión de la información.

Dicha facultad posee un proceso de desarrollo de *software*, en el cual se encuentran varios proyectos que exploran la creación de herramientas libres que permitan estandarizar el proceso sobre una vía totalmente legal, lo que facilita las bases de su creación.

Esta dirección maneja un conjunto de información referente a todas las gestiones que se realizan a nivel provincial respecto al medio ambiente, los eventos de innovación e investigación que realizan las entidades de la región, ya sean pertenecientes a las ramas de la educación, la agricultura, el control de los puertos y bahías ubicados dentro de la provincia y las penalidades y legislaciones del CITMA que corrigen y controlan el funcionamiento de las mismas.

La información es entregada personal e impersonalmente, viniendo en ocasiones vía email, telefónica, por mensajeros, en formato duro o en dispositivos de almacenamiento. Luego, es gestionada de forma manual en modelos de Excel y Access, los cuales son accedidos y modificados por varios especialistas lo cual no garantiza su confidencialidad e integridad, por lo cual se ve sujeta a errores humanos que provocan modelos ineficientes, duplica y dispersión de los datos, e incluso, pérdida de los mismos.

En la medida que la nueva administración se organice pueden surgir nuevos cambios, lo cual no promete que el sistema de recepción de la información siga siendo funcional en unos años, por lo que no hay

confiabilidad de los datos.

Posteriormente la información es analizada por una comisión y utilizada en la Toma de Decisiones respecto a la contaminación de las reservas de la biosfera, el vertimiento de residuos en áreas aledañas a ríos y mares, las inundaciones costeras durante la temporada ciclónica, la degradación de los suelos, así como el seguimiento y organización del Fórum de Ciencia y Técnica.

De este modo, se ve afectada la organización y centralización de la información lo cual retrasa la entrega de la misma y su posterior utilización en la toma de decisiones por parte del presidente y de la Dirección del CITMA del APA.

Teniendo en cuenta las dificultades descritas anteriormente se define como **problema de investigación**:

- ¿Cómo optimizar la gestión de la información que permita la eficacia de los datos en la Dirección CITMA de la Administración Provincial de Artemisa?

En función del problema de investigación identificado y con el objetivo de alcanzar una solución satisfactoria se determina como **objeto de estudio** de la presente investigación el **Proceso de Gestión de la Información** y como **campo de acción** los **Sistemas de Gestión de la Información en el sector del Medio Ambiente**.

Para dar solución a lo anteriormente expuesto se precisa como **objetivo general** del presente trabajo:

- Desarrollar el Servidor de la Dirección CITMA que optimice la gestión de la información en el AP de Artemisa.

Derivándose de los objetivos planteados anteriormente las siguientes **preguntas científicas:**

1. ¿Cuáles son los fundamentos teóricos de los sistemas de gestión de información?
2. ¿Cuál es el diagnóstico del estado actual de los sistemas de gestión de información en el sector del medio ambiente?
3. ¿Cómo implementar un servidor con las metodologías, herramientas y tecnologías a utilizar?
4. ¿Cómo validar con pruebas unitarias los requerimientos del servidor?

Para responder dichas preguntas, se plantearon las siguientes **tareas de investigación:**

1. Elaboración de la fundamentación teórica de la investigación.
2. Identificación de los procesos y requerimientos de la Dirección del CITMA de la Administración Provincial de Artemisa.
3. Implementar los eventos (metainformación), PlugIns y páginas xml en el servidor que dan soporte para atender los requerimientos de la aplicación cliente.
4. Validar mediante pruebas funcionales los resultados obtenidos de la implementación.

Para realizar las tareas antes propuestas se definieron métodos teóricos. Se emplearon para el conocimiento del funcionamiento de la

Dirección del CITMA de la Administración Provincial de Artemisa, además de sus antecedentes y procesos internos, el método **Histórico-lógico**.

Donde el método **Lógico** consiste en inferir de la semejanza de algunas características entre dos objetos, la probabilidad de que las características restantes sean también semejantes. Los razonamientos analógicos no son siempre válidos; mientras que el método **Histórico** analiza la trayectoria concreta de la teoría, su condicionamiento a los diferentes períodos de la historia.

Y para la explicar concretamente y mediante modelos, el **Método de Modelación**, que opera en forma práctica o teórica con un objeto, no en forma directa, sino utilizando cierto sistema intermedio, auxiliar, natural o artificial. [1]

Así como se analizaron algunos **métodos empíricos** los cuales permiten efectuar el análisis preliminar de la información, así como verificar y comprobar las concepciones teóricas. Dentro de estos, se tomó la **entrevista**, con muestreo intencional, dado que se realizaron encuentros en el gobierno con 4 dirigentes de la Dirección CITMA del APA para el levantamiento de información en la etapa inicial, lo que posibilitó un mejor entendimiento del problema existente en dicha dirección. Ver Anexo 1.

Además, se utilizó el método de **Análisis Documental** para la recopilación de los reportes a informatizar. Además, es una técnica de representación del contenido de los documentos en un sistema documental realizado, principalmente, para que puedan ser

recuperados cuando sean necesarios.

Al finalizar el presente trabajo los involucrados en este proceso esperan obtener como **posible resultado**:

- Servidor para la Dirección CITMA de la Administración Provincial de Artemisa.

El presente trabajo está estructurado de la siguiente forma: Introducción, Capítulo 1, Capítulo 2, Capítulo 3, Conclusiones, Referencias Bibliográficas, Bibliografía, Webgrafía y Anexos. Los capítulos están conformados como se muestra a continuación:

Capítulo 1: Fundamentación Teórica.

Se hace un análisis del estado del arte del objeto de estudio, se investiga acerca de los sistemas informáticos vinculados al campo de acción, se fundamentan las metodologías, tecnologías y herramientas utilizadas para el desarrollo del sistema de gestión.

Capítulo 2: Características, análisis y diseño del sistema.

Se define el negocio y se describe la solución propuesta para la situación problemática. Se presentan las características y funcionalidades del sistema a partir de los requisitos funcionales y no funcionales capturados. Se realiza el análisis y diseño del sistema.

Capítulo 3: Implementación y validación de los resultados del sistema.

Incluye la programación realizada a partir de los requerimientos y los diagramas del diseño elaborados, así como las métricas y pruebas utilizadas para la validación de la misma.

Capítulo 1. Fundamentación Teórica.

A continuación se ofrece una panorámica a las definiciones y conceptos relacionados con la solución que se propone, que han sido implementados para entender las bases desde la cual ha sido concebida. Se fundamenta la utilización de la metodología, las herramientas, los patrones y lenguajes, sus características y alcance, y los antecedentes considerados para lograr el desarrollo del producto final.

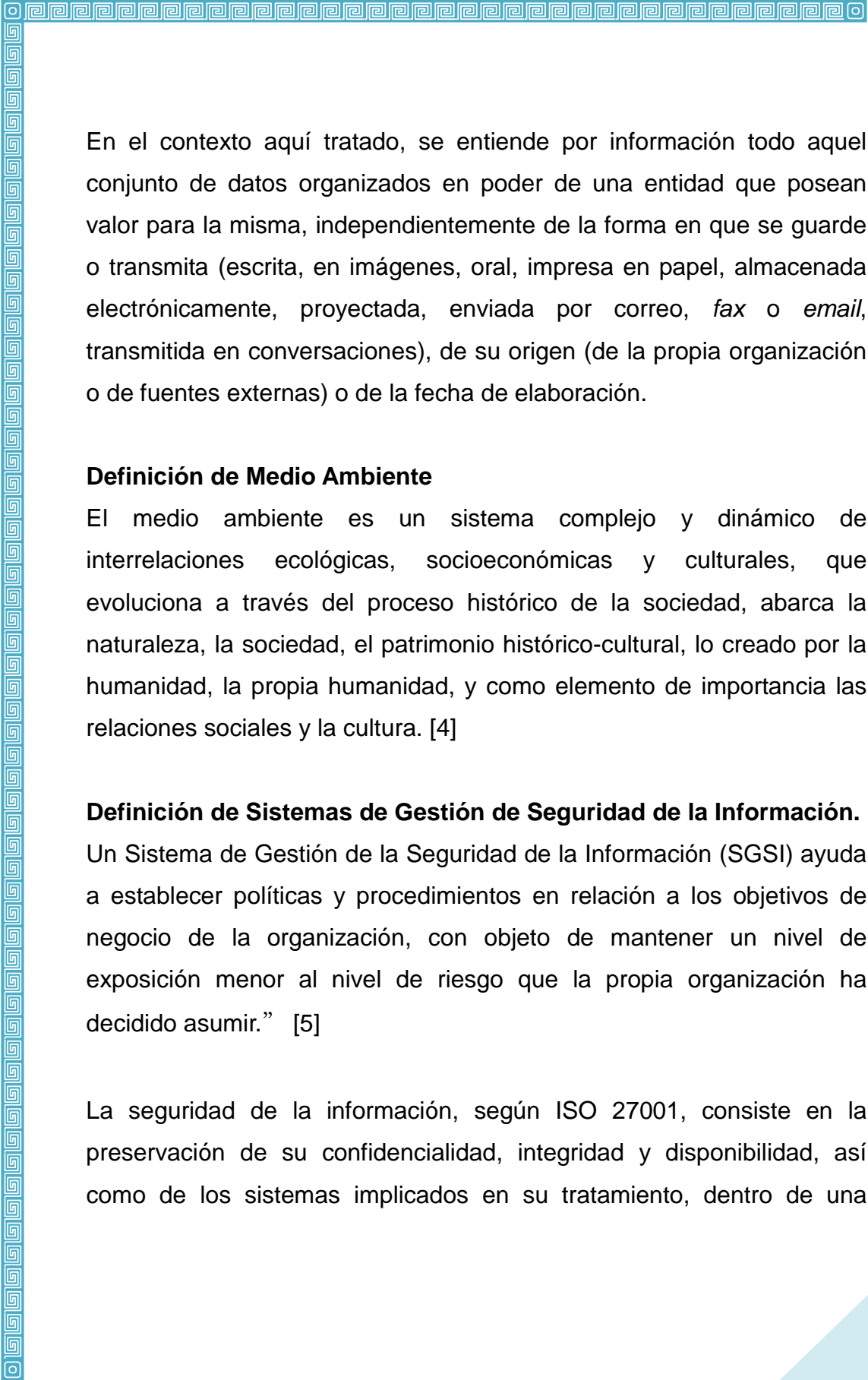
1.1. Principales conceptos y sus antecedentes.

El **Sistema de Gestión de la Información** que se propone es un **sitio web** compuesto por un **servidor** que optimiza la gestión de los datos en el sector del **medio ambiente**, realizado con **herramientas distribuidas bajo licencias libres**. Bajo esta idea se fundamentan las siguientes definiciones:

Definición de Información

Según Idalberto Chiavenato, **información** "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones". [2]

Para *Ferrell y Hirt*, la **información** "comprende los datos y conocimientos que se usan en la toma de decisiones". [3]



En el contexto aquí tratado, se entiende por información todo aquel conjunto de datos organizados en poder de una entidad que posean valor para la misma, independientemente de la forma en que se guarde o transmita (escrita, en imágenes, oral, impresa en papel, almacenada electrónicamente, proyectada, enviada por correo, *fax* o *email*, transmitida en conversaciones), de su origen (de la propia organización o de fuentes externas) o de la fecha de elaboración.

Definición de Medio Ambiente

El medio ambiente es un sistema complejo y dinámico de interrelaciones ecológicas, socioeconómicas y culturales, que evoluciona a través del proceso histórico de la sociedad, abarca la naturaleza, la sociedad, el patrimonio histórico-cultural, lo creado por la humanidad, la propia humanidad, y como elemento de importancia las relaciones sociales y la cultura. [4]

Definición de Sistemas de Gestión de Seguridad de la Información.

Un Sistema de Gestión de la Seguridad de la Información (SGSI) ayuda a establecer políticas y procedimientos en relación a los objetivos de negocio de la organización, con objeto de mantener un nivel de exposición menor al nivel de riesgo que la propia organización ha decidido asumir.” [5]

La seguridad de la información, según ISO 27001, consiste en la preservación de su confidencialidad, integridad y disponibilidad, así como de los sistemas implicados en su tratamiento, dentro de una

organización. Así pues, estos tres términos constituyen la base sobre la que se soporta todo el edificio de la seguridad de la información:

- **Confidencialidad:** la información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.
- **Integridad:** mantenimiento de la exactitud y completitud de la información y sus métodos de proceso.
- **Disponibilidad:** acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos, entidades o procesos autorizados cuando lo requieran.

Este proceso es el que constituye un SGSI, que podría considerarse, por analogía con una norma tan conocida como ISO 9001, como el sistema de calidad para la seguridad de la información.

El propósito de un sistema de gestión de la seguridad de la información es, por tanto, garantizar que los riesgos de la seguridad de la información sean conocidos, asumidos, gestionados y minimizados por la organización de una forma documentada, sistemática, estructurada, repetible, eficiente y adaptada a los cambios que se produzcan en los riesgos, el entorno y las tecnologías. [6]

Definición de Sistema de Gestión de la Información.

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. Un sistema de gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

El uso de un sistema de gestión probado le permite renovar

constantemente su objetivo, sus estrategias, sus operaciones y niveles de servicio. [7]

Un Sistema de Gestión de la Información puede estar constituido por un sitio *web* o por una aplicación de escritorio, que trabaje con una base de datos y un almacén para respaldar la información allí manejada. A continuación se exponen los conceptos que tienen que ver con la solución de esta investigación.

Definición de Sitio *Web*

Un sitio *web*, es un espacio virtual en Internet. Se trata de un conjunto de páginas *web* que son accesibles desde un mismo dominio o subdominio de la *World Wide Web (WWW)*.

Los sitios *web* incluyen documentos HTML, fotografías, sonidos, vídeos, animaciones *Flash* y otro tipo de contenidos que pueden compartirse en línea. La URL raíz del sitio *web* se conoce como portada o *homepage*. Lo más habitual es que esta portada facilite el acceso a todas las páginas del sitio mediante hipervínculos (también conocidos como enlaces o *links*).

Para acceder a un sitio *web* se necesita contar con un navegador en la computadora, el cual es un programa que nos permite recorrer páginas *web*, que puede ser *Internet Explorer, Mozilla Firefox, Safari, Opera* y *Google Chrome*, entre otros. [8]

Para la creación de sitios *webs* que constituyan Sistemas de Gestión de la Información se pueden utilizar diversos lenguajes y *software* privativos o libres, de acuerdo a las normas de desarrollo determinadas

en la fase de inicio del proyecto, conocidos como IDEs, quienes, básicamente, son un Ambiente o Entorno de Desarrollo Integrado (del inglés “*Integrated Development Enviroment*”) y lenguajes de programación, a continuación se definen los mismos.

Definición de Servidor

Un servidor es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. [9]

Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor. [10]

Definición de Software Libre

Hace referencia a la libertad de los usuarios o desarrolladores de tener acceso libre al código fuente para copiarlo, leerlo, distribuirlo, cambiarlo y mejorarlo a su gusto. Existen diferentes variantes. La principal, promovida por la *Free Software Foundation*, es la más utilizada en el mundo, respaldada legalmente por la licencia GNU/GPL (del inglés “General Public License”). Existen otras variantes de código abierto como Mozilla Public License, L-GPL. [11]

El *software* libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el *software*. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales:

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3).

Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello. [12]

Definición de *Software* Privativo

Se refiere a todo aquel *software* o fragmento de *software* cuya licencia comercial, de uso o de publicación no permite compartir el código fuente del mismo o, lo hace, pero restringe las libertades de uso del mismo. [11]

Se le llama *software* propietario, no libre, privado o privativo al tipo de programas informáticas o aplicaciones en el que el usuario no puede acceder al código fuente o tiene un acceso restringido y, por tanto, se ve limitado en sus posibilidades de uso, modificación y redistribución. [13]

1.2. Análisis de los Sistemas de Gestión de la Información existentes.

Debido a la gran demanda de sistemas automatizados para la gestión de la información es que se han venido creando nuevos *software*, capaces de controlar y garantizar la seguridad e integridad de los datos en las empresas que los han solicitado. A continuación se hace referencia a algunos de ellos.

1.2.1. Sistemas de Gestión de la Información en el sector del Medio Ambiente en el mundo.

No se encontraron Sistemas de Gestión de Información en este sector a nivel internacional.

1.2.2. Aplicaciones de los SGI en el sector del Medio Ambiente en Cuba.

Sistema de Gestión de Información en CEMARNA (Centro de Estudios del Medio Ambiente de la Universidad de Pinar del Río).

Ajustado a las necesidades reales de un cliente (CEMARNA), y tomando como referente para su estudio la Metodología AMIGA y otros trabajos correspondientes con la escuela española de gestión de información. Uno de los objetivos de este sistema es posibilitar un mejor desempeño informacional en la actividad investigativa del colectivo de investigadores de dicho centro de estudio. Para lo cual se realizó en primera instancia un diagnóstico que revelo la situación polémica existente, la inexistencia de habilidades en la gestión de información en

apoyo a la investigación y en segundo lugar la identificación de los principales flujos de información existentes dentro del grupo. Todo lo cual permitió orientar los principales objetivos del trabajo en la propuesta de diseño del Sistema de Gestión de Información para la Gestión del Conocimiento en el CEMARNA. [14]

1.2.3. Conclusiones del análisis de los sistemas.

El sistema encontrado permite el uso eficiente de la información, como medio de control y posibilitan toma de decisiones inteligentes y visión de futuro. Además de reflejar cuanto han avanzado los sistemas de gestión de la información en relación con la informatización de la sociedad. Sin embargo dicha aplicación no abarca todo el contenido que puede estar vinculado a la toma de decisiones en la Administración Provincial del Gobierno de Artemisa ni cuenta con todas las funcionalidades y prestaciones que aporta nuestra propuesta. Además, son meramente informativas o están desarrolladas con el uso de metodologías, herramientas y lenguajes que no se ajustan a las necesidades del cliente.

Es necesario apoyarse en las oportunidades que ofrecen muchos de los servicios y funcionalidades que brindan los portales *web* analizados. Dentro de ellos se puede destacar la sección de Noticias, de modo que favorece a que el usuario se mantenga al tanto del acontecer nacional y territorial. Contar con enlaces a los sitios de diferentes organizaciones. Brindar mayor cantidad de información centralizada, permitir a los usuarios ingresar información y modificarla.

La aplicación resultante de esta tesis soporta la gestión de los usuarios, evitando problemas de seguridad en la información y regulando la

accesibilidad a la misma.

1.3. Estudio de las herramientas y lenguajes de desarrollo web.

La gestión de la información se ve beneficiada cuando es controlada mediante un único sistema, donde todos los datos sean manejados de forma íntegra, confidencial y esté al alcance de todos los que la necesiten.

Como mismo se dijo antes, estos sistemas pueden ser aplicaciones de escritorio o *web*, sin embargo, se utiliza la alternativa *web* por las siguientes razones:

- Puede ser accedido desde cualquier computadora con conexión de red.
- Puede ser visualizado por cualquier navegador.
- Está bajo licencia de *software* libre por lo que su construcción es legal.
- Puede abrirse desde cualquier sistema operativo.
- No se limita a la RAM, el espacio en el Disco Duro, memoria de video, sino que con escasos recursos de *hardware* es capaz de ejecutar el código y mostrar al usuario la información que necesita. [15]

Para la implementación se consideró utilizar **Java** como lenguaje del lado del servidor puesto que se sustenta sobre cinco características fundamentales: la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en diversos sistemas operativos, la inclusión por defecto de soporte para trabajo en red, la opción de

ejecutar del código en sistemas remotos de manera segura y la facilidad de uso.

1.3.1. Lenguajes usados para el desarrollo.



Java

Las aplicaciones del lenguaje **Java** en la actualidad son numerosas y variadas. El lenguaje no solo se utiliza en aplicaciones de escritorio sino también en una gran variedad de dispositivos móviles, como teléfonos y pequeños electrodomésticos. En los navegadores *web*, **Java** permite desarrollar pequeñas aplicaciones conocidas como applets que se incrustan en el código HTML de las páginas. El navegador debe contar con un plugin que permita ejecutar las aplicaciones Java.

En resumen, Java permite:

- Simple:** Java fue diseñado para ser de rápido entendimiento y es por lo tanto, fácil de escribir, compilar, depurar y aprender, más que otros lenguajes de programación.
- Orientado a Objetos:** Esto le permite crear programas modulares y código reutilizable.
- Independiente de la plataforma:** Una de las ventajas más importantes de Java es su capacidad para moverse fácilmente de un sistema informático a otro.
- Distribuido:** Java está diseñado para hacer que la informática distribuida sea fácil con la capacidad de red que es inherentemente integrado en el propio lenguaje. La escritura de programas de red en Java es como enviar y recibir datos hacia y desde un archivo.

- **Seguro:** El lenguaje Java, el compilador, el intérprete, y el entorno de ejecución de cada uno se desarrolló con la seguridad en mente.
- **Robusto:** Java pone mucho énfasis en el control temprano de posibles errores, como compiladores de Java son capaces de detectar muchos problemas que en primer lugar aparecen durante el tiempo de ejecución en otros lenguajes programables.
- **Multiproceso:** Multiproceso es la capacidad de un programa para realizar varias tareas al mismo tiempo dentro de un programa. En Java, la programación de multiproceso se ha integrado progresivamente.
- Está bajo licencias libres, por lo que los usuarios no tienen que luchar con los impuestos sobre patente cada año.
- El poder de las API (librerías) de Java es alcanzada fácilmente por los desarrolladores.
- Java utiliza el recolector de basura de las ayudas inmediatamente, o sea, la gerencia de memoria es automática.
- Java asigna siempre objetos en el apilado.
- Java abrazó el concepto de especificaciones de la excepción.
- Es un lenguaje Multiplataforma y ayuda para la tela-servicio.
- Usando Java se pueden desarrollar aplicaciones *web* dinámicas.
- Permite crear programas modulares y códigos reutilizables. [16]

Luego se consideró utilizar algún IDE multiplataforma para el trabajo en la aplicación *web* con Java, por lo que se consideraron el Netbeans y el Eclipse, más adelante se hace una comparación a ambos IDEs y el

resultado de la misma.

Para un mejor entendimiento: un IDE es una herramienta que integra un conjunto de aplicaciones para crear otros *software* u otro tipo de contenidos digitales, ya sean imágenes, audios y videos. [11]

Para la programación del lado del cliente, se seleccionaron lenguajes compatibles con Java, que están en la mayoría de los sitios *web* actualmente publicados a nivel internacional, como son *JavaScript*, *jQuery*, *CSS3* y *HTML5*.



JavaScript

Como síntesis se puede decir que *JavaScript* es un lenguaje interpretado, basado en prototipos, mientras que Java es un lenguaje más orientado a objetos. Es soportado por la mayoría de los navegadores como *Internet Explorer*, *Netscape*, *Opera*, *Mozilla Firefox*, entre otros.

Se debe tener en cuenta que aunque *JavaScript* sea soportado en gran cantidad de navegadores nuestros usuarios pueden elegir la opción de Activar/Desactivar el *JavaScript* en los mismos.



HTML5

El HTML o *HyperText Markup Language*, tiene dos características esenciales, el hipertexto y la universalidad. Hipertexto significa que puede crear un vínculo en una página *Web* que lleve al visitante a cualquier otra página *Web* o prácticamente a

cualquier cosa en Internet.

Ahora convendría explicar qué es exactamente HTML 5, ya que no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo *web*. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá.

Así pues, **HTML 5** es una nueva versión de diversas especificaciones, entre las que se encuentran:

- HTML 4
- XHTML 1
- DOM Nivel 2 (DOM = *Document Object Model*)

A la par, HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones *web* más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Para ello se están creando unas APIs que permitan trabajar con cualquiera de los elementos de la página y realizar acciones que hasta hoy era necesario realizar por medio de tecnologías accesorias.

Estas API, que tendrán que ser implementadas por los distintos navegadores del mercado, se están documentando con minuciosidad, para que todos los *Browsers*, creados por cualquier compañía las soporten tal cual se han diseñado. Esto se hace con la intención que no ocurra lo que viene sucediendo en el pasado, que cada navegador hace la guerra por su parte y los que acaban pagándolo son los desarrolladores y a la postre los usuarios, que tienen muchas

posibilidades de acceder a *webs* que no son compatibles con su navegador preferido.

No sólo se trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías. *HTML 5* incluye novedades significativas en diversos ámbitos: estructura del cuerpo, etiquetas para contenido específico, *canvas*, bases de datos locales, *web workers*, aplicaciones *web offline*, geolocalización, nuevas *APIs* para interfaz de usuario y fin de las etiquetas de presentación. [18]



CSS3

Mientras que *HTML* nos permite definir la estructura de una página *web*, las hojas de estilo en cascada (*Cascading Style Sheets* o **CSS**) son las que nos ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos *web*.

Actualmente en definición, la nueva versión **CSS3**, nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño *web* actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, **CSS3** es el estándar que dominará la *web* por los siguientes años.

Entre sus ventajas están: capacidades visuales avanzadas y transformaciones, hojas de estilo aural, sombras, color, bordes y fondos, mejor soporte para fuentes, transición, cajas, contenido, fuentes *web*, múltiples columnas, selectores, *rgba*, múltiples imágenes de fondo

y textos sombreados. [19]

1.3.2. Protocolos de comunicación

Protocolo *WebSocket*

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-dúplex* sobre un único *socket* TCP. Está diseñada para ser implementada en navegadores y servidores *web*, pero puede utilizarse por cualquier aplicación cliente/servidor. La API de *WebSocket* está siendo normalizada por el W3C, y el protocolo *WebSocket*, a su vez, está siendo normalizado por el IETF. Como las conexiones TCP ordinarias sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios *WebSocket* sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

1.3.3. Frameworks de desarrollo.



JWebSocket

Este *framework* será utilizado para facilitar el desarrollo y funcionamiento de la aplicación. Contando con una gran variedad de funcionalidades tanto para la conectividad como para las propias funcionalidades y sistemas de seguridad por el lado del servidor.



JavaScriptMVC

Este *framework* será utilizado para facilitar el desarrollo y funcionamiento de la aplicación. Contando con una gran variedad de funcionalidades tanto para la conectividad como para las propias funcionalidades y sistemas de seguridad por el lado del cliente. [20]



JQuery

En la integración del proyecto por el lado del cliente, se empleó además *jQuery*, el cual es un *framework* para el lenguaje *JavaScript*, luego será un producto que nos simplificará la vida para programar en este lenguaje. Como probablemente se sabe, cuando un desarrollador tiene que utilizar *JavaScript*, generalmente tiene que preocuparse por hacer scripts compatibles con varios navegadores y para ello tiene que incorporar mucho código que lo único que hace es detectar el browser del usuario, para hacer una u otra cosa dependiendo de si es *Internet Explorer*, *Firefox* y *Opera*, entre otros. *jQuery* es donde más nos puede ayudar, puesto que implementa una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales.

Cuando se programa *JavaScript* con *jQuery* se tendrá a disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que dé seguridad de que funcionarán para todos los visitantes. Simplemente se deben conocer las librerías del *framework* y programar utilizando las clases, sus propiedades y métodos para la

consecución de los objetivos.

Además, todas estas ventajas que sin duda son muy de agradecer, con *jQuery* se obtienen de manera gratuita, ya que el *framework* tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Para ello, simplemente se incluyen en nuestras páginas un *script JavaScript* que contiene el código de *jQuery*, que se puede descargar de la propia página *web* del producto y comenzar a utilizar el *framework*. [21]

1.3.4. Herramientas usadas para el desarrollo.

A continuación se establecen las diferencias y ventajas de los IDEs anteriormente mencionados:

En ambos casos, un gran número de funciones se proporcionan: fuera de la caja, de un marco de acoplamiento, a un sistema de acción, para actualizar las instalaciones, y mucho más. En ambos casos, un rico conjunto de APIs (librerías) se otorgan, junto con muchos tutoriales y preguntas frecuentes, y varios libros.

Criterio	Netbeans IDE	Eclipse IDE
<i>Kit</i> de herramientas de interfaz de usuario	Herramientas <i>Swing</i> estándar.	SWT
Diseño de interfaz de usuario	Premios gratis, ganar, <i>Matisse GUI Builder</i> .	Alternativas comerciales
Sistema de Módulos	Estándar OSGi módulo del sistema o Netbeans específico	Estándar OSGi módulo

	de sistema de módulos.	del sistema.
Sistema de Construcción	Fuera de la caja, <i>Maven</i> o <i>Ant</i> .	Propiedad
Soporte de <i>JDK</i>	<i>VisualVM</i> , una aplicación de plataforma <i>NetBeans</i> , es en el <i>JDK</i> , JAR tantos de la Plataforma <i>NetBeans</i> se encuentran en el <i>JDK</i> también.	Sin un apoyo equivalente
Formación	Canciones de capacitación basados en la comunidad para organizaciones no comerciales.	Sin un apoyo equivalente

Fig. 1. Tabla comparativa entre los IDEs *Netbeans* y *Eclipse*.

A continuación se explican los criterios tomados en la tabla anterior:

- **Kit de herramientas de interfaz de usuario:** Cuando se utiliza la plataforma *NetBeans*, que va a programar con el oficial de juego de herramientas estándar de interfaz de usuario, que es *Swing*. *Eclipse* RCP, por otro lado, hace uso de SWT en su lugar.
- **Diseño de interfaz de usuario:** El galardonado *Matisse GUI Builder* es una parte estándar de *NetBeans* IDE. Se puede utilizar cuando el prototipo y el diseño de sus aplicaciones en la parte superior de la plataforma *NetBeans*. Sus arrastrar y soltar, las capacidades y funciones de apuntar y haga clic en Establecer un ambiente ideal para el diseño de la interfaz de usuario.
- **Sistema de módulos:** Tanto la plataforma *NetBeans* y el uso de *Eclipse* RCP marcas del sistema de módulos estándar de facto, que es OSGi. Además, la plataforma *NetBeans* proporciona su

propio sistema módulo, sobre la base del enfoque estándar de *Java* a la modularidad, contando con una extensión de la clase *ServiceLoader* *JDK 6* para la comunicación intermodal.

- **Sistema de Construcción:** Una característica específica de la plataforma *NetBeans* es el hecho de que su sistema de generación se basa en la hormiga. Ant es una herramienta estándar, de construcción no proveedor específico. Como resultado de ello, no están encerrados en el IDE *NetBeans*, pero puede, en cambio, la línea de comandos para construir sus aplicaciones de la plataforma *NetBeans*. De hecho, usted puede construir aplicaciones de la plataforma *NetBeans* fuera de la caja con la hormiga, y también con bastante facilidad con Maven.
- **Plataforma de NetBeans en el JDK:** Desde *JDK 6 Update 7*, muchos de los archivos *JAR* que componen la plataforma *NetBeans* son parte de la *JDK*. Los archivos *JAR* están allí porque *Java VisualVM*, la nueva herramienta *JDK*, se crea en la parte superior de la plataforma *NetBeans*. Cuando la construcción de su aplicación en la parte superior de la plataforma *NetBeans*, esté hecha puede ser útil tanto para el desarrollo, como para los propósitos de entrega.
- **Formación gratuita:** El equipo de *NetBeans* dispone de la certificación plataforma *NetBeans* para universidades, colegios y escuelas, así como instituciones no comerciales en general. Una versión personalizada de este curso está disponible para las empresas, a un precio estándar. *Eclipse RCP* no ofrece los entrenamientos libres de este tipo. [22]

Luego del análisis de estos IDEs se seleccionó la plataforma *Netbeans* para el desarrollo de la aplicación por las ventajas antes mencionadas.



El **NetBeans 7.1rc**, con licencia **GPL** será utilizada para facilitar el desarrollo con el lenguaje Java. Su *Kit* de Herramientas de Interfaz de Usuario es la *Swing* estándar.

Su Diseño de Interfaz de Usuario está hecho con premios gratis, a ganar y *Matisse GUI Builder*. Su Sistema de Módulos es el Estándar OSGi o *Netbeans* específico de sistema de módulos. Construye sus proyectos mediante *Maven* o *Ant* y Fuera de Caja. El Soporte de *JDK* es *VisualVM*, una aplicación para la plataforma *Netbeans*, *JAR* de la plataforma *Netbeans* se encuentran también en el *JDK*.

maven

Maven 2.2.23, con Licencia **Open Source**.

Maven es una herramienta para la gestión de proyectos de *software*, que se basa en el concepto de POM (*Project Object Model*). *Maven* es una herramienta de *software* para la gestión y construcción de proyectos *Java* creada por Jason van Zyl, de *Sonatype*, en 2002. Tiene un modelo de configuración de construcción simple, basado en un formato *XML*. Es decir, con *Maven* vamos a poder compilar, empaquetar, generar documentación, pasar los *test*, preparar las *builds*, entre otras funciones.



Ingeniería del **Software** asistida por computadoras

«**CASE**», el **Visual Paradigm v 6.4**, de Licencia Libre.

Herramienta *CASE* de diseño que utiliza *UML* como lenguaje de modelado. Muy útil para los desarrolladores de *software*

Lenguaje Unificado de Modelado (*UML*, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el *OMG* (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de *software*. *UML* ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de *software* reutilizables.



Gestor de Base de Datos PostgreSQL 9.0 con licencia GPL. Dicha herramienta será empleada para realizar las salvadas de todos los cambios que se realicen sobre la información que se gestiona en el producto a desarrollar.

Para el **control de versiones «CVS»** se utilizaron el *RapidSVN* del lado del cliente y el *Subversion* del lado del servidor.



RapidSVN v 0.12.0 con licencia **Open Source**. Se utiliza para gestionar los datos del repositorio. A cada integrante del proyecto se le asignarán diferentes permisos según los documentos y la información que sea necesaria para el rol que desempeñe.

El **Subversion v 1.6.12** de licencia **Open Source**. Es un sistema de control de versiones diseñado específicamente para reemplazar al

popular CVS. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como *svn* por ser el nombre de la herramienta utilizada en la línea de órdenes.

1.3.5. Otras herramientas



pgAdmin 3, Licencia *Open Source*.

pgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos *PostgreSQL*, siendo la más completa y popular para el desarrollo del sistema en cuestión. [20]

1.3.6. Metodología de Ingeniería de *Software* utilizada.

Metodología Ágil.

Las Metodologías Ágiles basan su fundamento en la adaptabilidad de los procesos de desarrollo, promueve la formalización de procesos adaptables y tienen como principal característica la habilidad de responder al cambio. Las cuales son sin duda uno de los temas recientes en ingeniería de *software* que están acaparando gran interés. Prueba de ello es que se están haciendo un espacio destacado en la mayoría de conferencias celebradas en los últimos años. Es tal su impacto que actualmente existen 4 conferencias internacionales de alto nivel y específicas sobre el tema. [23]

El ejemplo más destacado y exitoso de la metodología ágil, es la programación extrema o *XP* (por su nombre en inglés *Extreme*

Programming) como enfoque de la ingeniería de *software*.



Programación Extrema.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en una realimentación continua entre el cliente y el equipo de desarrollo, una comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. *XP* se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Entre los roles de *XP* tenemos:

Programador, Cliente, Entrenador, Rastreador y Probador. El ciclo de vida ideal de *XP* consta de cuatro a seis fases, las principales son: Planificación

(dentro de la que se encuentra Exploración), Desarrollo (donde se realizan tantas iteraciones como sean necesarias), Entrega y Mantenimiento y en algunos casos en que sea necesario se puede introducir una Fase Legal. [24]



Metodología SCRUM.

SCRUM es entonces la metodología ideal para toda la gestión de proyectos, sirve de soporte para acelerar el dinamismo que se identificó



en *XP*, la identificación de los pequeños sprint (iteraciones) y las reuniones con el grupo líder y tutores todos los días se acercaba más a la disciplina que se quería alcanzar en el grupo, donde líderes de solución y equipo de desarrollo se reunieran y controlaran los avances e identificaran los posibles riesgos que afectaban de una manera u otra la correcta ejecución del proyecto.

Metodología SXP

SXP es un híbrido cubano de metodologías ágiles que tiene como base las metodologías *SCRUM* y *XP* que permiten actualizar los procesos de desarrollo de *software* para el mejoramiento de su producción. Consta de 4 fases: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos de trabajo y actividades que generan artefactos.

Ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad.

Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados en una misma dirección, permitiendo además seguir de forma clara el avance de las tareas a realizar, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la producción, aumentando el nivel de interés del equipo.

La documentación discreta que brindan ambas metodologías se refuerza un poco, para garantizar que el conocimiento se comparta, incluso sirva de guía para los de menor formación profesional por encontrarse en grados iniciales e intermedios y para evitar que los que se encuentran en grados terminales una vez egresados de la UCI se lleven consigo el conocimiento y caiga en caos el futuro del proyecto.

Se hizo de vital importancia definir un documento de arquitectura donde se describiera la arquitectura del sistema, estrategias de integración con otros sistemas, así como la definición de herramientas y tecnologías de desarrollo a emplear por el equipo de solución. En gran medida los responsables de la definición arquitectónica de los sistemas estarán a cargo del profesor designado al proyecto.

1.3.7. Selección de la Metodología.

Se escoge la metodología SXP porque permite la comunicación y satisfacción del cliente como lo principal, define las necesidades y las pruebas de calidad para lograr esta meta. No hace mucho énfasis en la documentación pero exige un alto nivel de disciplina en las personas que participan en la construcción de la solución. Está diseñada para el trabajo de equipos pequeños y se apoya en las pruebas.

La idea es hacer las iteraciones lo más corta posibles y lograr una

retroalimentación frecuente durante las mismas, de este modo se conoce mejor porque parte se va.

En esta metodología el plazo de entrega del producto es corto, lo cual encaja perfectamente con el perfil de la presente tesis; sin significar esto que se limite al tiempo estimado, sino que la solución puede devenir cambios en el sistema luego de las pruebas de soporte que se hagan, flexibilizando a los futuros administradores del sistema realizar los cambios necesarios basados en las mismas de forma más segura y eficaz. [25]

1.3.7. Arquitectura de *Software*.

Estilo de Llamada y Retorno

Esta familia de estilos arquitectónicos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala. Miembros de la familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objeto y los sistemas jerárquicos en capas.

Este estilo arquitectónico permite que un arquitecto de *software* obtenga una estructura de programa que resulta relativamente fácil modificar y cambiar de tamaño. A continuación se muestra dentro de este estilo arquitectónico cuáles fueron las arquitecturas propuestas para la implementación del Sistema informativo del Gobierno de Artemisa. [20]

Arquitectura N-Capas



Figura 2. Arquitectura N-Capas.

Como propuesta de solución para el desarrollo se plantea el uso de la arquitectura N-Capas teniendo en cuenta que el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio, mecanismos de almacenamiento.

En este caso el servidor se centra en la capa lógica de negocio, la cual contiene los *plugins*, eventos y servicios del servidor en pos de responder las necesidades de los clientes.

Patrones GRASP

Los patrones *GRASP* describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen un apoyo para entender el diseño y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable. Los 5 patrones básicos se refieren a cuestiones y aspectos fundamentales del diseño, algunos de estos patrones utilizados en este trabajo son:

- **Creador:** se encarga de darle a la clase A la responsabilidad de crear objetos de la clase B. En este caso A es creador de los objetos B.
- **Bajo Acoplamiento:** garantiza que una clase no dependa de muchas otras ya que se hace difícil de reutilizar.
- **Alta Cohesión:** se encarga de asignar responsabilidades a las clases de modo que las funcionalidades estén estrechamente relacionadas y que cada elemento del diseño tenga una labor única. Una clase con baja cohesión se hace difícil de entender, de conservar y de reutilizar.

Patrones de Comportamiento

Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Entre los que se emplearon en el desarrollo del sistema se encuentran el Observador y el Mediador. [26]

1.4. Conclusiones del capítulo.

En este capítulo se hizo un estudio de los principales sistemas de gestión de la información existentes, partiendo de los principales conceptos asociados a la solución del sistema, luego se hizo una revisión de los principales *software* que realizan algunas de las funcionalidades que se desea que tenga la solución final.

Además del estudio de mercado realizado, también se evaluaron las herramientas, lenguajes y tecnologías que se utilizarán en el desarrollo, abordando las diferencias entre los posibles *IDEs* para la implementación y los lenguajes y protocolos, de acuerdo a las necesidades del cliente y a la metodología seleccionada.

Se hizo un análisis además, del uso de una u otra herramienta de modelado y control de versiones. En todos los casos se seleccionaron lenguajes de programación y herramientas libres de gran aceptación en el mercado mundial, las mismas, por su viabilidad y rapidez, facilitan el desarrollo y utilización de la aplicación *web* resultante.

Capítulo 2: Características del Sistema

En este capítulo se presentan los artefactos resultantes de la aplicación de la metodología determinada en el capítulo anterior (SXP) durante sus 3 primeras fases: exploración, planificación y desarrollo. Se describirán los procesos que se llevan a cabo en el negocio, así como sus actores y trabajadores, y se demostrarán mediante diagramas los casos de uso del negocio y del sistema. Además, se expondrán los requisitos funcionales del mismo.

2.1. Descripción general de los procesos involucrados en el negocio.

Con el objetivo de lograr una mejor organización de distribución político administrativa en Cuba y debido a la gran cantidad de municipios que contenía la provincia de La Habana se hizo necesario reestructurar los mismos. De esta manera surgen las provincias de Mayabeque y Artemisa. En esta última, en el municipio cabecera se encuentra ubicado el órgano de la Administración Provincial de Artemisa (APA), el cual está dividido en 32 direcciones y una de ellas es la dirección de Trabajo CITMA.

Esta dirección maneja un conjunto de información referente a todas las gestiones que se realizan a nivel provincial respecto al medio ambiente, los eventos de innovación e investigación que realizan las entidades de la región, ya sean pertenecientes a las ramas de la Educación, la Agricultura, el control de los puertos y bahías ubicados dentro de la provincia y las penalidades y legislaciones del CITMA que corrigen y controlan el funcionamiento de las mismas.

La información es entregada personal e impersonalmente, viniendo en ocasiones vía email, telefónica, por mensajeros, en formato duro o en dispositivos de almacenamiento. Luego, es gestionada de forma manual en modelos de Excel y Access, los cuales son accedidos y modificados por varios especialistas lo cual no garantiza su confidencialidad e integridad de los datos, por lo cual se ve sujeta a errores humanos que provocan errores en la información, duplica de datos, e incluso, pérdida de los mismos. En la medida en que la nueva administración se organice pueden surgir nuevos cambios, lo cual no promete que el sistema de recepción de la información siga siendo funcional en unos años, por lo que no hay confiabilidad de los datos.

Posteriormente la información es analizada por una comisión y utilizada en la Toma de Decisiones respecto a la contaminación de las reservas de biosfera, el vertimiento de residuos en áreas aledañas a ríos y mares, las inundaciones costeras durante la temporada ciclónica, la degradación de los suelos, así como el seguimiento y organización del Fórum de Ciencia y Técnica.

De este modo se ve afectada la organización y centralización de la información lo cual retrasa la entrega de la misma y su posterior utilización en la toma de decisiones por parte del presidente y de la Dirección del CITMA de la AP de Artemisa.

2.2. Objeto de Automatización

A petición del cliente se desea automatizar la gestión de la información entregada por las dependencias de los organismos que se controlan en la Dirección del CITMA de la AP de Artemisa, permitiendo a los usuarios del sistema el ingreso y modificación de información y modificación,

búsqueda y eliminación de reportes.

2.3. Propuesta del sistema

La solución que se presenta es un sistema basado en una plataforma *web*, hecho con lenguajes de programación *web* (*HTML5*, *JavaScript*, *CSS3*) y orientado a objetos (*Java*) que permite el trabajo con una base de datos (*PostgreSQL 8.4+* y *PgAdmin 3*), haciendo uso de dos *frameworks* de desarrollo (*jQuery* y *JWebSocket*) .

Tendrá todos los modelos y módulos manejados en la Dirección del CITMA de la AP de Artemisa, incluyendo además, los nuevos modelos del Fórum de Ciencia y Técnica, no manejados de forma digital y estandarizada hasta el momento.

2.2.1. Planificación del proyecto por Roles

Rol	Responsabilidad	Nombre y Apellidos
Gerente	Toma las decisiones finales. Tutor.	Ing. Dania Fernández Aguilar
Líder del Proyecto	Dirige y controla las tareas del equipo. Tutor.	Ing. Raisa Ortega Báez
Cliente	Participa en las tareas que involucran la Lista de Reserva del Producto.	Consejo de la Administración Provincial de Artemisa.
Jefe de Proyecto	Controla las tareas del equipo y la toma las decisiones. Tutor.	Ing. Raisa Ortega Báez

Programadores	Elabora el código de las nuevas funcionalidades a implementar. Escribe las pruebas unitarias. Estudiantes.	Gabriela García Rivero Mauris Quilez Reyes
Analistas	Escribe las pruebas de usuario y las pruebas funcionales para validar su implementación. Estudiantes.	Gabriela García Rivero Mauris Quilez Reyes
Diseñadores	Encargados de diseño del sistema; así como los prototipos de interfaces máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción. Estudiantes.	Gabriela García Rivero Mauris Quilez Reyes
Encargado de Pruebas	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente. Estudiantes.	Gabriela García Rivero Mauris Quilez Reyes
Arquitecto	Su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas. Estudiantes.	Gabriela García Rivero Mauris Quilez Reyes

Tabla 2.1. Roles encargados de ejecutar la propuesta de SGI.

2.2.2. Modelo de Dominio

El Modelo de Dominio es aquel que proporciona un entendimiento entre

los clientes y los desarrolladores del sistema. En caso de que el Modelo de Historias de Usuario del Negocio esté bien definido, no sería necesario.

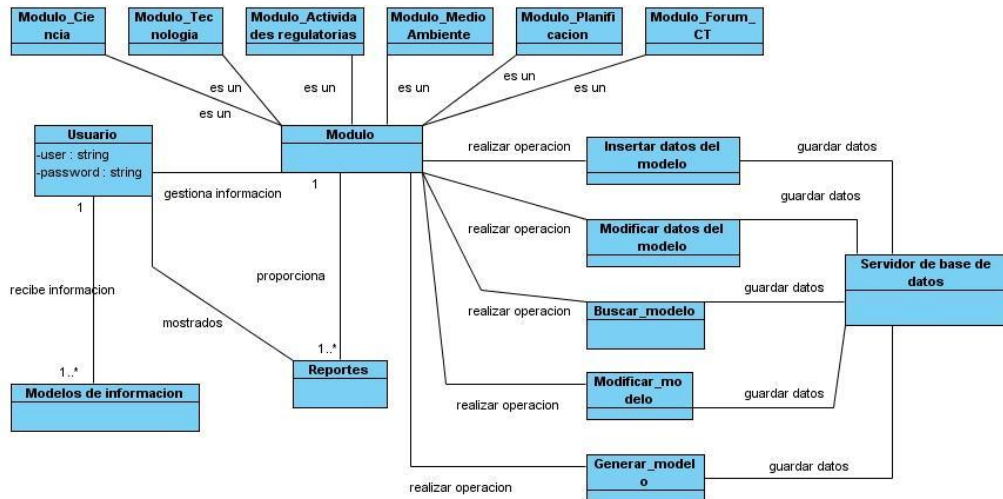


Diagrama 1. Modelo de Dominio de la propuesta de SGI.

Descripción de las partes:

1. **Modelo de Información:** Hace referencia al producto final que es enviado a la DOPI cada período indicado.
2. **Usuario:** Es la persona que trabaja directamente con el sistema.
3. **Reporte:** Es una parte del módulo de información.
4. **Módulo:** Es el producto final de la recopilación de información.
 - **Módulo Ciencia:** Es el documento final donde se resumen las informaciones de las obras científicas, los programas y proyectos en ejecución.
 - **Módulo Tecnología:** Es el documento final donde se concentra la información del cumplimiento de la nomenclatura provincia y nacional de impacto.

- **Módulo de Medio Ambiente:** Es el documento donde se recopila la información de las áreas protegidas, reducción de cargas contaminantes y empresas inscritas.
- **Módulo de Planificación:** Es el documento final donde se resumen las informaciones de las listas de proyectos que se administran, la ejecución financiera del plan de ciencia y técnica, las salidas finales de la ejecución del plan de ciencia y técnica, entre otros.
- **Módulo Actividades Regulatorias:** Es el documento final donde se concentra la información de las solicitudes de licencias ambientales, la inspección de licencias ambientales otorgadas, entre otros.
- **Módulo Fórum de Ciencia y Técnica:** Es el documento donde se recopila la información del evento del FCT de la provincia.

5. **Insertar, modificar y eliminar datos, buscar y generar modelos:** Permite gestionar la información.

6. **Servidor BD:** Base de datos que contiene todas las entidades de para el trabajo con la información.

2.2.3. Lista de Reserva del Producto

En la Lista de Reserva del Producto se identifican los Requisitos Funcionales y No Funcionales de la solución propuesta, además, se declaran los responsables de su estimación y la cantidad de días que se le va a dedicar a cada uno de estos requerimientos.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Requerimientos Funcionales				
Muy Alta				
	1	Insertar información sobre Programas y Proyectos en Ejecución.	1 día	Analista
	2	Modificar información sobre Programas y Proyectos en Ejecución.	1 día	Analista
Alta				
	3	Generar reporte sobre Programas y Proyectos en Ejecución.	1 día	Analista
	4	Buscar reporte sobre Programas y Proyectos en Ejecución por Proyecto.	1 día	Analista
	5	Eliminar reporte sobre Programas y Proyectos en Ejecución.	1 día	Analista
Media				
Baja				
Requerimientos No Funcionales				
	6	El servidor debe poseer al menos 1 GB RAM o superior.		
	7	El servidor debe poseer un Microprocesador <i>Pentium</i> o superior.		
	8	Instalar Máquina Virtual de <i>Java</i> (jdk-6u3).		
	9	Instalar el Entorno de Desarrollo Integrado (<i>Netbeans</i> 7.0.1)		
	10	Instalar el servidor <i>Maven</i> 2.0 o superior.		
	11	La comunicación entre la base de datos de integración y el servidor-cliente es a través del protocolo <i>JWebSocket</i> .		
	12	El sistema necesita estar conectado		

		directamente a un dispositivo de red.		
	13	Utilizar el lenguaje de programación definido durante la investigación.(JAVA)		
	14	Lograr que los elementos definidos en las clases tengan una estructura homogénea.		
	15	Las herramientas a utilizar se encuentran bajo licencia GPL.		
	16	Garantizar la integridad de la información.		
	17	El tiempo de respuesta debe ser en tiempo real.		

Tabla 2.2. Lista de Reserva del Producto. Ver Anexo 2.

2.2.4. Historia de Usuario

Las Historias de Usuario brindan una pequeña descripción de que es lo que se debe brindar en cada requisito funcional que se implemente, así como mostrar una vista no funcional de cómo podría quedar este requerimiento, entre otras características.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar Reportes de Programas y Proyectos en Ejecución.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Gabriela García Rivero y Mauris Quílez Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: La presente historia de usuario tiene como objetivo añadir y modificar toda la información referente a los Programas y	

Proyectos en Ejecución de la dirección. Además permite eliminar los Reportes de Programas y Proyectos en Ejecución.

Observaciones: ninguna

Prototipo de interfaz:

The screenshot shows a window titled "Modulo_Ciencia_Programas_y_Proyectos_en_Ejecucion_Insertar". The interface is organized into several sections:

- Top Section:** Contains two input fields for "IdModulo" and "Fecha".
- Datos del Informante:** A sub-section containing:
 - Input fields for "Nombre y Apellidos", "Dependencia", and "Teléfono".
 - A text area for "Dirección".
 - Input fields for "Email" and "Período que Informa".
- Datos de los Proyectos:** A sub-section containing:
 - Input fields for "Nombre del Programa" and "Tipo de Programa".
 - Input fields for "Título del Proyecto" and "Institución Ejecución Principal".
 - Dropdown menus for "Seleccione el Estado" and "Seleccione el OACE".
 - A radio button labeled "Plan Turquino-Manatí".
 - An "Insertar Proyecto" button.
- Bottom Section:** Contains two buttons: "Insertar" and "Cerrar".

Modulo_Ciencia_Programas_y_Proyectos_en_Ejecucion_Modificar

IdModulo Fecha

Datos del Informante

Nombre y Apellidos Dependencia Teléfono

Dirección

Email Período que Informa

Nombre del Programa

Tipo de Programa

Datos de los Proyectos

Título del Proyecto Institución Ejecución Principal

Seleccione el Estado Seleccione el OACE Plan Turquino-Manatí

Insertar Proyecto

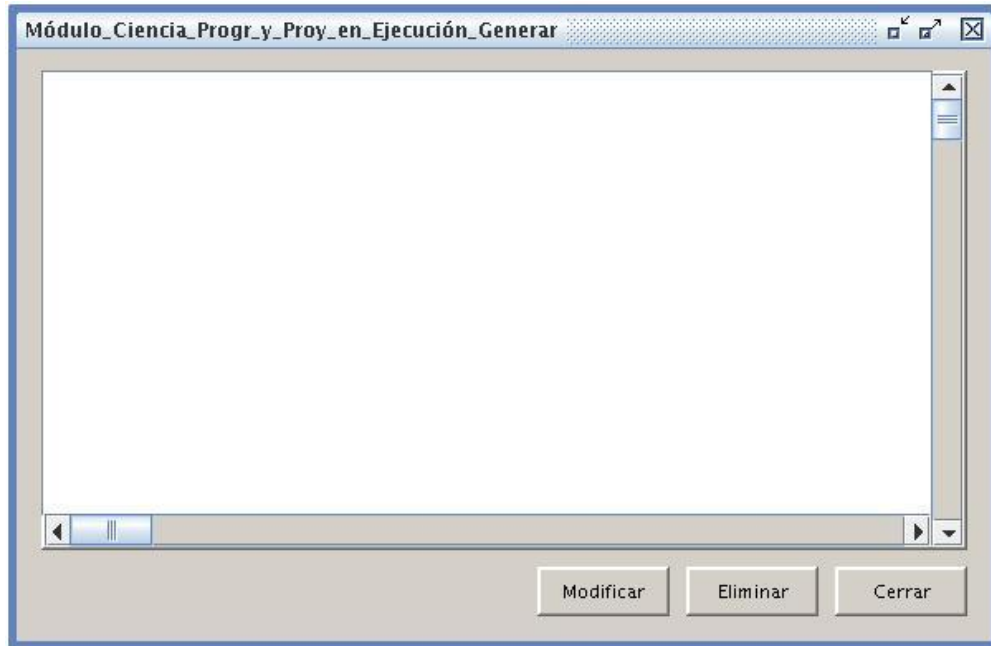
Modificar Cerrar

Historia de Usuario

Número: HU_2	Nombre Historia de Usuario: Generar Reporte de Programas y Proyectos en Ejecución.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Gabriela García Rivero y Mauris Quílez Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: La presente historia de usuario tiene como objetivo generar un reporte con toda la información referente a los Programas y Proyectos en Ejecución.	

Observaciones: Para que esto sea posible se debe realizar anteriormente la inserción de los datos de al menos un Programa y Proyecto en Ejecución.

Prototipo de interfaz:



Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Buscar reportes de Programas y Proyectos en Ejecución.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Gabriela García Rivero y Mauris Quílez Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: La presente historia de usuario tiene como objetivo la	

búsqueda de reportes de Programas y Proyectos en Ejecución.

Observaciones: Para que esto sea posible se debe realizar anteriormente la creación de reportes de Programas y Proyectos en Ejecución.

Prototipo de interfaz:

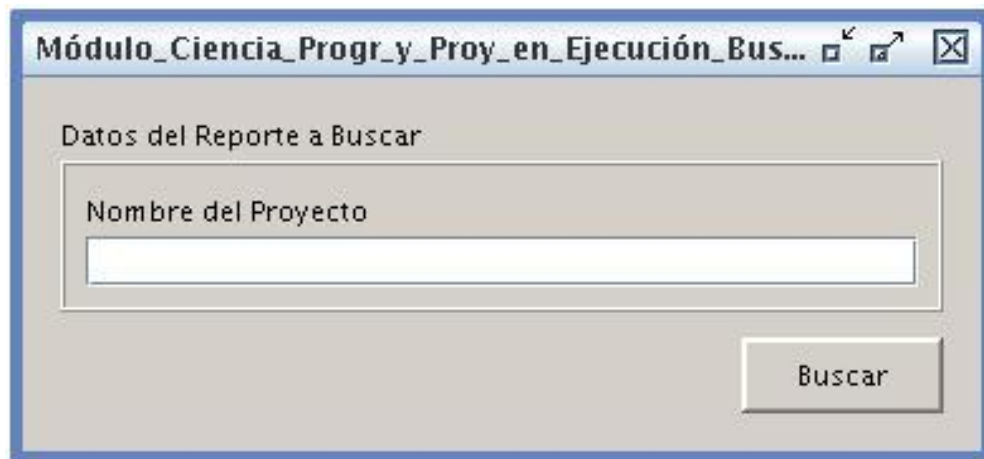


Tabla 2.3. Historia de Usuario. Reporte de Programas y Proyectos en Ejecución. Ver Anexo 3.

2.2.5. Tareas de Ingeniería

Las Tareas de Ingeniería son aquellas que desglosan las operaciones que se deben realizar en cada una de las Historias de Usuario. Pueden ser de Desarrollo, Corrección, Mejora, Otra (Especificar), por lo que una misma Historia de Usuario puede tener más de una Tarea de Ingeniería.

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar la funcionalidad Añadir información de los	

Programas y Proyectos en Ejecución.

Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 30/01/2012	Fecha Fin: 30/01/2012

Programador Responsable:: Gabriela García, Mauris Quilez

Descripción: En la presente tarea se va a implementar la funcionalidad de Añadir toda la información referente a los Programas y Proyectos en Ejecución de la dirección con los datos insertados por el usuario.

Tarea de Ingeniería

Número Tarea: 1.2 **Número Historia de Usuario:** HU_1

Nombre Tarea: Implementar la funcionalidad Modificar información de los Programas y Proyectos en Ejecución.

Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 31/01/2012	Fecha Fin: 31/01/2012

Programador Responsable:: Gabriela García, Mauris Quilez

Descripción: En la presente tarea se va a implementar la funcionalidad de Modificar toda la información referente a los Programas y Proyectos en Ejecución de la dirección que dependerá de los datos que el usuario quiera modificar.

Tarea de Ingeniería

Número Tarea: 2.1 **Número Historia de Usuario:** HU_2

Nombre Tarea: Implementar la funcionalidad Generar Reportes de los Programas y Proyectos en Ejecución.

Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
-----------------------------------	--------------------------------

Fecha Inicio: 01/02/2012	Fecha Fin: 01/02/2012
Programador Responsable:: Gabriela García, Mauris Quilez	
Descripción: En la presente tarea se va a implementar la funcionalidad de Generar los reportes con toda la información referente a los Programas y Proyectos en Ejecución donde el usuario podrá modificar o eliminar su selección.	
Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: HU_3
Nombre Tarea: Implementar la funcionalidad Buscar Reportes de los Programas y Proyectos en Ejecución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 01/02/2012	Fecha Fin: 01/02/2012
Programador Responsable:: Gabriela García, Mauris Quilez	
Descripción: En la presente tarea se va a implementar la funcionalidad de Buscar reportes de Programas y Proyectos en Ejecución teniendo en cuenta una especificación del usuario.	

Tabla 2.4. Tareas de Ingeniería. Reporte de Programas y Proyectos en Ejecución. Ver Anexo 4.

2.2.6. Plan de Releases

El Plan de *Releases* es una estimación de la cantidad total de semanas que debe tomar la implementación del proyecto. Propone un resumen por iteración de las Historias de Usuario que serán desarrolladas.

Releases	Descripción	Orden de la HU a implementar	Duración
----------	-------------	------------------------------	----------

	de la Iteración		Total
1	Gestionar la información de los reportes	HU_1, HU_3, HU_4, HU_6, HU_7, HU_9, HU_10, HU_12, HU_13, HU_15, HU_16, HU_18, HU_19, HU_21, HU_22, HU_24, HU_25, HU_27, HU_28, HU_30, HU_31, HU_33, HU_34, HU_36, HU_37, HU_39, HU_40, HU_42, HU_43, HU_45, HU_46, HU_48, HU_49, HU_50, HU_51, HU_53, HU_54, HU_56, HU_57, HU_59, HU_60, HU_62, HU_63, HU_64, HU_66, HU_67, HU_69, HU_70, HU_72, HU_73, HU_75, HU_76, HU_78, HU_79, HU_81	9 semanas
2	Generar reportes	HU_2, HU_5, HU_8, HU_11, HU_14, HU_17, HU_20, HU_23, HU_26, HU_29, HU_32, HU_35, HU_38, HU_41, HU_44, HU_47, HU_50, HU_53, HU_55, HU_58, HU_61, HU_65, HU_68, HU_71, HU_74, HU_77, HU_80,	4 semanas

Tabla 2.5. Plan de Releases.

2.2.7. Diagrama de Paquetes

El Diagrama de Paquetes especifica cuáles van a ser las vistas y los *plugins* que van a ser creados en el lado del servidor y el cliente, así como las clases del modelo de las cuales van a pedir los datos y mostrar al usuario la información solicitada.

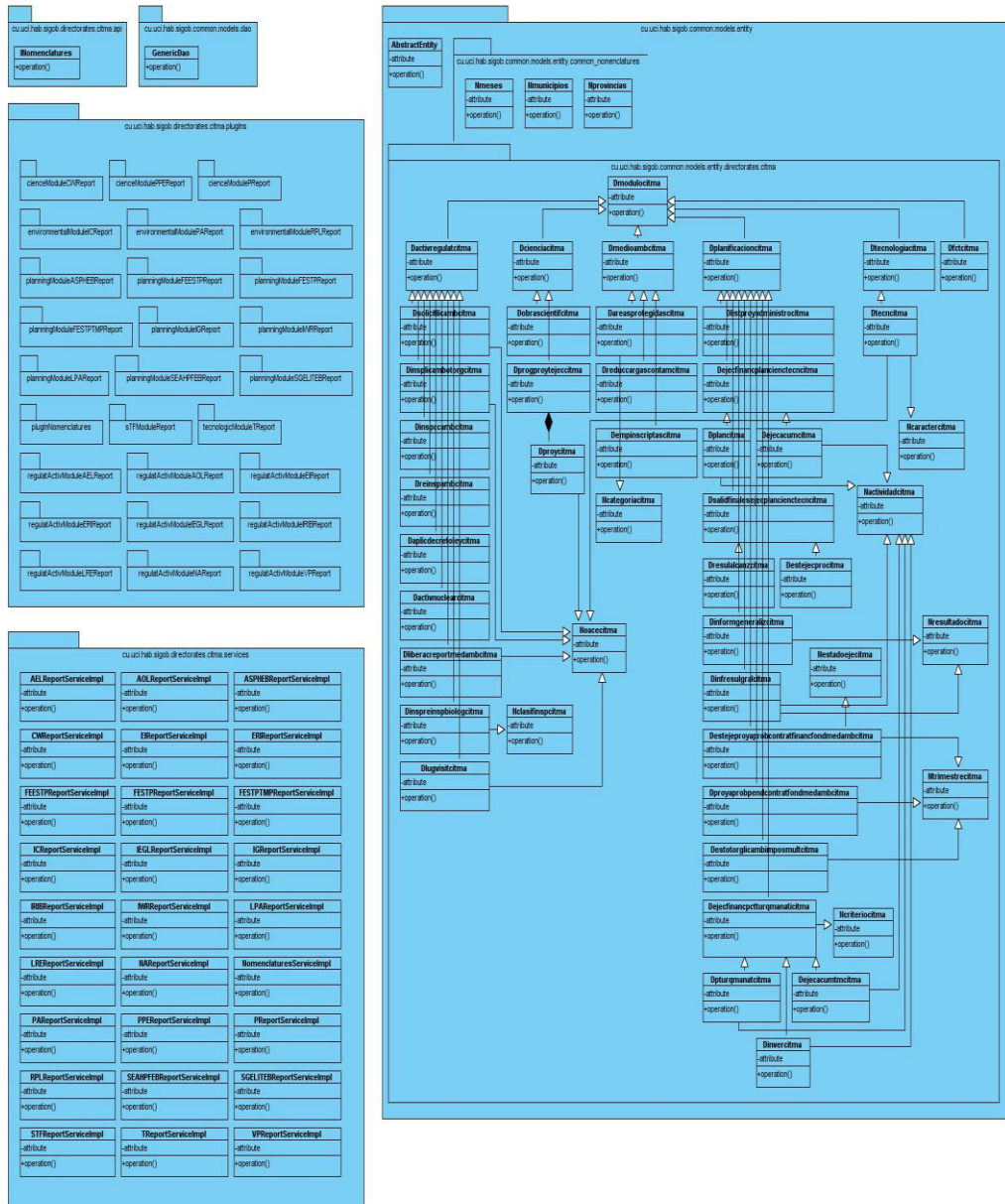


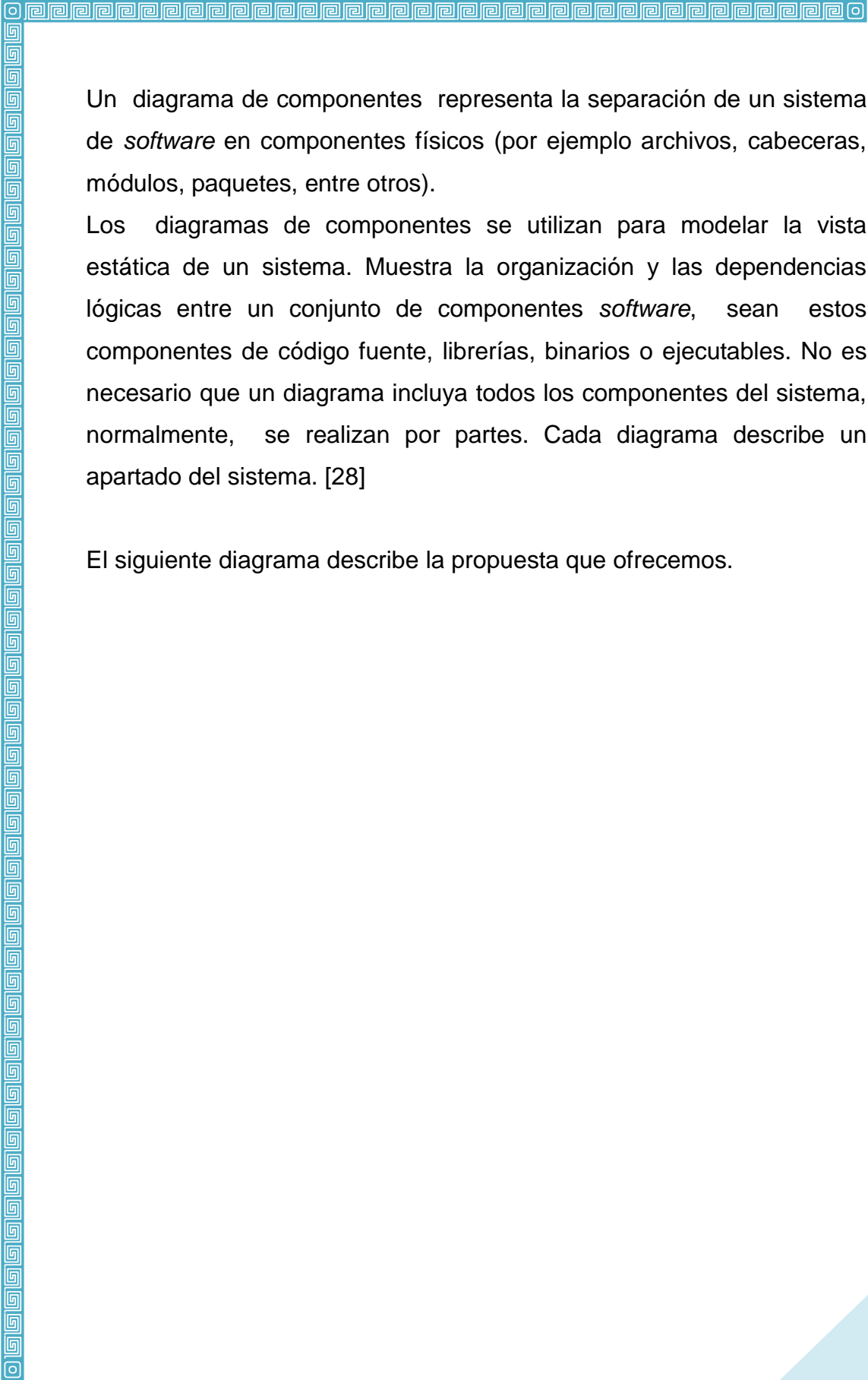
Diagrama 2. Diagrama de Paquetes de la propuesta de SGI.

A continuación se reflejan las descripciones de los paquetes:

Paquetes	Descripción
cu.uci.hab.sigob.common.models.dao	Se encuentra el dao genérico para poder realizar las consultas a la base de datos.
cu.uci.hab.sigob.common.models.entity	Se encuentra el <i>AbstractEntity</i> del cual heredan el resto de las entidades de la base de datos
cu.uci.hab.sigob.common.models.entity.common_nomenclatures	Se encuentran todas las entidades comunes para el sistema
cu.uci.hab.sigob.common.models.entity.directorates.citma	Se encuentran las entidades de la dirección del CITMA.
cu.uci.hab.sigob.directorates.citma.plugins	Se encuentran todos los <i>plugins</i> y eventos del módulo CITMA.
cu.uci.hab.sigob.directorates.citma.servicios	Se encuentra las clases servicios que sirven como intermediarios entre los <i>plugins</i> y la base de datos.
cu.uci.hab.sigob.directorates.citma.api	Se encuentran las clases interfaces de los nomencladores.

2.2.8. Diagrama de Componentes

Los Diagramas de Componentes se pueden identificar porque describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de *software* que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros. [27]



Un diagrama de componentes representa la separación de un sistema de *software* en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, entre otros).

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes *software*, sean estos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente, se realizan por partes. Cada diagrama describe un apartado del sistema. [28]

El siguiente diagrama describe la propuesta que ofrecemos.

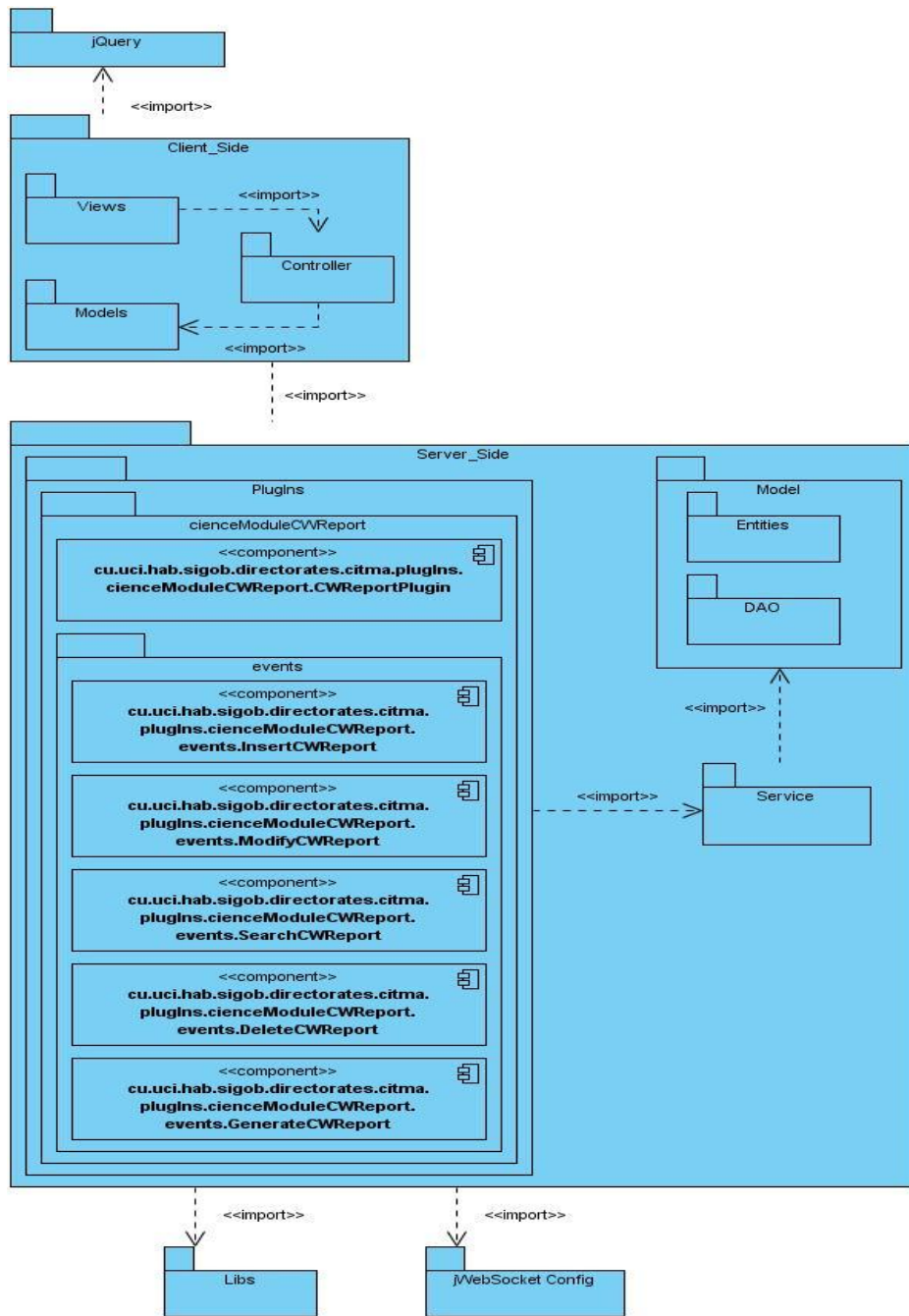


Diagrama 3. Diagrama de Componentes de la propuesta de SGI.

2.4. Conclusiones del capítulo

Al terminar el capítulo quedaron determinados los requisitos funcionales y no funcionales del sistema y las historias de usuarios que los describirían, de este modo el sistema se basa en una estructura funcional que responde a las necesidades del cliente a la hora de satisfacer la información.

Se definieron además, los paquetes a utilizar en la solución, especificando las vistas, las clases controladoras y las clases del modelo que intervendrían en la gestión de los datos. También se estimó la duración del proyecto, obteniéndose el documento Plan de Releases de la propuesta basado en las descripciones e iteraciones del documento Historia de Usuarios.

Capítulo 3: Adquisición y resultados del Sistema.

En el presente capítulo se exponen los resultados de los casos de pruebas realizadas al sistema durante cada una de las iteraciones, las mismas fueron necesarias para progresar en la calidad del producto final.

3.1. Pruebas Unitarias

Las pruebas unitarias son una forma de probar el correcto funcionamiento del código; estas sirven para asegurar que cada uno de los módulos funcione correctamente por separado. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Estas pruebas aisladas proporcionan cuatro ventajas básicas:

- **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se llama refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- **Simplifica la integración:** Puesto que permiten llegar a la integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- **Separación de la interfaz y la implementación:** Dado que la única interacción entre los casos de prueba y las unidades bajo prueba son las interfaces de estas últimas, se puede cambiar cualquiera de los dos sin afectar al otro.

- **Los errores están más acotados y son más fáciles de localizar:** dado que tenemos pruebas unitarias que pueden desenmascararlos.

Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto. Además, puede no ser trivial anticipar todos los casos especiales de entradas que puede recibir en realidad la unidad de programa bajo estudio. Las pruebas unitarias sólo son efectivas si se usan en conjunto con otras pruebas de software. [29]

3.1.1. Pruebas de Caja Blanca

Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

En la prueba de caja blanca se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, comprobando los bucles y condiciones, y examinado el estado del programa en varios puntos.

A primera vista, la prueba de caja blanca profunda nos llevaría a tener "programas 100 por cien correctos", es decir:

- Definir todos los caminos lógicos
- Desarrollar casos de prueba para todos los caminos lógicos
- Evaluar los resultados

Prueba del Camino Básico

El método del camino básico (propuesto por McCabe) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Notación del grafo de flujo o grafo del programa

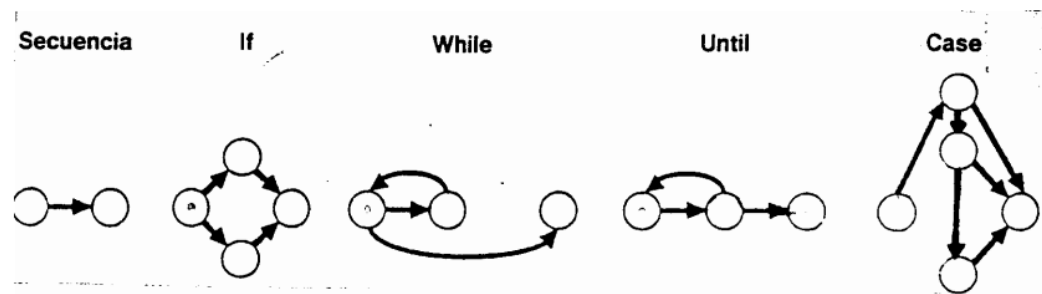


Figura 3. Ejemplos de Flujo de Control Lógico.

A continuación se muestra un ejemplo basado en un diagrama de flujo que representa la estructura de control del programa.

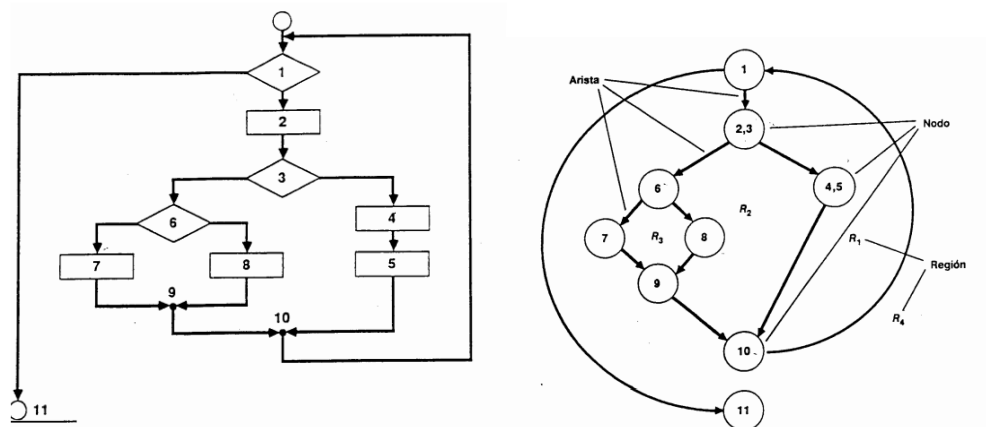


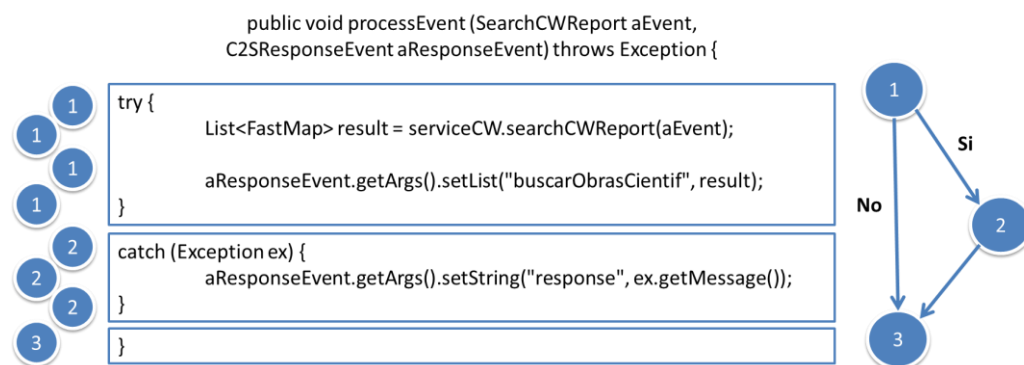
Figura 4. Ejemplo de Diagramas de Flujo.

En el grafo de flujo:

- Cada nodo representa una o más sentencias procedimentales.
- Un solo nodo puede corresponder a una secuencia de pasos del proceso y a una decisión.
- Las flechas (aristas) representan el flujo de control. [30]

Complejidad Ciclomática

Un código es eficiente si su complejidad Ciclomática es mayor o igual al número de caminos básicos.



Complejidad Ciclomática = número de aristas - número de nodos +2

$$CC = 3 - 3 + 2 = 2$$

Caminos Básicos

C1: 1 – 3, C2: 1 – 2 – 3

Caso de Prueba 1 para C1

Secuencia de Pasos	Respuesta del método
try {	Lista(buscarObrasCientif)
List<FastMap>result=serviceCW.searchCWReport(

<code>aEvent);</code>	
<code>aResponseEvent.getArgs().setList("buscarObrasCientif", result); }</code>	
Resultado	Se siguió un flujo correcto y el método devolvió una lista con los objetos encontrados.

Caso de Prueba 2 para C2	
Secuencia de Pasos	Respuesta del método
<code>try {</code> <code>List<FastMap>result=serviceCW.searchCWReport(aEvent);</code>	Error
<code>aResponseEvent.getArgs().setList("buscarObrasCientif", result); }</code>	
<code>catch (Exception ex) {</code> <code>aResponseEvent.getArgs().setString("response", ex.getMessage());</code> <code>}</code>	Hubo un fallo en el flujo de pasos y el método envía una excepción al cliente.
Resultado	El método devolvió una excepción con el mensaje que devuelve el servicio.

3.2. Resultados obtenidos

Se obtuvo como resultado el Sistema Informativo de la Administración Provincial de Artemisa (SIGOB), específicamente el servidor, versión 1.0, para el módulo que controla y administra las funcionalidades e información destinadas a la Dirección del CITMA.

3.3. Funcionalidades obtenidas

Entre las funcionalidades concentradas en la versión 1.0 del módulo para la Dirección del CITMA se encuentran:


- Insertar, Modificar y Buscar información de los reportes que se manejan en la dirección antes mencionada.
- Eliminar y Generar Reportes.
- Además de futuras funcionalidades que se implementarán en el marco del soporte que se le dará al sistema.

3.4. Aporte social y económico

El sistema está concebido para brindar rapidez y manejabilidad de forma eficaz y organizada, permitiendo el acceso de forma restringida de acuerdo al usuario y al tipo de acción a realizar, lo que garantiza la integridad de los datos. La información podrá ser accedida en cualquier momento y desde cualquier puesto de trabajo a través de un navegador proporcionando centralidad. Además al ser desarrollado con herramientas libres no genera pérdidas al país e incluye su entrega gratis a la empresa en la cual será instalado.

3.1.5. Conclusiones del capítulo

Al concluir este período de pruebas se puede concluir que el sistema está apto para las funcionalidades para las que fue concebido, garantizando de este modo que sean capturados todos los errores y en otro caso no exista ninguno, comprobando que los requerimientos y la estructura de las clases determinados en las etapas iniciales han sido los más óptimos posibles para el desarrollo del sistema resultante.



Además de asegurar que el sistema cumpla con los requerimientos del cliente a modo de brindar mayor seguridad en la captura y el almacenamiento de los datos a procesar.

Conclusiones Generales

La actual investigación sentó las bases para que la solución propuesta fuera desarrollada de la mejor forma posible, además de brindar los conocimientos básicos y avanzados para la realización de dicho trabajo a sus desarrolladores. Demostrando que con el uso de nuevas tecnologías se pueden desarrollar en poco tiempo sistemas capaces de brindar centralidad, seguridad y eficacia en el tratamiento de la información.

Con el estudio de este trabajo investigativo:

- Se crearon las bases teóricas como fundamento de la presente tesis y la solución que propone.
- Se realizó un estudio del estado actual de los sistemas de gestión de la información y los procesos implicados en el manejo de la información.
- Se escogieron las metodologías, herramientas y tecnologías a utilizar para el desarrollo de la aplicación.
- Se desarrollaron los requerimientos en un módulo para el servidor de la aplicación propuesta para la dirección del CITMA de la Administración Provincial de Artemisa.

Recomendaciones

Luego de terminada la investigación se recomienda que se desarrollen las funcionalidades que modifican las estructuras de los reportes para que de este modo el sistema siga siendo funcional en caso de cambios estructurales en la Dirección CITMA de la Administración Provincial de Artemisa.

Así como la de Exportar a PDF que permitirá la salida de los datos del sistema para una mejor portabilidad y uso en la toma de decisiones.

Referencias

Referencias Bibliográficas

- [1b] Del libro: «Introducción a la Teoría General de la Administración», Séptima Edición, de Chiavenato Idalberto, McGraw-Hill Interamericana, 2006, Pág. 110.
- [1c] Del libro: «Introducción a los Negocios en un Mundo Cambiante», Cuarta Edición, de Ferrell O. C. y Hirt Geoffrey, McGraw-Hill Interamericana, 2004, Pág. 121.
- [1d] Del Tabloide “Introducción al conocimiento del Medio Ambiente”. Curso de Universidad para Todos.
- [] Eleanne Magnolia Guilbeaux Hernández, Reinier Villavicencio de la Rosa. “Sistema automatizado para la gestión de la información generada en el proceso de evaluación a los proyectos de Software de la Universidad de las Ciencias Informáticas (SGIGPE).” Universidad de las Ciencias Informáticas. Ciudad de la Habana, Julio de 2009. 127.
- [5c] Jorge Antonio Díaz Gutiérrez. "Desarrollo de un IDE libre y multiplataforma para la creación de componentes visuales de ActionScript para *Software* Educativo: codeDraw". Universidad de las Ciencias Informáticas. Ciudad de la Habana, Junio de 2009. 83.
- [13] Letelier Torres, Patricio y Sánchez López, Emilio A. Metodologías Ágiles en el Desarrollo de *Software*. Alicante: ISSI, 2003.
- [15] Peñalver, G., Meneses, A., García, S. SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE *SOFTWARE*, Mayo 2010, Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.
- [18] Documento Arquitectura_de_*Software*_SXP-SIGOB.pdf
- [20] Osdel Hernández Caballero. "Portal *Web* para la gestión de la

información del deporte de lucha en Cuba.". Universidad de las Ciencias Informáticas, Junio 2011. 92.

Bibliografía

Andrés Bryan Bello Blanco. "Sistema para la gestión de la información del área de Mercadotecnia, del Centro de Tecnologías de Gestión de Datos.". Universidad de las Ciencias Informáticas, Junio 2011. 80.

Elda Montoya Rodríguez. "Sistema de Gestión de Información de los Recursos Humanos (SGIRH) de la Facultad 2.". Universidad de las Ciencias Informáticas. Ciudad de la Habana, Junio de 2009. 131.

Helen Caridad García González, Mario Enrique Olivares Ferreira. "Sistema para la Gestión de Información de los Proyectos Productivos en la Facultad 8". Universidad de las Ciencias Informáticas. Ciudad de la Habana, Junio de 2008. 116.

Yenisleidis Rodríguez Martínez. "Diseño e implementación de una herramienta webmétrica libre para el análisis personalizado de sitios webs para GEWEB.". Facultad 1, Universidad de las Ciencias Informáticas. Ciudad de la Habana, Junio 2011. 167.

Yenisleidis Rodríguez Martínez. "Diseño e implementación de una herramienta webmétrica libre para el análisis personalizado de sitios webs para GEWEB.". Facultad 1, Universidad de las Ciencias Informáticas. Ciudad de la Habana, Junio 2011. 167.

Aristegui, O. J. L. 2010. "Los casos de prueba en la prueba de *software*. Revista Digital Lámpsakos", No. 3, pp. 27B34.

Referencias Webgráficas

[1a] Rafael Campillo Lorenzo. "El método científico", 2006-2007.

Disponible en: <http://www.scribd.com/doc/58663071/33/Metodo-de-la-modelacion>

[2]<http://seguridad.cai.es/paginas/paginafinal.asp?idNodo=224>,<http://sgsi-iso27001.blogspot.com/>

[3]<http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>

[4] <http://definicion.de/sitio-web/>

[5a] http://www.veasuiip.com/conceptos/concepto_servidor.html

[5b] <http://es.wikipedia.org/wiki/Servidor>

[5d] <http://www.gnu.org/philosophy/free-sw.es.html>

[5e] <http://www.definicionabc.com/tecnologia/software-propietario.php>

[6] <http://netbeans.org/features/platform/compare.html>

[7] <http://blog.develoft.com/?p=18>

[8]<http://www.compute-rs.com/es/consejos-636152.htm>,
<http://eddi.ith.mx/Curso/java.htm>,<http://es.e-articles.info/t/i/5006/>,
<http://definicion.de/java/>

[9] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>

[10] <http://www.desarrolloweb.com/articulos/que-es-html5.html>

[11]<http://juanegas.blogspot.com/2011/11/novedades-de-css3.html>,
<http://blog.consultec.es/index.php/2012/01/8-aspectos-basicos-de-css-3-que-todo-disenador-debe-conocer/>,

<http://www.marcelopedra.com.ar/blog/2009/08/30/6-nuevas-funciones-de-css3-que-ya-pueden-implementarse/>

[12]<http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-de-informacion-en-estudio-de-medio-ambiente.htm>

[14] PENADÉS, M CARMEN; LETELIER, PATRICIO. Metodologías

ágiles para el desarrollo de *software*: eXtreme Programming (XP).

Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>.

[17] FINEANS. [Online] 2011.

[http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-](http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf)

[15DBYBZ-FH/UML-diagramaComponentes.pdf](http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf).

[19] FINEANS. [http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-](http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf)

[15DBYBZ-FH/UML-diagramaComponentes.pdf](http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf).

[21] indalog.ual.es/mtorres/LP/Prueba.pdf