



Facultad Regional Mártires de Artemisa

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

***TÍTULO: Servidor para el módulo de la Dirección de Auditoría de la
Administración Provincial de Artemisa.***

AUTOR: Williams González Montesino.

TUTOR: Ing. Leosmel Zayas Castillo.

CO-TUTOR: Lic. Roberto Cruz Acosta.

Artemisa, Junio 2012

“Año 54 de la Revolución”

Declaración de Autoría

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Aplicaciones e Integración 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 _____.

Williams González Montesino

Ing. Leosmel Zayas Castillo

Firma del Autor

Firma del Tutor

Lic. Roberto Cruz Acosta

Firma del Co-Tutor

Frase

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica

Aristóteles

Agradecimientos

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

A mi papá, a quien respeto por su sabiduría de la vida y su carácter.

A mi hermana, que cada vez se crece ante mis ojos, para mi seguirá siendo querida.

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

A la personas de la Facultad Regional "Mártires de Artemisa", que desde su inicio vienen dándome que hacer y dejándome ser incansable con ellos, buscando el modo de estar mejor en esta facultad.

A mis compañeros de aula, a los amigos de cuarto, ha sido tanta la presión que se que ninguno salió igual a como entró, pero algo aprendimos.

Dedicatoria

A mis padres: que han sido mi apoyo y tanto han hecho por mí.

A mis amigos: que todos de una forma u otra me han apoyado en mi búsqueda de ser una mejor persona.

Resumen

A partir de los cambios realizados en la estructura política-económica de Cuba surge como acuerdo de los lineamientos dividir la provincia Habana en el gobierno de Mayabeque y Artemisa, esta última cuenta con una dirección de auditoría que se encarga del control y recopilación de información para su posterior utilización en la toma de decisiones. 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* como solución.

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

Palabras Claves: procesos, gestión de información, APA, auditoría, sistemas.

Introducción.....	11
Capítulo 1: Fundamentación Teórica.....	19
1.1. <i>Conceptos asociados al dominio del problema.</i>	<i>19</i>
1.2. <i>Sistemas de Gestión de la Información. Tendencias Actuales.</i>	<i>27</i>
1.3. <i>Tecnologías, herramientas, lenguajes de programación y metodologías asociadas al desarrollo del sistema.....</i>	<i>29</i>
1.4. <i>Arquitectura de software</i>	<i>43</i>
Capítulo 2: Características, análisis y diseño del sistema	46
2.1. <i>Características y principios funcionalidad del sistema.</i>	<i>46</i>
2.2. <i>Planificación de proyecto por roles.</i>	<i>47</i>
2.3. <i>Modelo de Dominio.</i>	<i>49</i>
2.4. <i>Lista de Reserva del Producto (LRP).</i>	<i>50</i>
2.5. <i>Historias de usuarios y tareas de ingeniería.</i>	<i>55</i>
2.6. <i>Plan de Releases.</i>	<i>59</i>
2.7. <i>Diseño con metáforas</i>	<i>60</i>
2.8. <i>Diagrama de componentes</i>	<i>61</i>
2.9. <i>Conclusiones del capítulo</i>	<i>63</i>
Capítulo 3: Adquisición y validación de los resultados del sistema.....	64
3.1. <i>Pruebas funcionales</i>	<i>64</i>
3.2. <i>Pruebas al sistema.....</i>	<i>65</i>
3.3. <i>Pruebas unitarias.....</i>	<i>65</i>
3.4. <i>Pruebas de caja blanca.....</i>	<i>66</i>
3.5. <i>Resultado de las pruebas.....</i>	<i>71</i>
3.6. <i>Aporte Social y Económico</i>	<i>71</i>
3.7. <i>Conclusiones del capítulo</i>	<i>71</i>
Conclusiones.....	72
Recomendaciones	73

Referencias bibliográficas	74
Bibliografía.....	76
Glosario de términos	77
Anexos.....	79

Índice de Figuras

Figura 1. Arquitectura en N-Capas.....	44
Figura 2. Diagrama del Modelo de Dominio.....	49
Figura 3. Diagrama de paquetes	61
Figura 4. Diagrama de componentes.....	62
Figura 5. Modelo 002.....	82
Figura 6. Modelo 003.....	82

Índice de Tablas

Tabla 1. Planificación de proyecto por roles.	47
Tabla 2. Lista de Reserva del Producto (LRP).....	50
Tabla 3. Historias de usuarios y tareas de ingeniería.....	56
Tabla 4. Plan de Releases.	60

Introducción

Las *Tecnologías de la Información y las Comunicaciones (TIC)* son incuestionables y están ahí, forman parte de la cultura tecnológica. La sociedad del conocimiento, como es definida actualmente, se caracteriza por el papel que juegan la ciencia y la tecnología en el desarrollo socio-económico, hasta el punto de considerarse que están abocadas a sustituir al factor capital en las relaciones de producción, por constituir el principal recurso productor de riqueza.

Algunas características que distinguen la transformación social que se viene produciendo actualmente viene dado por la aparición de tecnologías que integran la telefonía, el video, la televisión, la computadora y las redes globales de computadora (Internet) que posibilitan el acceso a un amplio caudal de información y el vasto impacto de su uso en el terreno de la educación como una herramienta de vital importancia tanto para enseñar como para aprender.

La humanidad se encuentra actualmente en una era donde la información y el conocimiento están considerados como un recurso estratégico de las organizaciones, Esta era se caracteriza por un vertiginoso desarrollo científico-técnico, sustentado en adelantos informáticos, las telecomunicaciones y la industria audiovisual, que han devenido en llamarse *Tecnologías de la Información y la Comunicación (TIC)*, produciendo saltos cualitativos que influyen en todos los sectores laborales, y sociales.

Un país que desee ser verdaderamente independiente debe garantizar a todos sus ciudadanos la oportunidad de adquirir conocimientos sobre la ciencia y la tecnología, debe fomentar la capacidad de dar a ambas un uso apropiado y de desarrollarlas para satisfacer necesidades colectivas.

Estas tecnologías básicamente proporcionan información, herramientas para su proceso y canales de comunicación. Comprenden a todas las tecnologías que permiten el manejo de información y facilitan diferentes formas de comunicación, las cuales pueden ser subdivididas en tecnologías de capacitación, almacenaje, procesamiento, comunicación y visualización. Las TIC impactan en todos los ámbitos de la vida humana por las características que aportan a la sociedad del conocimiento, y por todas las posibilidades que brindan al desarrollo de nuevas formas de organizarse, comunicarse, educar, enseñar, aprender y con ello la transformación y evolución de la sociedad.

La era de la información, período en que vive la sociedad, representa un profundo avance intelectual que repercute en todas las esferas tanto económicas, políticas como sociales. Se habla constantemente de esta nueva sociedad de la información, con organizaciones basadas en el aprendizaje, cuyo capital es el ser humano y que se sustenta en un desarrollo tecnológico sin precedentes, de los cuales grandes empresas planifican sus productos en función de la gestión de la información y del conocimiento. Por lo que demuestra que las tecnologías de la información y las telecomunicaciones no son más que un medio para transmitir y gestionar datos, información y conocimiento.

Los *Sistemas de Información (SI)* y la *Tecnología de la Información (TI)* que los soportan se han convertido en una parte esencial de los productos, servicios, operación y gestión de las organizaciones. La gestión de información ocupa, cada vez más, un espacio mayor en el desarrollo de las organizaciones. La información para la toma de decisiones no está únicamente localizada en las experiencias acumuladas del capital humano, sino que también se encuentra dispersa en redes electrónicas, que avanzan cada vez más en aquellas organizaciones que dan frente a los retos de la sociedad del conocimiento.

El desarrollo de las nuevas tecnologías para la gestión ha dado paso a herramientas informáticas de última generación, con diferentes funcionalidades que proporcionan resultados a corto plazo.

Cuba, con un proyecto de desarrollo que tiene como pilares la justicia social, la participación popular, la equidad y la solidaridad, ha diseñado e iniciado la aplicación de estrategias que permiten convertir los conocimientos y las TIC en instrumentos a disposición del avance y las profundas transformaciones revolucionarias.

Hoy el país se encuentra inmerso en un proceso de reordenamiento institucional, buscando una estructura compacta y funcional para los Organismos de la Administración Central del Estado. Recientemente fueron aprobados cambios en la División Político-Administrativa con el objetivo de elevar la atención a la población mediante una organización más racional y funcional de la Administración y el Gobierno; todo esto, enmarcado en la actualización del modelo económico cubano, aligerando estructuras y trámites; y ordenando su fuerza de trabajo. En este contexto, las TIC deben constituirse en poderosas herramientas que permitan alcanzar los objetivos planteados.

La Universidad de las Ciencias Informáticas (UCI) surgió como parte del proyecto de informatización que asume el país y tiene entre sus principales objetivos la formación de ingenieros informáticos altamente calificados. Debido a la política que sigue la universidad, de vincular directamente a sus estudiantes a la producción de *software* destinados tanto al mercado nacional como a la exportación, se ha creado una infraestructura productiva que le ha permitido convertirse en una universidad de excelencia, donde se vincula la docencia, la producción y la investigación.

En el año 2011 Cuba realizó una nueva distribución política administrativa debido a la gran cantidad de municipios que contenía la provincia de La Habana. Con esta nueva distribución surgen las provincias de Mayabeque y Artemisa. En esta última,

en el municipio cabecera se encuentra ubicado el órgano del Gobierno Provincial, el cual está dividido en 32 órganos de direcciones y una de ellas es la de Auditoría. Este órgano de dirección tiene la misión de ejecutar auditoría y otras acciones de control a los recursos materiales, humanos y financieros que dispone la Administración Provincial de Artemisa y las entidades subordinadas.

El proceso de gestión de información es defectuoso debido a que se realiza de forma manual, por vía telefónica, correo electrónico o mediante mensajeros de las entidades que se encuentran en los municipios de la provincia y entregan la información en formato duro o digital. En la entidad no existe una adecuada forma de gestionar la información. Se dificulta el análisis de los reportes ya que los datos se presentan en formatos “Excel” y “Word”, la información no está íntegra y es inconsistente.

Además existen problemas que comprometen la confiabilidad de los datos en el órgano de dirección. Cuando se necesita una información urgente se tiene que buscar de forma manual y dentro de grandes volúmenes de información contribuyendo a la falta de eficiencia. Además la información con la que se toman las decisiones muchas veces no se encuentra centralizada, sino en varias computadoras.

Esta mala gestión de la información ocasiona problemas como la falta de seguridad pudiendo existir borrado indebido de archivos o modificaciones no autorizadas así como pérdida y duplicado de dicha información principalmente en los procesos de deficiencias detectadas y medidas disciplinarias aplicadas a los cuadros, dirigentes y trabajadores. También existen deficiencias en cuanto a la rapidez con que se acceden a los datos pues estos son guardados en hojas de cálculo y en formato duro dispersos en la dirección de auditoría, causando grandes consecuencias para la toma de decisiones en la provincia.

Por lo expuesto anteriormente el proceso de gestión de la información llevado a cabo en la Dirección de Auditoría carece de la eficiencia y confiabilidad necesarias para la generación de modelos en el tiempo requerido resultando poco fiable la información registrada en los mismos.

Teniendo en cuenta las dificultades descritas anteriormente se define como

problema de investigación: ¿Cómo contribuir a la eficiencia y confiabilidad del proceso de gestión de la información en el Órgano de Dirección de Auditoría de la Administración Provincial de Artemisa?

Se define como **objeto de estudio:** La gestión de la información. Dentro del cual se enmarca como **campo de acción:** Los procesos de gestión de información a través de un servidor para el módulo de la Dirección de Auditoría de la Administración Provincial de Artemisa.

La presente investigación tiene como **objetivo general:** Desarrollar un servidor para el módulo de la Dirección de Auditoría que contribuya a la eficiencia y confiabilidad en el proceso de gestión de información de la Administración Provincial de Artemisa (APA).

Para complementar este objetivo general se definen los siguientes **objetivos específicos:**

1. Establecer los fundamentos teórico-metodológicos para el desarrollo de los procesos de gestión de información.
2. Desarrollar el servidor para el módulo de la Dirección de Auditoría basado en sus procesos de gestión de información.
3. Validar mediante pruebas de *software* los resultados obtenidos con la solución.

Como **idea a defender** se plantea lo siguiente: Mediante el diseño de un servidor para el módulo de la Dirección de Auditoría se contribuirá a mejorar la gestión de la información en la APA.

Para llevar a cabo la investigación de manera que se lograra el objetivo propuesto, así como para dar respuesta a la idea a defender se trazaron las **tareas de la investigación** que se enumeran a continuación:

1. Establecer los fundamentos teórico-metodológicos para el desarrollo de los procesos de gestión de la información.
2. Caracterizar el proceso de gestión de la información en la Dirección de Auditoría en cuanto a su eficiencia y confiabilidad.
3. Desarrollar el servidor para el módulo de la Dirección de Auditoría basado en sus procesos de gestión de información.
4. Validar la contribución lograda a través del cumplimiento de los requisitos propuestos, mediante el uso de técnicas con esta finalidad.

Métodos Teóricos.

Analítico - Sintético: Para entender y resumir la situación que presentan los servidores, además de interiorizar y sintetizar el estudio que se realiza a lo largo del capítulo uno, incluyendo el estudio del estado del arte sobre las herramientas más usadas en la gestión de información, así como herramientas a utilizar para el desarrollo del sistema y los principales conceptos relacionados con la investigación.

Histórico-lógico: Mediante el método histórico se analizó la trayectoria concreta de la teoría, así como su condicionamiento en los diferentes períodos de la historia. Permitió consultar la bibliografía necesaria para dar cumplimiento a las tareas de la investigación y resumir los aspectos fundamentales de cada una de ellas, además de visualizar teóricamente cómo fueron desarrollados los sistemas de gestión de información en los procesos de desarrollo.

Métodos Empíricos.

Observación: Para la percepción selectiva de las restricciones y propiedades del sistema y para la determinación de la problemática que dio origen a la investigación.

Análisis Documental: se utilizó para el análisis de toda la documentación existente en la Dirección de Auditoría para tener información sobre los procesos que allí se realizan. Ver **(Anexo 1)**.

Entrevista: Este método se utilizó para obtener información a través del personal que labora en la Dirección de Auditoría de la Administración Provincial de Artemisa, dicha información hace referencia a cómo ocurren los procesos de gestión y búsqueda de información dentro de la dirección.

Para poder realizar la entrevista se asumió una **población** de 12 trabajadores de la Dirección de Auditoría con la selección de una **muestra** de 7 personas, mediante la técnica no probabilística de muestreo intencional, lo cual representa aproximadamente un 58% de la población. Ver **(Anexo 2)**.

Variable Independiente: Gestión de la información

Variabes dependientes: Eficiencia y Confiabilidad.

Actualidad y Necesidad.

En la actualidad la dirección no cuenta con un sistema informático que controle la gran cantidad de información generada en cuanto al control y ejecución de las actividades de auditoría, lo cual trae como consecuencia que el trabajo de las personas que laboran en dicha dirección se vea limitado, existiendo pérdida de tiempo, debido a los motivos expuestos anteriormente surge la necesidad de realizar un servidor para la gestión de la información dentro del módulo de la Dirección de Auditoría de la APA.

A nivel mundial existen sistemas de gestión de información que se encargan de hacer más factible el proceso llevado a cabo en distintas instituciones que manejan grandes cúmulos de información, estos, a pesar de tener características similares con la solución a implementar, contienen marcadas diferencias pues dichos sistemas responden solamente a los intereses trazados en instituciones de acuerdo a sus necesidades, además la mayoría son desarrollados bajo licencias de *software* propietario. No existen sistemas aplicables al entorno y la realidad del problema que se desea resolver, de esta manera se evidencia la novedad y el aporte del trabajo actual en el marco de la problemática a solucionar, adecuado a las condiciones de Cuba para implementar *software*.

La investigación realizada posibilitará como **aporte práctico** el desarrollo de un servidor para la gestión de la información dentro del módulo de la Dirección de Auditoría de la Administración Provincial de Artemisa.

Estructura del Documento

Capítulo 1: Fundamentación teórica: Se realiza un estudio del estado del arte que permita conocer la situación relacionada con el diseño de sistemas de gestión de la información, plasmando además, los principales conceptos referentes a dicho proceso.

Capítulo 2: Características, análisis y diseño del sistema: Se realiza una propuesta del sistema, describiendo cómo debe funcionar y se destacan sus características distintivas; además, se especifican sus requisitos funcionales y no funcionales y se muestran los modelos necesarios para la creación del diseño de la herramienta o sistema a realizar.

Capítulo 3: Adquisición y validación de los resultados del sistema o pruebas realizadas. Se realiza una evaluación de la propuesta a partir de pruebas piloto, cuya evaluación permite seguidamente la valoración de la propuesta realizada.

Capítulo 1: Fundamentación Teórica

Introducción.

En el desarrollo de este capítulo se realizará un estudio de los sistemas para la gestión de la información, además de una valoración de las posibles herramientas, tecnologías, lenguajes de programación y metodologías utilizadas para el análisis, diseño e implementación de dicho sistema en la Facultad Regional Mártires de Artemisa. Por ello es necesario conocer primero todas sus características, para saber como desarrollarlas mejorando así las funcionalidades que ellas ofrecen, teniendo en cuenta las prestaciones que debe tener finalmente el sistema.

Por otra parte se deben conocer bien los conceptos relacionados a la utilización de las diferentes tecnologías y el manejo de la información de forma que todo quede especificado correctamente en el marco teórico de la información.

1.1. Conceptos asociados al dominio del problema.

La **Auditoría** es la recopilación de datos sobre información cuantificable de una entidad económica para determinar e informar sobre el grado de correspondencia entre la información y los criterios establecidos.

Es un proceso a través del cual un sujeto lleva a cabo la revisión de un objeto con el fin de emitir una opinión acerca de la fidelidad de éste (grado de correspondencia), a un patrón o estándar establecido.

Es una función asesora técnica al servicio de la Dirección Superior de la Empresa, cuya misión fundamental es apoyar la gestión empresarial en lo relativo a las necesidades de información, para el proceso de toma de decisiones útiles para desarrollos futuros.

Eficiencia

Capacidad de alcanzar un objetivo fijado con anterioridad en el menor tiempo posible y con el mínimo uso posible de los recursos, lo que supone una optimización.

Simón Andrade, define la eficiencia de la siguiente manera: "expresión que se emplea para medir la capacidad o cualidad de actuación de un sistema o sujeto económico, para lograr el cumplimiento de objetivos determinados, minimizando el empleo de recursos". (ANDRADE, 2005).

Una empresa, organización, producto o persona es "eficiente" cuando es capaz de obtener resultados deseados mediante la óptima utilización de los recursos disponibles y el cumplimiento de todos los objetivos propuestos.

Confiabilidad

La confiabilidad se puede definir como la probabilidad en que un producto realizará su función prevista sin incidentes por un período de tiempo especificado y bajo condiciones indicadas.

Según Enrique Ellmann, la confiabilidad es la "capacidad de un elemento de desempeñar una función requerida, en condiciones establecidas". Habremos logrado la confiabilidad requerida cuando el elemento hace lo que queremos que haga. Al decir elemento podemos referirnos a una máquina, una planta industrial, un proceso, un rodado, un sistema y también a una persona. (Ellmann, 2008).

Proceso.

Oscar Barros señala que "un proceso es un conjunto de tareas lógicamente relacionadas que existen para conseguir un resultado bien definido dentro de un negocio; por lo tanto, toman una entrada y le agregan valor para producir una salida. Los procesos tienen entonces clientes que pueden ser internos o externos, los cuales reciben a la salida, lo que puede ser un producto físico o un servicio. Éstos establecen las condiciones de satisfacción o declaran que el producto o

servicio es aceptable o no” (Barros, 1994).

En informática, cuando se habla de proceso se refiere a un concepto que se maneja dentro del ámbito de los sistemas operativos, como por ejemplo *Windows*, *GNU/Linux*, *Mac OS X*, entre otros. En este contexto, un proceso se refiere a las instrucciones que ejecutará el microprocesador mientras lee un programa determinado. Esto también implica a la memoria reservada y a sus contenidos, el estado de ejecución en determinado momento, y la información que permite al sistema operativo planificar.

Sistema.

Un **sistema** (del latín *systema*) es un objeto compuesto cuyos componentes se relacionan con al menos algún otro componente; puede ser material o conceptual. Todos los sistemas tienen composición, estructura y entorno, pero sólo los sistemas materiales tienen mecanismo, y sólo algunos sistemas materiales tienen forma.

Un **sistema informático** como todo sistema, es el conjunto de partes interrelacionadas, *hardware*, *software* y de recurso humano que permite almacenar y procesar información. El *hardware* incluye computadoras o cualquier tipo de dispositivo electrónico inteligente, que consisten en procesadores, memoria y sistemas de almacenamiento externo. El *software* incluye al sistema operativo, *firmware* y aplicaciones. Por último el soporte humano incluye al personal técnico que crean y mantienen el sistema y a los usuarios que lo utilizan.

Módulo.

Es una pieza o un conjunto unitario de piezas que, en una construcción, se repiten para hacerla más sencilla, regular y económica. El módulo, por lo tanto, forma parte de un sistema y mantiene algún tipo de relación o vínculo con el resto de los componentes. (Gómez, 2010)

En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso).

Gestión.

La gestión (del latín *gestio*) hace referencia a la acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera.

Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización. Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener. (Salazar, 2000).

La gestión son las actividades coordinadas para dirigir y controlar una organización. (QUIROGA, 2011)

Importancia de la gestión: Realizar esta operación es muy importante ya que es una forma de dirigir, controlar y seguir muy de cerca los pasos que se estén llevando a cabo para realizar una operación.

Cuando se realiza este proceso se llevan a cabo un conjunto de actividades para lograr un determinado propósito que puede ser la administración de una empresa, negocio o de una determinada cantidad de información; estas actividades hacen que las cosas funcionen, con la capacidad para generar procesos de transformación de la realidad.

Información.

En sentido general, la información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

La Real Academia Española define este término como “la comunicación o adquisición de conocimientos que permiten ampliar o precisar los que ya se poseen sobre una materia determinada”. (Real Academia Española)

Se propone el siguiente concepto de información: conjunto organizado de datos procesados y organizados de un tema específico, la cual puede ser mostrada a través de textos, esquemas, gráficos, imágenes o tablas que deben transmitir el contenido que se está abordando. Esta constituye un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

La **información** es un "agente activo", un principio universal que especifica el significado de las cosas e indica, mediante códigos, los modelos del pensamiento humano. (GOÑI, 2008)

Desde el punto de vista de la ciencia de la computación, la información es un conocimiento explícito extraído por seres vivos o sistemas expertos como resultado de interacción con el entorno o percepciones sensibles del mismo entorno. En principio la información, a diferencia de los datos o las percepciones sensibles, tienen estructura útil que modificará las sucesivas interacciones del ente que posee dicha información con su entorno.

Formas de encontrar la información: Desde la digitalización de la información, ésta se puede encontrar en cualquier computadora y se puede transmitir hacia otra computadora o hacia cualquier dispositivo de almacenamiento electrónico. Esto permitió que se pudieran manipular grandes volúmenes de información y dio paso a la creación de sistemas capaces de procesarla, acorde a los intereses y necesidades particulares de cada individuo. Además también surgió Internet como fuente abastecedora de toda información o de sistemas capaces de procesarla.

Grupos que manejan la información.

Organización: sistema social conformado por individuos y grupos de individuos, que dotados de recursos y dentro de un determinado contexto, desarrolla regularmente un conjunto de tareas orientadas por valores comunes hacia la obtención de un determinado fin.

Institución: cuerpos normativos, jurídicos y culturales, conformados por un conjunto de ideas, creencias, valores y reglas que condicionan las formas de intercambio social.

Entidades gubernamentales: son empresas e instituciones que dependen del Estado en una economía nacional. La actividad económica del sector público abarca todas aquellas actividades que el estado y sus empresas posee o controla. El papel y el volumen del sector público dependen en gran medida de lo que en cada momento se considera que constituye el interés público.

Importancia de la información: constituye el renglón fundamental de cada esfera de la vida y en una empresa es el bien máspreciado, por tanto el que más se debe proteger porque en sí forma parte de la vida y obra de la institución y de los procesos que se llevan a cabo en ella; además puede ser gestionada.

Sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

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

El término **Sistemas de Información** hace referencia a un concepto genérico que

tiene diferentes significados según el campo del conocimiento al que se aplique dicho concepto, a continuación se enumeran algunos de dichos campos y el sentido concreto que un Sistema de Información tiene en dicho campo:

- En informática, un sistema de información es cualquier sistema o subsistema de equipo de telecomunicaciones o computacional interconectados y que se utilicen para obtener, almacenar, manipular, administrar, mover, controlar, desplegar, intercambiar, transmitir o recibir voz y datos, e incluye tanto los programas de computación ("*software*" y "*firmware*") como el equipo de cómputo.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de dicha propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operadores preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.

Salida de Información: La salida es la capacidad de sacar la información procesada o bien datos de entrada al exterior. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfaz automática de salida.

Gestión de la información e información para la gestión.

Se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades.

La información para la gestión y la gestión de la información son dos conceptos diferentes; la información para la gestión es un tipo de información (los datos); la gestión de la información es un tipo de gestión (el sistema).

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. La información para la gestión es la información necesaria para tomar decisiones de gestión.

Esta información se recoge durante las fases de planificación e implementación. La información ayuda a detectar cualquier cosa que vaya mal en el proyecto. En consecuencia, los administradores pueden encontrar soluciones para asegurar el éxito.

Sistemas de gestión de la información.

Los Sistemas de Gestión de la Información (SGI) para la gestión del conocimiento constituyen hoy una alternativa de imprescindible presencia en cada organización. Al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización. (DÍAZ; AMADOR, 2007)

Con los sistemas de gestión de la información se busca que los usuarios realicen su trabajo fácilmente y con mayor calidad, repercutiendo esto en el nivel económico de las empresas.

Sistema de gestión de la información: proceso continuo, que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad. Este sistema trata de ser lo más amplio posible en su gestión de datos. Consta de cuatro etapas que son la ideación, planeación, implementación y control de la información. Se utiliza para el registro de las transacciones diarias y la generación de reportes que presentan información, es una herramienta informática que permite almacenar, mostrar y recuperar información, minimizando los errores que puedan ocurrir tratando la información de forma manual y que permita llevar el absoluto y total control de la misma.

Necesidad del surgimiento de los sistemas de gestión de la información.

Dada la creciente necesidad de la globalización, velocidad de cambios, competitividad y crecimiento cada vez mayor de la información surgen los sistemas de gestión de la información como una potente herramienta para poder comercializar y competir en el mercado; ya que estos sistemas bien implementados ayudan a eliminar las barreras en el mercado dando de esta forma un mayor beneficio para los clientes y demás personas beneficiadas; además de desarrollar el nivel de innovación, mejorando la capacidad operativa y dando mucha más claridad y desenvolvimiento dentro del mercado.

1.2. Sistemas de Gestión de la Información. Tendencias Actuales.

Debido al gran avance de las tecnologías de la informática y las comunicaciones, en el mundo existe una numerosa cantidad de sistemas de gestión, entre ellos los sistemas de gestión vinculados al sector de la economía, el control y la auditoría. Generalmente estos sistemas de gestión implantados en las instituciones gubernamentales, responden solamente a los intereses trazados en dichos centros, por lo que resultaría muy difícil adaptar uno de estos sistemas a otro centro de control y auditoría, siendo ello una de las causas fundamentales por la cual se

tengan que desarrollar los sistemas de gestión de información de acuerdo a cada institución.

Sistemas de gestión de información a nivel internacional.

Existen muchos países que utilizan los sistemas de gestión de la información, estos mismos son utilizados para sistemas gubernamentales como es el caso de Ecuador con el Sistema de Información para la Gobernabilidad Democrática (SIGOB), el cual permite gestionar toda la información acerca de las metas presidenciales, de la agenda de actividades interministeriales, del centro de gestión, de compromisos presidenciales, de decretos ejecutivos, de acuerdos y disposiciones interministeriales, de gabinetes sectoriales entre otras. (SIGOB, 2009).

El Sistema de Gestión de la Información Electrónica (EIMS), es un sistema integral para la gestión de las actividades de publicación electrónica y para la clasificación de palabras clave y metadatos en las publicaciones sobre agricultura, páginas *Web*, imágenes y vídeos. El *WAICENT* comenzó a utilizar el EIMS en 1999 cuando se le encargó la ambiciosa tarea de pasar el gran cúmulo de conocimientos de la *FAO* al formato electrónico *html*. Actualmente, la mayor parte de los departamentos de la *FAO*, así como su red de oficinas en todo el mundo, utilizan el sistema EIMS para organizar la vasta producción de información de la Organización. (FAO, 2012).

La creación de un Sistema de Información en el Departamento Nacional de Planeación (DNP), perteneciente a Colombia, permite informar a los usuarios de posibles cambios en el mismo.

El DNP es una entidad eminentemente técnica que impulsa la implantación de una visión estratégica del país en los campos social, económico y ambiental, a través del diseño, la orientación y evaluación de las políticas públicas colombianas, el manejo y asignación de la inversión pública y la concreción de las mismas en planes, programas y proyectos del Gobierno. (DNP, 2012).

Sistemas de gestión de información en Cuba.

En Cuba, se encuentran resultados valiosos como los sistemas de gestión de información de las Delegaciones Territoriales del CITMA en Las Tunas, en Camagüey, en Santiago de Cuba. A nivel de ramas, la Dirección de Ciencia del Ministerio de Educación Superior en colaboración con la Facultad de Matemática y Computación de la Universidad de La Habana, ejecutan un proyecto de desarrollo de sistema de información para el control de la actividad científica.

Todos estos sistemas de gestión de información son resultado de la necesidad de tomar decisiones gerenciales oportunas, pertinentes y en un plazo relativamente breve a partir de la información disponible.

El Sistema de Información para la Gestión de Programas de Ciencia e Innovación (SIPROGIT), es un ejemplo de sistema de gestión de la información en Cuba el cual fue concebido a partir de la experiencia acumulada por directivos y especialistas a cargo de la dirección estratégica del Sistema de Programas y Proyectos (SPP), de los Programas Nacionales de Ciencia e Innovación. Constituye un sistema de apoyo a la planificación, control, evaluación y proyección de los Programas y Proyectos de Ciencia e Innovación, acorde a las prioridades establecidas para un período determinado en Cuba y que contribuye a incrementar el nivel de integración del SPP.

1.3. Tecnologías, herramientas, lenguajes de programación y metodologías asociadas al desarrollo del sistema.

1.3.1. *Java Development Kit (JDK) 1.7.0.*

- Se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar (compilar, ejecutar, generar documentación, etc.) programas en lenguaje *Java*.

- Existen versiones del *JDK* para prácticamente todos los Sistemas Operativos y existen también distintos programas comerciales
- *Sun* distribuye gratuitamente el *JDK* “oficial” para los siguientes sistemas operativos: *Windows 95/98/NT*, *Solaris* y *Linux*
- Los *JDK* incorporan una herramienta de *Debugger* (detener la ejecución de un programa en la línea que se desee y poder conocer el valor de las variables en ese momento)
- *JRE* (*Java Runtime Environment*)
- Versión reducida del *JDK* que permite ejecutar código *Java*
- No permite compilar ni utilizar el *Debugger*
- Viene incluido en el *JDK* pero también puede instalarse separadamente
- Es de libre distribución

1.3.2. *Java*.

Java es un lenguaje orientado a objetos, eso implica que su concepción es próxima a la forma de pensar humana. También posee otras características muy importantes:

- Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas, son en realidad interpretadas por la máquina virtual de *java*. Siendo la máquina virtual de *java* la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: El mismo código *java* que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual *java*.
- Es un lenguaje seguro: La máquina virtual, al ejecutar el código *java*, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Para poder trabajar con java es necesario emplear un *software* que permita desarrollar en java. Estas pueden ser *JBuilder*, *Visual Age*, *Visual Café*, *NetBeans* y un conjunto de herramientas *shareware*, e incluso *freeware*, que permiten trabajar con *java*. Pero todas estas herramientas en realidad se basan en el uso de una herramienta proporcionada por *Sun*, el creador de *java*, que es el *Java Development Kit (JDK)*.

1.3.3. Entorno de desarrollo integrado (IDE).

Un **entorno de desarrollo integrado**, llamado también **IDE** (sigla en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (*GUI*). Los *IDEs* pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Los *IDE* proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como *C++*, *PHP*, *Python*, *Java*, *C#*, *Delphi*, *Visual Basic*, etc. En algunos lenguajes, un *IDE* puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

Actitudes en diferentes plataformas computacionales.

Muchos de los programadores de *Unix* argumentan que las herramientas de la línea de comandos *POSIX* constituyen una *IDE*, aunque una con un estilo diferente de la interfaz y en el entorno *Unix*. Además, muchos programadores de *Unix* utilizan *Emacs* o *Vim*.

En las distintas plataformas de *Microsoft Windows*, las herramientas de línea de comando para el desarrollo rara vez se utilizan. En consecuencia, hay muchas

soluciones comerciales y no comerciales, sin embargo cada uno tiene un diseño diferente que comúnmente crean incompatibilidades. La mayoría de los principales proveedores de compiladores para *Windows* siguen ofreciendo copias gratuitas de sus herramientas de línea de comando.

Además las herramientas del *software* libre *GNU* están disponibles en muchas plataformas, incluyendo *Windows*. Algunos entornos libres, como *Code Blocks*, *Eclipse*, *Lazarus*, *KDevelop* y *Netbeans*, que a su vez se desarrollan utilizando un lenguaje multiplataforma (por ejemplo, *Free Pascal* o *Java*), se ejecutan en diversas plataformas, incluyendo *Windows*, *GNU/Linux*, y *Mac OS X*.

1.3.4. *Maven* 2.2.1.

Maven es una herramienta de *software* para la gestión y construcción de proyectos *Java* creada por *Jason van Zyl*, de *Sonatype*, en 2002. Es similar en funcionalidad a *Apache Ant* (y en menor medida a *PEAR* de *PHP* y *CPAN* de *Perl*), pero tiene un modelo de configuración de construcción más simple, basado en un formato *XML*. Estuvo integrado inicialmente dentro del proyecto *Jakarta* pero ahora ya es un proyecto de nivel superior de la *Apache Software Foundation*.

Maven utiliza un *Project Object Model (POM)* para describir el proyecto de *software* a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de *Maven* es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar *plugins* de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos *Open Source* en *Java*, de *Apache* y otras organizaciones y desarrolladores. *Maven* provee soporte no sólo para obtener archivos de su repositorio, sino también para

subir artefactos al repositorio al final de la construcción de la aplicación, dejándola al acceso de todos los usuarios. Una caché local de artefactos actúa como la primera fuente para sincronizar la salida de los proyectos a un sistema local.

Maven está construido usando una arquitectura basada en *plugins* que permiten que utilice cualquier aplicación controlable a través de la entrada estándar. En teoría, esto podría permitir a cualquiera escribir *plugins* para su interfaz con herramientas como compiladores, herramientas de pruebas unitarias, etcétera, para cualquier otro lenguaje. En realidad, el soporte y uso de lenguajes distintos de *Java* es mínimo.

1.3.5. *NetBeans 7.0.1.*

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación *Java*. Existe además un número importante de módulos para extenderlo. *NetBeans IDE* es un producto libre y gratuito sin limitación de uso.

NetBeans es un proyecto de éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Microsystems* fundó el proyecto de código abierto *NetBeans* en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma *NetBeans* permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados *módulos*. Un módulo es un archivo *Java* que contiene clases de *java* escritas para interactuar con *NetBeans* y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma *NetBeans* pueden ser extendidas fácilmente por otros desarrolladores de *software*.

1.3.6. *Scrum*.

Scrum es un marco de trabajo para la gestión y desarrollo de *software* basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de *software*.

Aunque *Scrum* estaba enfocado a la gestión de procesos de desarrollo de *software*, puede ser utilizado en equipos de mantenimiento de *software*, o en una aproximación de gestión de programas: *Scrum* de *Scrums*.

Características de *Scrum*.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en *Scrum* son el *ScrumMaster*, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que incluye a los desarrolladores.

Roles en *Scrum*.

En *Scrum* se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas. El nombre de los grupos está inspirado en la siguiente frase: “En un plato de huevos con tocino el cerdo está comprometido, la gallina sólo está involucrada.”

De esta forma, los “cerdos” están comprometidos a desarrollar el *software* de forma regular y frecuente, mientras que todos los demás son 'gallinas' que sólo interesados en el proyecto, y si este falla, ellos no son los cerdos, es decir, ellos no fueron los que se comprometieron a hacerlo.

Las necesidades, deseos, ideas e influencias de los roles “gallina” se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto *Scrum*.

Roles "Cerdo".

El *Product Owner* representa la voz del cliente. Se asegura de que el equipo *Scrum* trabaja de forma adecuada desde la perspectiva del negocio. El *Product Owner* escribe historias de usuario, las prioriza, y las coloca en el *Product Backlog*.

ScrumMaster (o Facilitador): El *Scrum* es facilitado por un *ScrumMaster*, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El *ScrumMaster* se asegura de que el proceso *Scrum* se utiliza como es debido

Roles "Gallina".

Los roles gallina en realidad no son parte del proceso *Scrum*, pero deben tenerse en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (*stakeholders*). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada *sprint*.

Usuarios: Es el destinatario final del producto.

Stakeholders (Clientes, Proveedores, Inversores): Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del *sprint*.

Managers: Es la gente que establece el ambiente para el desarrollo del producto.

1.3.7. Programación extrema (XP).

La programación extrema o *eXtreme Programming (XP)* es un enfoque de la ingeniería de *software* por *Kent Deck* autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change (1999)*. Es el más destacado de los procesos ágiles de desarrollo de *software*. Al igual que éstos, la

programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. *XP* se define como especialmente adecuada para proyectos con requisitos imprecisos, muy cambiantes, y donde existe un alto riesgo técnico. (TORRES; LÓPEZ, 2003)

Los defensores de *XP* consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del *software*.

Valores.

Los Valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (*feedback*) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de *Extreme Programming Explained*. Los cinco valores se detallan a continuación:

- **Simplicidad:**

La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece.

- **Comunicación:**

La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código autodocumentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el código a medida que es modificado. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de como utilizar su funcionalidad. Los programadores se comunican constantemente gracias a la programación por parejas.

- **Retroalimentación (*feedback*):**

Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante. Considérense los problemas que derivan de tener ciclos muy largos, meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo.

- **Coraje o valentía:**

Muchas de las prácticas implican valentía. Una de ellas es siempre diseñar y programar para hoy y no para mañana. Esto es un esfuerzo para evitar empantanarse en el diseño y requerir demasiado tiempo y trabajo para implementar todo lo demás del proyecto. Por otra parte, valentía significa persistencia: un programador puede permanecer sin avanzar en un problema complejo por un día entero, y luego lo resolverá rápidamente al día siguiente, solo si es persistente.

- **Respeto:**

El respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en

el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código. Los miembros del equipo respetan el trabajo del resto, logrando una mejor autoestima en el equipo y elevando el ritmo de producción en el mismo.

1.3.8. Scrum-XP (SXP).

Scrum-XP o SXP surgió en Cuba, en la Universidad de las Ciencias informáticas (UCI) y no es más que la unión de *XP* y *Scrum*, para el logro de un buen desarrollo de *software*, propuesta en el 2007 por la ingeniera Malay Rodríguez Villar, y probada en los proyectos que trabajan con *software* libre, obteniendo buenos resultados. Esta metodología surgió con el nombre metodología ágil Gladys Marsi Peñalver Romero UNICORNIOS revisión 2 (MA-GMPR-UR2) la cual fue renombrada poco tiempo después.

La **metodología está dividida en cuatro fases**, que son precisamente la base de la estructura del nuevo expediente de proyecto, estas son:

Planificación-Definición: en esta fase se generan todos los documentos que se encuentran relacionados con la concepción inicial del sistema, así como la definición del mismo. También se incluyen algunos que están vinculados a la primera parte de los procesos de Ingeniería de *Software* tales como los relacionados con el negocio, los requisitos y el diseño.

Desarrollo: en la primera parte de esta fase se generan todos los documentos relacionados con la planificación de las iteraciones, y además se recogen las principales definiciones que se manejan en la metodología y otros términos de difícil entendimiento para los clientes, así como de las tareas a realizar durante la implementación. Además, se genera el código fuente en la etapa de implementación y, como última parte de esta etapa, los documentos relacionados con las pruebas.

Entrega: en esta fase se realiza la entrega del *software* y su documentación, generándose aquellos documentos que son imprescindibles para el entrenamiento y entendimiento del producto.

Mantenimiento: se realizan las actividades relacionadas con el soporte del *software* y se generan los documentos relacionados con los cambios que puedan ocurrir en el mismo.

Cada una de estas fases está compuesta por una serie de actividades tales como escribir la visión y escribir la reserva del producto que son las que generan los artefactos como la plantilla de concepción del sistema y la lista de reserva del producto que quedan incluidos en el nuevo expediente de proyecto. Estas actividades están recogidas en el guión de la metodología.

Para la construcción de este sistema se escogió la metodología **Scrum-XP**, esta es la unión de las metodologías mencionadas anteriormente *Scrum* y *XP*, la cual toma las ventajas de ambas metodologías. Se utiliza por ser la metodología más adecuada para el tipo de sistema que se está elaborando, además de ser una metodología ágil y que sirve para sistemas pequeños.

1.3.9. Lenguaje de Modelado *UML*.

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.

Un detalle importante a tener en cuenta es que *UML* no define métodos ni procesos, sino que como su nombre lo indica, "modela", detalla los artefactos del sistema. Es además, un lenguaje bastante comprensible para aquellas personas

que no poseen un conocimiento profundo sobre informática lo cual permite la comunicación y un mejor entendimiento entre los clientes y el equipo de desarrollo de un proyecto. Es lo suficiente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como los mecanismos de la ingeniería de *software*, como son la encapsulación y los componentes.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y las reglas para combinar tales elementos. Estos diagramas son los que se usan para dar solución al problema que se plantea. Los diagramas generados por el *UML* pueden clasificarse en dos tipos: estáticos y dinámicos.

1.3.10. Visual Paradigm 6.4.

Es una herramienta *CASE* que utiliza *UML* como lenguaje de modelado profesional la cual soporta el ciclo de vida completo del desarrollo de *software*, esta herramienta permite realizar ingeniería tanto directa como inversa además soporta múltiples usuarios trabajando sobre el mismo proyecto, garantiza la generación de la documentación del proyecto automáticamente en varios formatos como *Web* o *PDF* y cabe destacar igualmente su robustez, usabilidad y portabilidad. (CARVAJAL, 2008)

Visual Paradigm es una herramienta *CASE*: Ingeniería de *Software* Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos *UML* al instante, creando de manera simple toda la documentación. Incorpora el soporte para el trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los

cambios hechos por sus compañeros.

Propósito.

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Constituye una herramienta de *software* libre de probada utilidad para el analista.

Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Esta herramienta permite aumentar la calidad del *software*, a través de la mejora de la productividad en el desarrollo y mantenimiento del *software*. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del *software*, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del *Software*.

1.3.11. Subversion 1.6.12.

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.

Una característica importante de *Subversion* es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente, en cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la

posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones.

Ventajas.

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- La creación de ramas y etiquetas es una operación más eficiente. Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
- Puede ser servido mediante *Apache*, sobre *WebDAV/DeltaV*. Esto permite que clientes *WebDAV* utilicen *Subversion* de forma transparente.
- Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- Cuando se usa integrado a *Apache* permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (*SQL*, *LDAP*, *PAM*, etc.).

Carencias.

- El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.
- No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita llevar la cuenta de qué cambios se han realizado. Esto se resuelve siendo cuidadoso con los mensajes de *commit*.

Subversion no es un sistema de gestión de la configuración pero es posible implementar sobre *Subversion* buenas prácticas de gestión de la configuración utilizando la estructura habitual de *Subversion*, sin embargo no existe un sistema automático para obligar a que se cumplan.

1.3.12. *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).

1.4. *Arquitectura de software*

La arquitectura de *software*, tiene que ver con el diseño y la implementación de estructuras de *software* de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales. Para el desarrollo de esta investigación y una mayor comprensión de la misma se explica como está construida la arquitectura de *software* usada en la aplicación, la cuál esta conformada por n-capas como se representa en la siguiente figura.



Figura 1. Arquitectura en N-Capas.

Capa de presentación: Es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio o Lógica de negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica de negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Capa de acceso a datos: Capa que sirve como puente entre la capa lógica de negocio y el proveedor de datos. Esta capa pretende encapsular las especificidades

del proveedor de datos (tales como SQL, archivos XML, texto, hojas electrónicas), a la siguiente capa, para que si cambia el proveedor de datos solo necesitemos cambiar en una sola capa el proveedor de datos.

Capa de datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Capítulo 2: Características, análisis y diseño del sistema

Introducción.

Al incrementarse de manera precipitada los volúmenes de información llevando a las empresas e instituciones a una pérdida de tiempo a la hora de acceder y buscar dicha información, la cual es empleada para tomar decisiones importantes para la sociedad. Es por ello que surgen los sistemas de gestión de la información para ahorrar trabajo a la hora de acceder y modificar a la misma.

2.1. Características y principios funcionalidad del sistema.

El objetivo principal del sistema, es gestionar toda la información referente a las auditorías que se realizan en la provincia de Artemisa, para asegurar la eficiencia y la confiabilidad de los reportes e informes que se gestionan. La solución propuesta informatiza la gestión de los datos referentes a las auditorías que se realizan en los diferentes municipios de la provincia, contribuye a obtener la información del sistema de forma actualizada y segura, con el propósito de apoyar la certeza de los datos generados u obtenidos de la dirección de auditoría. La gestión de la información de los datos consiste en Insertar, Modificar, Eliminar y Buscar los datos referentes a las auditorías. Cuando se realizan auditorías es necesario llenar una serie de modelos para que queden registradas las deficiencias detectadas, las medidas disciplinarias que fueron aplicadas en determinada entidad, los presuntos hechos delictivos detectados, entre otros. Estos son los datos que se precisan en determinado momento eliminar, modificar o buscar, además de añadir. El sistema también ofrece la opción de generar reportes. El sistema cuenta con una interfaz principal que será asequible para cualquier usuario, garantizando que la gestión de los datos sea eficiente y segura.

2.2. Planificación de proyecto por roles.

Algunas de las personas involucradas en un proyecto de *software* pueden ser los clientes, usuarios finales, administrativos, desarrolladores entre otros. Un tipo de trabajador es el papel que un individuo puede desempeñar en el desarrollo de *software*. El término rol se emplea para hablar de los papeles que cumple determinado trabajador. A continuación se muestra la tabla de planificación de proyecto por roles.

Tabla 1. Planificación de proyecto por roles.

Rol	Responsabilidad	Nombre
Gerente	Dirige y controla las tareas del equipo. Toma las decisiones finales. Participa en la selección de objetivos y requerimientos. Controla el progreso y da seguimiento a cada iteración. Evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes.	Dania Fernandez
Cliente	Participa en las tareas que involucran la lista de reserva del producto.	Administración Provincial de Artemisa
Líder de Proyecto	Encargado de asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas y que todo funciona según lo planeado. Remueve los impedimentos que pudiera presentar el proyecto. Define y reduce los riesgos del producto, además de coordinar reuniones.	Maidel Ojeda

Administrador de la Calidad	Administra las acciones de calidad que se implementen en el proyecto. Encargado de llevar a cabo el control y aseguramiento de la calidad con el objetivo de detectar errores lo antes posible en la vida del proyecto, y así disminuir su impacto tanto económico como en tiempo del mismo.	Yanisleydi León William González
Administrador de la BD	Tiene el control total sobre el sistema de base de datos. Encargado de definir los esquemas así como de la concesión de autorizo para el acceso a los datos.	Yenisey de la Torre
Analista	Escribe las historias de usuario y las pruebas funcionales para validar su implementación.	Yanisleydi León William González
Programador	Elabora el código de las nuevas funcionalidades a implementar. Escribe las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y el resto del equipo.	Yanisleydi León William González
Diseñador	Encargados del diseño del sistema; así como el de los prototipos de interfaces, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Yanisleydi León William González
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.	Yanisleydi León William González
Encargado de	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas	Yanisleydi León

Pruebas	regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.	William González
---------	---	------------------

2.3. Modelo de Dominio.

Después de analizar las descripciones de los procesos, se llega a la conclusión de que el negocio que se está desarrollando no tiene claramente definidas las actividades ni las personas que responden ante estos procesos. Por lo anteriormente explicado se llega a la conclusión de utilizar el modelo del dominio, el cual permite mostrar visualmente al usuario los conceptos fundamentales que se manejan en el dominio del sistema. Posibilitando a usuarios, clientes y desarrolladores al uso de un lenguaje común de entendimiento. Todo lo anteriormente expresado se muestra en la siguiente figura.

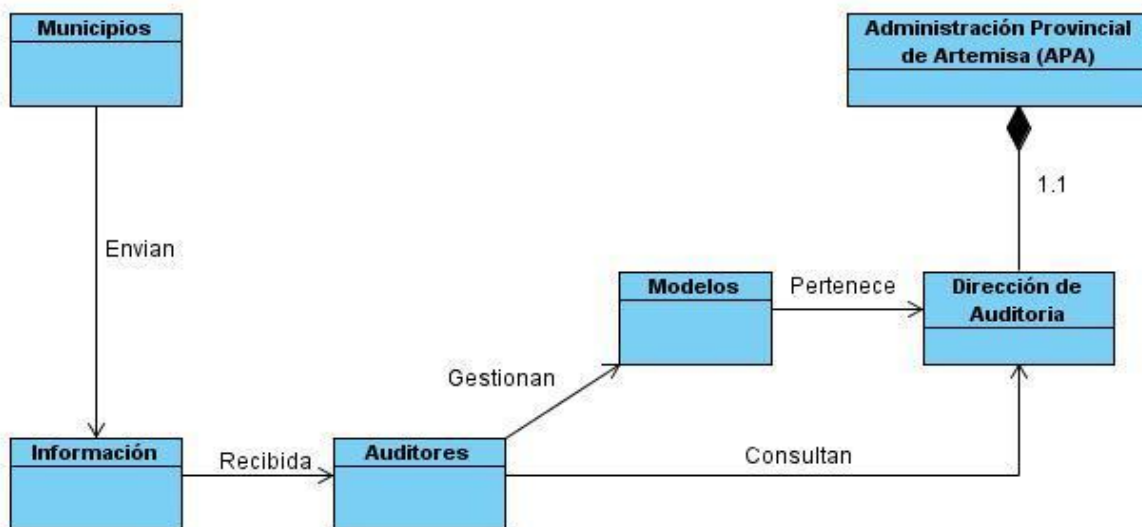


Figura 2. Diagrama del Modelo de Dominio.

Conceptos del modelo de dominio.

- Administración Provincial de Artemisa: Es la dirección central de la provincia Artemisa.
- Dirección de Auditoría: Representante de la dirección de auditoría de la Administración Provincial de Artemisa.
- Modelos: Planillas que son llenadas por los auditores con la información que es recibida de los municipios.
- Auditores: Usuarios que se encargan de gestionar la información contenida en los modelos.
- Información: Información obtenida de las auditorías y controles en los diferentes municipios.
- Municipios: Lugares auditados de donde se genera la información.

2.4. Lista de Reserva del Producto (LRP).

Colección organizada y priorizada de los requisitos sobre el producto cuyo objetivo fundamental es cumplir con las cualidades específicas del *software* a los cuales se les da cumplimiento progresivamente mediante iteraciones. Cada requerimiento posee una prioridad la cual es dada por el grado de necesidad del producto. En la siguiente tabla lista de reserva del producto se recogen una serie de datos necesarios para dar cumplimiento a todos los requerimientos.

Tabla 2. Lista de Reserva del Producto (LRP).

Prioridad	Ítem *	Descripción	Estimación	Estimado por
-----------	-----------	-------------	------------	-----------------

Muy Alta				
	1	Insertar información sobre la Situación de presuntos hechos delictivos.	3 días	Analista
	2	Modificar información sobre la Situación de presuntos hechos delictivos.	2 días	Analista
	3	Eliminar información sobre la Situación de presuntos hechos delictivos.	1 día	Analista
	4	Buscar información sobre la Situación de presuntos hechos delictivos.	1 día	Analista
	5	Insertar información sobre los reportes de la acción de control con presuntos hechos delictivos.	3 días	Analista
	6	Modificar información sobre los reportes de la acción de control con presuntos hechos delictivos.	2 días	Analista
	7	Eliminar información sobre los reportes de la acción de control con presuntos hechos delictivos.	1 día	Analista
	8	Buscar información sobre los reportes de la acción de control con presuntos hechos delictivos.	1 día	Analista
	9	Insertar información sobre el Resumen de las acciones de control.	3 días	Analista
	10	Modificar información sobre el Resumen de las acciones de control.	2 días	Analista

Capítulo 2

11	Eliminar información sobre el Resumen de las acciones de control.	1 día	Analista
12	Buscar información sobre el Resumen de las acciones de control.	1 día	Analista
13	Insertar información sobre el Análisis de la situación del completamiento de la Plantilla.	3 días	Analista
14	Modificar información sobre el Análisis de la situación del completamiento de la Plantilla.	2 días	Analista
15	Eliminar información sobre el Análisis de la situación del completamiento de la Plantilla.	1 día	Analista
16	Buscar información sobre el Análisis de la situación del completamiento de la Plantilla.	1 día	Analista
17	Insertar información sobre la aplicación de las medidas disciplinarias.	3 días	Analista
18	Modificar información sobre la aplicación de las medidas disciplinarias.	2 días	Analista
19	Eliminar información sobre la aplicación de las medidas disciplinarias.	1 día	Analista
20	Buscar información sobre la aplicación de las medidas disciplinarias.	1 día	Analista
21	Insertar información sobre las Principales Deficiencias detectadas en Acciones de	3 días	Analista

		Control.		
	22	Modificar información sobre las Principales Deficiencias detectadas en Acciones de Control.	2 días	Analista
	23	Eliminar información sobre las Principales Deficiencias detectadas en Acciones de Control.	1 día	Analista
	24	Buscar información sobre las Principales Deficiencias detectadas en Acciones de Control.	1 día	Analista
	25	Insertar información sobre los resultados de acciones de control a empresas en perfeccionamiento empresarial.	3 días	Analista
	26	Modificar información sobre los resultados de acciones de control a empresas en perfeccionamiento empresarial.	2 días	Analista
	27	Eliminar información sobre los resultados de acciones de control a empresas en perfeccionamiento empresarial.	1 día	Analista
	28	Buscar información sobre los resultados de acciones de control a empresas en perfeccionamiento empresarial.	1 día	Analista
Alta				
	29	Generar modelo sobre la Situación de presuntos hechos delictivos.	3 días	Analista

Capítulo 2

	30	Generar modelo sobre la Acción de control con presuntos hechos delictivos.	3 días	Analista
	31	Generar modelo sobre el Resumen de las acciones de control.	3 días	Analista
	32	Generar modelo sobre el Análisis de la situación del completamiento de la Plantilla.	3 días	Analista
	33	Generar modelo sobre la Aplicación de las medidas disciplinarias.	3 días	Analista
	34	Generar modelo sobre las Principales Deficiencias detectadas en Acciones de Control.	3 días	Analista
	35	Generar modelo sobre los Resultados de acciones de control a empresas en perfeccionamiento empresarial.	3 días	Analista
Baja				
Requisitos no funcionales				
	36	Tener un ordenador con más de 1 GB de memoria RAM.		Gerente de Proyecto

	37	Interfaz amigable.		Gerente de Proyecto
	38	El sistema debe proveer de forma ordenada y detallada las funcionalidades del sistema.		Gerente de Proyecto
	39	Las salidas del sistema deben de ser exactas evitando los errores en el procesamiento de los datos.		Gerente de Proyecto
	40	El sistema debe ser escalable de modo que futuras funcionalidades puedan ser implementadas e incorporadas.		Gerente de Proyecto
	41	La información manejada por el sistema estará protegida de acceso no autorizado y divulgación a terceras personas ya que los usuarios del sistema deben autenticarse antes de realizar cualquier actividad.		Gerente de Proyecto

2.5. Historias de usuarios y tareas de ingeniería.

En la metodología SXP las historias de usuario son las encargadas de describir las tareas que debe hacer el sistema, dependiendo de las especificaciones que realice el cliente. Se escriben con un lenguaje común y con palabras concisas y significativas, abreviando para no exceder su tamaño. Serán la base para la futura realización del *software* y posteriormente de las pruebas de aceptación por parte del cliente. En la siguiente tabla se muestran los datos de las historias de usuarios.

Tabla 3. Historias de usuarios y tareas de ingeniería .

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar Información del reporte de la situación de presuntos hechos delictivos.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanisleydi León Rodríguez Williams González Montesino	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Muy Alto	Puntos Reales:
Descripción: El objetivo de la presente historia de usuario es añadir, modificar, buscar y eliminar toda la información referente la situación de presuntos hechos delictivos.	
Observaciones: Ninguna.	
Prototipo de interfaz Ver (Anexo 3).	
Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Implementación de la funcionalidad Insertar información del reporte de la situación de presuntos hechos delictivos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 30/01/2012	Fecha Fin: 02/02/2012
Programador Responsable: Yanisleydi León Rodríguez	

Williams González Montesino

Descripción: A partir de la investigación realizada se sientan las bases para la implementación de esta funcionalidad sobre la dirección de auditoría.

Tarea de Ingeniería

Número Tarea: 1.2

Número Historia de Usuario: HU_1

Nombre Tarea: Implementación de la funcionalidad Eliminar información del reporte de la situación de presuntos hechos delictivos.

Tipo de Tarea: Desarrollo

Puntos Estimados: 1

Fecha Inicio: 02/02/2012

Fecha Fin: 04/02/2012

Programador Responsable: Yanisleydi León Rodríguez

Williams González Montesino

Descripción: A partir de la investigación realizada se sientan las bases para la implementación de esta funcionalidad sobre la dirección de auditoría.

Tarea de Ingeniería

Número Tarea: 1.3

Número Historia de Usuario: HU_1

Nombre Tarea: Implementación de la funcionalidad Modificar información del reporte de la situación de presuntos hechos delictivos.

Tipo de Tarea: Desarrollo

Puntos Estimados: 1

Fecha Inicio: 04/02/2012

Fecha Fin: 05/02/2012

Programador Responsable: Yanisleydi León Rodríguez

Williams González Montesino

Descripción: A partir de la investigación realizada se sientan las bases para la implementación de esta funcionalidad sobre la dirección de auditoría.

Tarea de Ingeniería

Número Tarea: 1.4

Número Historia de Usuario: HU_1

Nombre Tarea: Implementación de la funcionalidad Buscar información de los reportes de la situación de presuntos hechos delictivos.

Tipo de Tarea: Desarrollo

Puntos Estimados: 1

Fecha Inicio: 05/02/2012

Fecha Fin: 06/02/2012

Programador Responsable: Yanisleydi León Rodríguez

Williams González Montesino

Descripción: A partir de la investigación realizada se sientan las bases para la implementación de esta funcionalidad sobre la dirección de auditoría.

Historia de Usuario

Número: HU_8

Nombre Historia de Usuario: Generar reportes sobre la situación de presuntos hechos delictivos.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Yanisleydi León Rodríguez

Iteración Asignada: 3

Williams González Montesino

Prioridad en Negocio: Alta

Puntos Estimados: 0.42

Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: La presente historia de usuario tiene como objetivo generar los reportes sobre la situación de presuntos hechos delictivos ocurridos en los municipios de la provincia de Artemisa.	
Observaciones: Para que sea posible se debe realizar con anterioridad la inserción de los datos en el reporte de la acción de control con presuntos hechos delictivos.	
Prototipo de interface: Ver (Anexo 4).	
Tarea de Ingeniería	
Número Tarea: 8.1	Número Historia de Usuario: HU_8
Nombre Tarea: Implementación de la funcionalidad Generar reporte sobre la situación de presuntos hechos delictivos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.42
Fecha Inicio: 19/03/2012	Fecha Fin: 22/03/2012
Programador Responsable: Yanisleydi León Rodríguez Williams González Montesino	
Descripción: Se efectuará un estudio sobre el proceso de cómo realizar la funcionalidad de generar reportes sobre la situación de presuntos hechos delictivos.	

2.6. Plan de Releases.

En este paso se define el plan de releases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene

en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de historias se llegó a la siguiente planificación:

Tabla 4. Plan de Releases.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
2	En esta iteración se implementarán las historias de usuario que tienen prioridad muy alta.	HU_1, HU_2, HU_3, HU_4, HU_5, HU_6, HU_7	7 Semanas.
3	En esta iteración se desarrollarán las historias de usuario de prioridad alta y se integrarán con las historias de usuario ya implementadas.	HU_8, HU_9, HU_10, HU_11, HU_12, HU_13, HU_14	3 Semanas.

2.7. Diseño con metáforas

Define un conjunto de nombres que operan como vocabulario para hablar sobre el dominio del problema y describe como debería funcionar el sistema. Es el diseño de la solución más simple que pueda ser implementado. A partir de él se genera el artefacto Modelo de Diseño que está compuesto por el diagrama de paquetes, que representa dicho diseño. El diagrama de paquetes muestra las dependencias lógicas entre los componentes. Permite modelar cualquier sistema de *software* sin importar su nivel de complejidad. A continuación se muestra el diagrama de paquetes para el servidor del sistema que se propone.

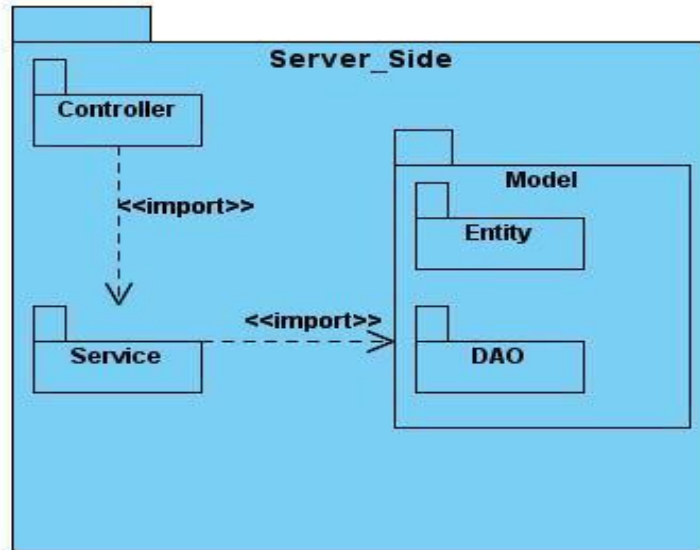


Figura 3. Diagrama de paquetes

2.8. Diagrama de componentes

Son utilizados para describir elementos físicos del sistema y relaciones existentes entre los mismos. Esta generalmente compuesto por bibliotecas, archivos, paquetes, entre otros. Además se representa todos los elementos de tipo *software* que van a integrar la aplicación (interrelacionándose). En la siguiente figura se muestra el diagrama de componentes del servidor para el módulo de la Dirección de Auditoría de la Administración Provincial de Artemisa.

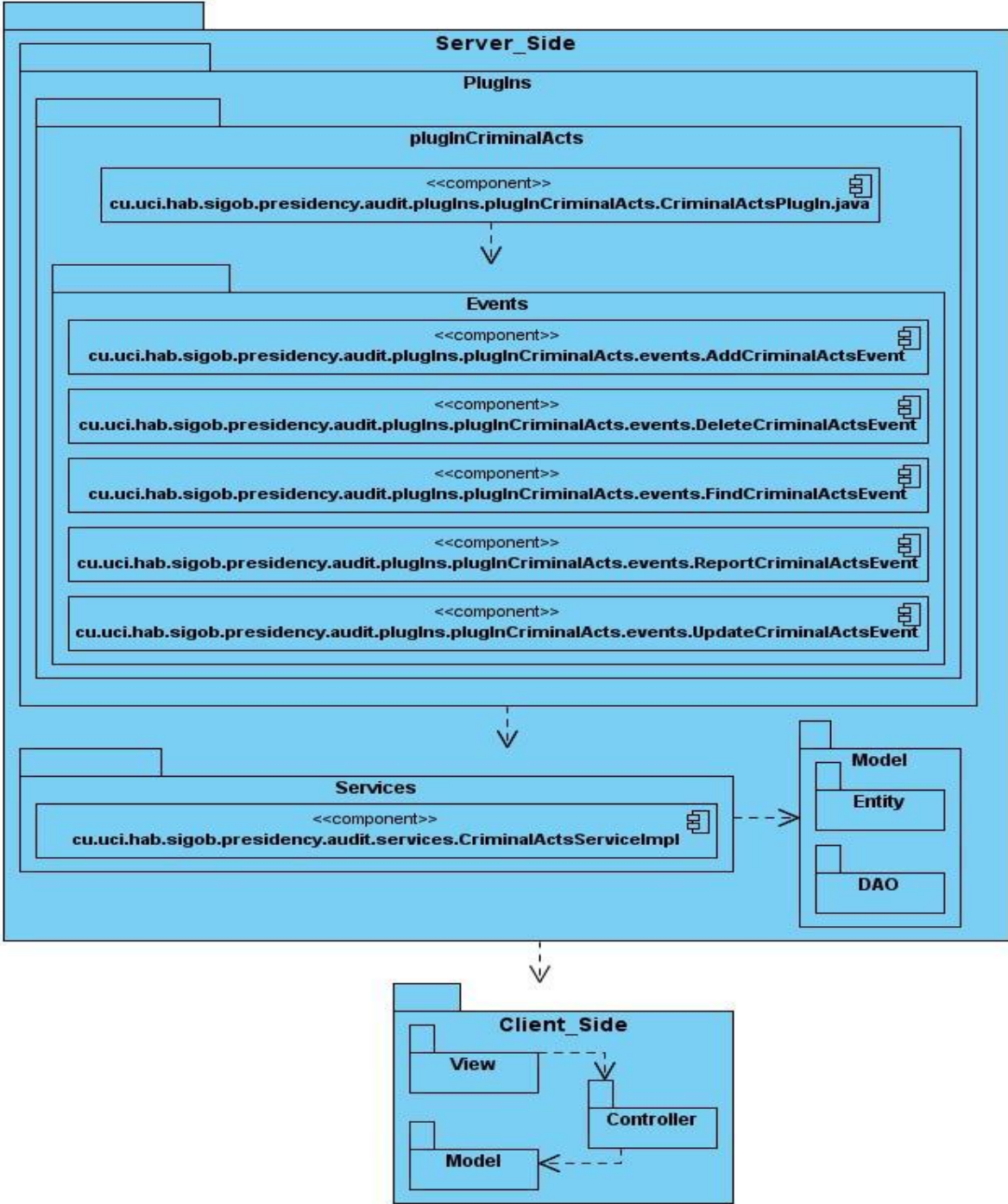


Figura 4. Diagrama de componentes

2.9. Conclusiones del capítulo

En este capítulo se hizo referencia a las principales características del negocio delimitando y describiendo los procesos existentes en el mismo. A partir de cada uno de estos procesos llegando así a definir las actividades que serán automatizadas las cuales junto con otras especificaciones conforman los requisitos funcionales los cuales serán implementados y solucionados a medida que se desarrolle el producto. Se delimitaron además los requisitos no funcionales los cuales debe cumplir el sistema. Una vez desarrollados tanto los requisitos funcionales como los no funcionales el *software* creado entraría en la fase de prueba.

Capítulo 3: Adquisición y validación de los resultados del sistema

Introducción

Las pruebas de *software* son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador, estas se integran dentro de las diferentes fases del ciclo del *software* dentro de la Ingeniería de *software*. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir los posibles errores.

El período de pruebas realizado a todo *software* es de vital importancia para lograr asegurar el correcto funcionamiento del mismo de forma tal que una vez entregado al cliente incurra en la menor cantidad de inconvenientes. Es importante realizar pruebas desde el mismo instante en que se comienza a implementar las funcionalidades para controlar futuros errores.

Existen varios tipos de pruebas que pueden ser realizadas a un sistema entre ellas las pruebas de caja negra que ejercitan los requisitos funcionales desde el exterior del módulo, las pruebas de caja blanca están dirigidas a las funciones internas. Sin embargo si se trata de desarrollo ágil con pruebas constantes, es interesante destacar la utilización de herramientas que automatizan esta actividad.

3.1. Pruebas funcionales

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el *software*. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Estas son esenciales para determinar si el producto de *software* desarrollado cumple con las funcionalidades requeridas por el cliente. Las mismas

validan que la aplicación funcione como un todo.

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, sino en el análisis de los datos de entrada y en los de salida. A continuación se describen un conjunto de pruebas diseñadas y ejecutadas para validar la solución propuesta. Fueron realizadas las pruebas a las Historias de Usuario pertenecientes al Módulo de Auditoría; que constituyen funcionalidades importantes para el cliente.

3.2. Pruebas al sistema

Las pruebas de sistema buscan discrepancias entre el programa y sus objetivos o requerimientos, enfocándose en los errores hechos durante la transición del proceso al diseñar la especificación funcional. Esto hace a las pruebas de sistema un proceso vital de pruebas, ya que en términos del producto, número de errores hechos, y severidad de esos errores, es un paso en el ciclo de desarrollo generalmente propenso a la mayoría de los errores. Las pruebas del sistema tienen un propósito particular: comparar el sistema o el programa con sus objetivos originales (requerimientos funcionales y no funcionales).

3.3. Pruebas unitarias

Sirven para utilizar otro código fuente, llamando directamente a los métodos de una clase pasando los parámetros apropiados y comparar los valores que se generan con los valores esperados. (Microsoft Corporation. MSDN, 2011)

La realización de las pruebas unitarias permite comprobar la correcta ejecución de un segmento de código de la aplicación, examinando el estado del programa en varios puntos, determinando si el estado real coincide con el esperado.

Con la realización de estas pruebas unitarias cuyo objetivo principal es probar el correcto funcionamiento del código fuente, se comprobó que existiera un estándar

de código adecuado, en el cual, se observa que cada funcionalidad debe ser comentareada para su mejor entendimiento, que el tiempo de ejecución fuera el menor posible, que cumpliera con la sintaxis del lenguaje y que se validará la aplicación en su totalidad. Estas pruebas arrojaron resultados satisfactorios en la aplicación. Es importante destacar que las pruebas unitarias no descubrirán todos los errores del código, solo 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.

3.4. Pruebas de caja blanca

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Las pruebas de caja blanca intentan garantizar que se ejecutan al menos una vez todos los caminos independientes de cada módulo así como que sean utilizadas las decisiones en su parte verdadera y en su parte falsa. (ROJAS Y BARRIOS, 2011)

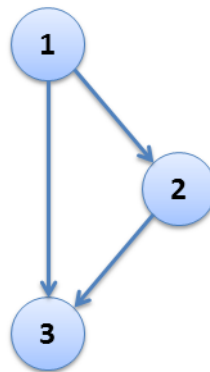
Las pruebas de caja blanca intentan garantizar que:

1. Se ejecutan todos los bucles en sus límites.
2. Se utilizan todas las estructuras de datos internas.

En esta aplicación se realizaron las pruebas de caja blanca en las cuales se comprobaron los flujos de ejecución dentro de cada unidad como clases, plugins, eventos y servicios. Una de las funciones aplicadas fueron las de controlar el flujo de datos, los caminos básicos y su bifurcación.

```
public void processEvent(FindApplicationDisciplinaryActionsEvent aEvent,
    C2SResponseEvent aResponseEvent) throws Exception {
```

1	<pre>try{ aResponseEvent.getArgs().setList("list", getService().findApplicationDisciplinaryActions(aEvent)); }</pre>
2	<pre>catch (Exception ex) { aResponseEvent.getArgs().setString("aResponse", ex.getMessage()); }</pre>
3	<pre>}</pre>



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

$$CC = 3 - 3 + 2$$

$$CC = 2$$

Caminos Básicos

C1 → 1 – 3 C2 → 1 – 2 – 3

Caso de Prueba para C1

<pre>try{ aResponseEvent.getArgs().setList("list", getService(). findApplicationDisciplinaryActions(aEvent));</pre>	<pre>true</pre>
---	-----------------

}	
Resultado	El método ha recorrido el camino adecuado devolviendo una lista con los componentes encontrados

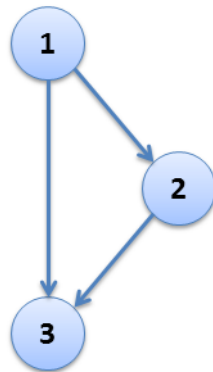
Caso de Prueba para C2

<pre>try{ aResponseEvent.getArgs().setList("list", getService(). findApplicationDisciplinaryActions(aEvent)); }</pre>	false
<pre>catch(Exception ex) { aResponseEvent.getArgs().setString("aRe sponse", ex.getMessage()); }</pre>	Falla en el try por lo tanto va a la sección de código del catch.
Resultado	El método lanza una Excepción en forma de mensaje "No se encontraron parámetros verifique el periodo"

Nota: un código es eficiente cuando sus caminos básicos >= que la complejidad ciclomática

```
public void processEvent(AddControlCriminalActsEvent aEvent,
    C2SResponseEvent aResponseEvent) throws Exception {
```

1	<pre>try{ getService().addControlCriminalActs(aEvent); aResponseEvent.getArgs().setString("aResponse", "Se ha insertado correctamente"); }</pre>
2	<pre>catch (Exception ex) { aResponseEvent.getArgs().setString("aResponse", ex.getMessage()); }</pre>
3	<pre>}</pre>



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

$$CC = 3 - 3 + 2$$

$$CC = 2$$

Caminos Básicos

C1 → 1 – 3 C2 → 1 – 2 – 3

Caso de Prueba para C1

<pre>try{ getService().addControlCriminalActs(aEvent); aResponseEvent.getArgs().setString("aResp</pre>	<pre>true</pre>
--	-----------------

<pre>onse", "Se ha insertado correctamente"); }</pre>	
Resultado	El método ha recorrido el camino adecuado insertando en la base de datos y mostrando el mensaje "Se ha insertado correctamente"

Caso de Prueba para C2

<pre>try{ getService().addControlCriminalActs(aEvent); aResponseEvent.getArgs().setString("aResponse", "Se ha insertado correctamente"); }</pre>	false
<pre>catch(Exception ex) { aResponseEvent.getArgs().setString("aResponse", ex.getMessage()); }</pre>	Falla en el try por lo tanto va a la sección de código del catch.
Resultado	El método lanza una Excepción en forma de mensaje "Ya existe la fecha general, verifique los datos"

Nota: un código es eficiente cuando sus caminos básicos \geq que la complejidad ciclomática.

3.5. Resultado de las pruebas

En una primera iteración se realizaron 35 pruebas en total, de ellas el 100% arrojaron resultados satisfactorios con el objetivo de encontrar la mayor cantidad de errores existentes y darle solución a las mismas. Con el desarrollo de estas pruebas fueron encontradas 5 no conformidades que se subsanaron en el período correspondiente. Luego se realizó la segunda iteración donde se comprobó que todos los errores de la iteración anterior habían sido corregidos.

3.6. Aporte Social y Económico

- Permite la informatización de la Dirección de Auditoría de la Administración Provincial de Artemisa, ya que le brindará una herramienta eficiente que se ajusta a sus necesidades.
- No supone de grandes gastos económico, de tiempo, ni de material de oficina.
- La aplicación está desarrollada completamente en *software* libre, lo que permite mayor libertad a la hora de su distribución y desarrollo.
- Facilita y contribuye a mejorar la calidad del trabajo en la Dirección de Auditoría de la Administración Provincial de Artemisa.

3.7. Conclusiones del capítulo

Mediante la automatización de las pruebas funcionales que fueron realizadas al sistema se realiza de forma ágil y dinámica el proceso de desarrollo, influyendo en una óptima corrección de errores. Se validaron todos los caminos lógicos del código escrito con el margen de error mínimo en cuanto a las pruebas realizadas.

Conclusiones

Mediante el avance acelerado de la ciencia y las tecnologías, y su aplicación en todos los ámbitos del desarrollo de la humanidad se hace posible el uso de sistemas informáticos como los sistemas de gestión de información es el caso del sistema que se logró finalizar en la presente investigación. Durante el desarrollo de la misma se han ejecutado una serie de pasos o algoritmos lógicos que tributan a la obtención del resultado esperado por el cliente.

Con el uso de las herramientas, tecnologías, metodologías de desarrollo y lenguajes de programación libres que le dieron soporte a todo el proceso de desarrollo. Obteniéndose además un producto completamente libre o *freeware*, a través de un ciclo de desarrollo ágil, permitiendo enfrentarse a constantes cambios en los requisitos. El diseño y ejecución de un conjunto de pruebas funcionales realizadas al culminar cada nueva funcionalidad aseguraron la calidad y la garantía de funcionamiento del producto o *software* final

El estudio realizado en la Dirección de Auditoría de la Administración Provincial de Artemisa, sobre el proceso de gestión de la información, ha permitido llegar a las siguientes conclusiones, una vez cumplidos los objetivos y tareas trazadas:

- Se establecieron de los fundamentos teórico-metodológicos para el desarrollo de los procesos de gestión de información.
- Se establecieron de los fundamentos que deben sostener los procesos de gestión de información en la Dirección de Auditoría de la APA.
- Se desarrolló el Servidor para la Dirección de Auditoría basado en sus procesos de gestión de información.
- Se validó la contribución lograda a través del cumplimiento de los requisitos propuestos, mediante el uso de técnicas con esta finalidad.

Recomendaciones

Los conocimientos técnicos y prácticos adquiridos para llevar a su fin este proyecto, la experiencia acumulada en el diseño del módulo, así como consultas a profesores y personal de la Dirección de Auditoría de la Administración Provincial de Artemisa permiten proponer algunas recomendaciones:

- Continuar perfeccionando el módulo para la dirección de Auditoría de la Administración Provincial de Artemisa.
- Que se tome el contenido de esta investigación como referencia para próximos sistemas de gestión de información a desarrollar.

Referencias bibliográficas

ANDRADE, SIMÓN. Diccionario de Economía, Tercera Edición, Editorial Andrade, página 253 [Citado: 2005.].

ELLMAN, ENRIQUE. Confiabilidad. Una Estrategia de Negocios diferente. [Citado: 2008] <http://freedownload.is/pdf/enrique-ellmann-ellmann-sueiro-y-asociados-18796053.html>

GÓMEZ, A. M. eumed.net. [En Línea]. 04,2010. [Citado: 18 01, 2012.] <http://www.eumed.net/rev/cccss/08/bgsv.htm>

SALAZAR, A. A. Gestión del Conocimiento.com. [En Línea] [Citado: 11 01, 2012.] <http://www.gestiondelconocimiento.com/documentos2/apavez/gdc.htm>

QUIROGA, LOURDES A. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. [Citado 19 Diciembre 2011] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm

GOÑI, IVIS CAMEJO. Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias, http://bvs.sld.cu/revistas/aci/vol8_3_00/aci05300.htm

DÍAZ, Y. DE L. C.; AMADOR. Diseño del sistema de gestión de información del Centro de Estudios de Medio Ambiente y Recursos Naturales (CEMARNA) de la Universidad de Pinar Del Río [Citado 4, 2007] <http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-de-informacion-en-estudio-de-medio-ambiente.htm>

SIGOB. SIGOB Ecuador - Sistema de Información para la Gobernabilidad Democrática. [Citado: 2009] <http://www.sigob.gov.ec>

Referencias bibliográficas

FAO. WAICENT. Programas para la gestión de información, [Citado: 2012]
http://www.fao.org/waicent/portal/software_es.asp

DNP. Dirección Nacional de Presupuestos perteneciente a Colombia, [Citado: 2012]
<http://www.dnp.gov.co/>

TORRES, P. L.; LÓPEZ. Metodologías Ágiles en el Desarrollo de *Software*. [Citado 11 12, 2003] <http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf>.

CARVAJAL, A. A. Lenguaje Unificado de Modelado. [Citado 11 4, 2008] <http://syacomputadores.googlepages.com/LenguajeUnificadodeModeladoUML.doc>

BARROS, OSCAR (1994) - "Reingeniería de Procesos de negocio", Editorial Dolmen, Chile, 1994.

Microsoft Corporation. MSDN. [En línea] [Citado 16 5, 2011]
[http://msdn.microsoft.com/es-es/library/ms182516\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms182516(v=vs.80).aspx).

ROJAS, JOHANNA Y BARRIOS, EMILIO. [En línea] Grupo ARQUIISOFT, 2007. [Citado 16 5, 2011]
<http://gemini.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node37.html>

Bibliografía

NÚÑEZ, I. La gestión de la información, el conocimiento, la inteligencia y el aprendizaje organizacional desde una perspectiva socio-psicológica. [En Línea]. [Consultado: 13 11, 2011.] www.bogliolo.eci.ufmg.br.

La sociedad de la información y su influencia en la gestión en unidades de información [en línea]. Biblioteca Provincial “Rubén Martínez Villena” [En Línea]. [Consultado: 16 11, 2011.] www.magon.cu/infociencia/.../sociedad20información501.pdf.

PEÑALVER, GLADYS. “MA-GMPR-UR2 Metodología ágil para proyectos de *software* libre. “ Universidad de las Ciencias Informáticas. Ciudad de La Habana, Cuba: 2008. http://bibliodoc.uci.cu/TD/TD_1309_08.pdf. 94

.ARTILES, S.M. La gestión documental, de información y el conocimiento en la empresa, el caso de Cuba. [En Línea]. Mayo 2009. [Consultado: 12 11, 2011.] http://scielo.sld.cu/scielo.php?pid=S102494352009000500002&script=sci_abstract.

Glosario de términos

C

Código abierto: en inglés; *Open Source*, es el término con el que se conoce al *software* distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código

E

Entorno de Desarrollo Integrado llamado también **IDE** (sigla en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios.

F

Framework estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto

H

HTML, siglas de *HyperText Markup Language* (lenguaje de marcado de hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

L

Lenguaje de servidor: son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

U

UML 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.

X

XML son las siglas de *Extensible Markup Language*, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML, diseñado especialmente para los documentos de la web.

XHTML, siglas del inglés *eXtensible HyperText Markup Language*. XHTML es básicamente HTML expresado como XML válido. 15

W

W3C: El *World Wide Web Consortium*, abreviado **W3C**, es un consorcio internacional que produce recomendaciones para la World Wide Web (WWW).

World Wide Web (WWW) o **Red informática mundial** es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.

Anexos

Anexo 1. Modelos mediante los cuales se realizo el análisis documental.

No.	Denominación de los cargos	En el nivel central		En las Unidades Subordinadas			Totales			Total	
		Plantilla Aprobada	Completamiento		Plantilla Aprobada	Completamiento		Plantilla Aprobada	Completamiento		Inscriptos en Registro
			Al inicio del Año	Al final del Año		Al inicio del Año	Al final del Año		Al inicio del Año	Al final del Año	
UCAI del CAP Artemisa											
1	Directivos	8	8	7	1	1	1	9			
2	Audidores con formación Superior		9	7							
3	Audidores con formación media		16	16							
4	Audidores con formación emergente		8	8							
Totales		41	41	38							

Figura 5. Modelo 002.

INFORME SOBRE LA APLICACIÓN DE LAS MEDIDAS DISCIPLINARIAS																			
Entidad: UCAI del CAP Artemisa										fecha: I Trimestre 2011									
No.	Entidad Controlada	Medidas Disciplinarias Aplicadas																	
		Cuadros				Dirigentes y Funcionarios				Trabajadores									
		Amonestación	Democión Temporal	Democión Definitiva	Separación Definitiva	Totales	Amonestación	Democión Temporal	Democión Definitiva	Separación Definitiva	Totales	Amonestación	Multas	Inhabilitación	Suspensión del Derecho	Suspensión Vínculo Laboral	Traslado	Separación Definitiva	Totales
Auditorias																			
1	Dir. Mcpal de Trabajo Seg. Social SAB				1	1				1	1		1					4	5
2	Dir. Mcpal de Trabajo Seg. Social Caimito											1							1
3	Dir. Mcpal de Trabajo Seg. Social Candelana											1							1
4	Ofic. de Ctról y Cobros Multas Bahía Honda																		
5	Dirección de Deporte San Cristóbal																		
6	Dirección de Deporte Candelana																		
7	U/p Salud Pública Guira de Melena																		
8	Dir. Mcpal de Trabajo Seg. Social Artemisa																		
9	U/p Salud Pública S.A. Baños.																		

Figura 6. Modelo 003.

Anexo 2. Encuesta aplicada en la dirección de Auditoría de la Administración Provincial de Artemisa.

- ¿Cuáles son los objetivos de su organización? ¿Qué se está tratando de resolver?

- ¿Cómo obtiene habitualmente los datos? ¿Cuál es el proceso de obtención de los mismos? ¿Qué hace con la información una vez que la obtiene?

- ¿Cuál es la información o los tipos de reportes que actualmente se obtienen?

- ¿Con qué frecuencia se obtiene información?

- ¿Qué cantidad de información se maneja en general?

- ¿Cómo le gustaría que se le presentaran los datos en la pantalla? (gráficos, tablas, indicadores en la parte superior).

- Describa sus fuentes de datos (u otra clase de dimensión).