



Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Título: Sistema Informativo de la Administración Provincial de Artemisa (SIGOB).

Autor(a): Guelmis L. Martínez Ramírez

Tutor(a): Ing. Dania Fernández Aguilar

Co-Tutor(a): Msc. Isleny Orta Rodríguez

Artemisa, Julio 2012



Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Guelmis Lázara Martínez Ramírez
Autor(a)

Ing. Dania Fernández Aguilar
Tutor(a)

Msc. Isleny Orta Rodríguez
Co-Tutor(a)

Muchas personas especiales me han ayudado a llegar a este momento decisivo de mi vida y a ellos les agradezco por todo el cariño y la dedicación.

No puedo dejar de mencionar en primer lugar a mi familia, en especial a mi “Madre” y a “Pipo”, sin ustedes no habría llegado tan lejos, gracias por todo el apoyo emocional que me han brindado haciéndome saber en todo momento que están ahí para mí, aun cuando las cosas se ponían difíciles pues tenía nervios por algún examen o simplemente extrañaba mi hogar. A ustedes, mi familia les agradezco por estar siempre motivándome a llegar más lejos, siempre haciéndome ver que puedo lograrlo.

A mi abuelita “Polanda” porque tu espíritu siempre estuvo a mi lado, porque eres a quien siempre le pedí serenidad y tranquilidad antes de realizar un examen y siempre supe que no estaba sola porque tú estabas conmigo.

Gracias a “Edel” por las horas compartidas donde apenas dormíamos por estar trabajando, por su enorme paciencia, apoyo incondicional, la capacidad de demostrarme que todo tiene solución y que no debía desesperarme. Te estaré eternamente agradecida porque gracias a ti he llegado a este momento.

Gracias a “mis amigos”, los que me quieren de verdad como yo les digo, a Jeyt, Vity, Dariel, AlexK@ge, Koly, Orlando, Mayra, Mimi, a Marcos y Osvaldo por las consulticas de 10 minutos. A todos, gracias por la ayuda, por ese tiempo donde aclarábamos dudas y sabía que podía contar con ustedes ante cualquier inconveniente.

Agradezco también a mi tutora “Dania” y a mi co-tutora “Isleny” por la ayuda invaluable prestada en todo el proceso de desarrollo de tesis, sin ustedes no hubiera podido llegar hasta aquí.

En fin, a todos los que de una forma u otra han contribuido a mi proceso de formación como futura Ingeniera de las Ciencias Informáticas les doy mis más sinceros agradecimientos.

Le dedico este trabajo, el que define mi futuro, a algunas de las personas que considero especiales en mi vida:

A ti, mi “Mami Linda” porque te has sacrificado toda tu vida para hacerme llegar hasta aquí. Esto es para ti mamá, para que veas que todo tu amor, esfuerzo y dedicación no han sido en vano, porque tu hija será la profesional que soñaste y espero que estés siempre orgullosa de mí porque gracias a ti soy la mujer que siempre quisiste ser. Vivirás tus sueños a través de mí, este título es tuyo.

A mi “Nanita” mi hermanita hermosa, te dedico este trabajo para que veas que todo lo que te propongas en la vida lo podrás lograr y para que sigas diciendo a todos que quieres ser como yo cuando crezcas, siempre trataré de ser tu ejemplo a seguir, un día quiero que me dediques a mí tu trabajo de diploma.

En fin, a mi familia, a todos le dedico este, mi más grande logro hasta ahora.

El presente estudio fue realizado con los proyectos desarrollados actualmente en la Facultad Regional “Mártires de Artemisa” en el período 2011-2012 con el objetivo de desplegar un sistema informático de gestión de la información que contribuya a elevar los grados de centralidad, seguridad e integridad de la información en el proceso de gestión de la misma en la Administración Provincial de Artemisa (AP).

Para guiar la implementación del sistema se empleó la metodología SXP apoyada en el Lenguaje de Modelado Unificado (UML) además Java y JavaScript como lenguajes de programación.

Como propuesta de solución se desarrolla un Sistema Informativo para la Administración Provincial de Artemisa (SIGOB) compuesto por las siguientes herramientas: como marcos de trabajo para la implementación de la solución a JWebSocket, JavascriptMVC, JQuery, JUnit; como herramienta CASE a Visual Paradigm; como sistema gestor de base de datos PostgreSQL; como software de gestión de proyectos a Maven; como cliente para el control de versiones a RapidSVN, como IDE de desarrollo a NetBeans.

Finalmente se valida la propuesta de solución haciendo uso de las técnicas de prueba, seleccionando como pruebas de caja blanca a las pruebas unitarias y como pruebas de caja negra las pruebas funcionales obteniendo resultados satisfactorios que responden a las especificaciones del cliente.

Palabras Clave: Centralidad, gestión de la información, integridad, seguridad, sistema informático.

INTRODUCCIÓN	- 1 -
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	- 9 -
1.1. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	- 9 -
1.2. ESTADO DEL ARTE DE SOLUCIONES SIMILARES.....	- 11 -
1.2.1. <i>Sistemas Informativos Gubernamentales en el mundo</i>	- 12 -
1.2.2. <i>Sistemas Informativos Gubernamentales en Cuba</i>	- 13 -
1.3. ESTADO DEL ARTE DE METODOLOGÍAS.....	- 14 -
1.3.1. <i>Metodologías Pesadas: Proceso Unificado de Desarrollo (RUP)</i>	- 15 -
1.3.2. <i>Metodologías Ágiles: SXP</i>	- 16 -
1.4. ESTADO DEL ARTE DE HERRAMIENTAS, LENGUAJES Y TECNOLOGÍAS A UTILIZAR.....	- 17 -
1.4.1. <i>Lenguajes de Programación y Modelado</i>	- 17 -
1.4.2. <i>Otros Lenguajes</i>	- 21 -
1.4.3. <i>Tecnologías usadas para el desarrollo de la solución</i>	- 22 -
1.4.4. <i>Herramientas a emplear para el desarrollo de la solución</i>	- 28 -
CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.....	- 37 -
2.1. CONCEPCIÓN INICIAL	- 37 -
2.2. LEVANTAMIENTO DE REQUISITOS	- 37 -
2.3. PLANIFICACIÓN DEL PROYECTO POR ROLES.....	- 38 -
2.4. DESCRIPCIÓN DE LA ARQUITECTURA.....	- 39 -
2.5. MODELO DE DOMINIO	- 41 -
2.6. HISTORIAS DE USUARIO Y TAREAS DE INGENIERÍA.....	- 42 -
2.7. PLAN DE RELEASES.....	- 45 -
2.8. PROPUESTA DE SOLUCIÓN.....	- 46 -
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA	- 47 -
3.1. DIAGRAMA DE DESPLIEGUE.....	- 47 -
3.2. ESTÁNDAR DE CÓDIGO	- 48 -
3.3. ESTRUCTURA DE LA IMPLEMENTACIÓN.....	- 48 -
3.3.1. <i>En el lado del Servidor</i>	- 48 -
3.3.2. <i>En el lado del Cliente</i>	- 52 -
3.4. DISEÑO DE CASOS DE PRUEBA	- 53 -
3.5. PRUEBAS DE CAJA BLANCA REALIZADAS AL SIGOB	- 54 -
3.6. PRUEBAS DE CAJA NEGRA REALIZADAS AL SIGOB	- 56 -
<i>Casos de prueba de Aceptación</i>	- 56 -
3.6.1. <i>Casos de Prueba para la HU Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).</i>	- 56 -
3.6.2. <i>Casos de Prueba para la HU Gestionar la información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).</i>	- 61 -
3.7. RESULTADOS OBTENIDOS	- 64 -
3.8. FUNCIONALIDADES OBTENIDAS	- 64 -
3.9. APORTE SOCIAL Y ECONÓMICO.....	- 65 -
CONCLUSIONES GENERALES.....	- 67 -
RECOMENDACIONES.....	- 68 -

REFERENCIAS BIBLIOGRÁFICAS	- 69 -
BIBLIOGRAFÍA	- 71 -

Introducción

La información es uno de los conceptos más conocidos a nivel mundial debido a la gran cantidad de significados relacionados a la misma. Inicialmente se asociaba al incremento de conocimiento acerca de algo o a la incertidumbre. Su uso racional es la base del conocimiento humano y de los modelos del pensamiento.

Actualmente uno de los recursos más importantes que posee una empresa u organización es la información, puesto que incrementa las posibilidades de negocios al ser capaz de determinar de cierto modo el éxito o fracaso de los mismos. Para lograr maximizar la utilidad que posee la misma debe ser manejada de forma correcta y eficiente a través del uso de las Tecnologías de la Información y las Comunicaciones (TIC).

Las Tecnologías de la Información o Tecnologías de la Información y de la Comunicación (TIC) posibilitan dar un tratamiento específico a la información y garantizan el acceso a la misma, constituyendo un soporte y un canal que permite dar forma, almacenar, registrar y hasta distribuir contenidos relacionados con la misma.

Estas se han convertido en una herramienta esencial para el funcionamiento de nuestra sociedad, dígame en los sistemas políticos, instituciones económicas y hasta en nuestra vida cotidiana. Entre las ventajas que nos ofrecen las TIC se destaca el hecho de que permiten sentar las bases necesarias para lograr reducir la Brecha Digital sobre la que se debe construir una sociedad de la información y una economía del conocimiento actualmente.

Estas ventajas inciden en la apropiación de las innovaciones tecnológicas por parte de las empresas demostrando que los procesos de innovación tecnológica pueden ser entendidos como un proceso de innovación social que puede movilizar las capacidades de una organización determinada, haciendo que en diferentes áreas

de una empresa se desate un proceso dinámico, continuo y acumulativo que finalmente sea capaz de modificar y reelaborar las competencias organizativas de la misma.

Para ello, en gran parte de las empresas se utilizan diferentes sistemas informáticos o **Sistemas de Información (SI)** desarrollados para optimizar la gestión empresarial en diversos campos. Estos sistemas básicamente trabajan en un conjunto de datos estructurados de una organización en un entorno usuario-computadora, permitiendo recopilar, procesar y distribuir la información necesaria para lograr la operabilidad de una organización y las actividades propias de la misma, puesto que con ellos se puede gestionar todo tipo de recursos de información.

La existencia de una situación activa de cooperación entre las TIC, el SI y la estrategia corporativa posibilita a una organización cualquiera ser capaz de identificar las acciones a tomar dentro de la planificación estratégica del SI que la organización debe incorporar a su funcionamiento para mejorar su posición competitiva y su capacidad organizativa. Trabajar los SI desde esta perspectiva, permite tener el poder de los datos de una organización y así poder incidir sobre ellos en la gestión oportuna de la información y el conocimiento organizacional para la toma oportuna de decisiones institucionales.

Múltiples centros de desarrollo y universidades a lo largo del mundo se han sumado al desarrollo de SI que gestionen los procesos de una empresa u organización y en Cuba una de las universidades que más se destaca en esta esfera es la **Universidad de las Ciencias Informáticas (UCI)**.

La UCI es un centro de estudios universitarios que radica en Ciudad de la Habana, Cuba, este centro nació como como un proyecto de la Revolución Cubana y en sus inicios fue llamado "Proyecto Futuro". Conocida como universidad de nuevo tipo se creó básicamente con dos objetivos: informatizar el país y desarrollar la industria

del software para contribuir al desarrollo económico del mismo. En el año 2007 este proyecto se expandió con la creación de tres nuevas facultades que estarían ubicadas fuera de la sede central y serían conocidas como facultades regionales ubicadas en las provincias Artemisa, Ciego de Ávila y Granma.

En la actualidad esta universidad es uno de los centros de desarrollo más poderosos de Latinoamérica debido a que participa y coordina múltiples programas de informatización con ministerios o entidades nacionales e internacionales, participación a la cual se ha integrado la Facultad Regional "Mártires de Artemisa".

Considerar a la información como la base del conocimiento, la inteligencia y el talento organizacional, y como una de las principales fuentes de ventaja competitiva sostenible, permite colocar a una organización en condiciones emprendedoras ante el aprendizaje y la innovación. Es debido a esta consideración que Cuba se ha sumado a la tendencia de informatizar los procesos de sus principales instituciones y organizaciones en cada una de sus provincias tratando de remediar problemas de carácter organizativo.

Una de las provincias que presentaba problemas debido a la gran extensión territorial con que contaba era la provincia La Habana, pues el funcionamiento de los organismos de dirección provincial no era óptimo. Por este motivo se toma como decisión dividir la provincia de La Habana en dos nuevas provincias que llevarían por nombre Artemisa y Mayabeque con el objetivo de lograr un perfeccionamiento estructural y funcional de la administración y del Gobierno de las mismas.

Este trabajo de diploma está enfocado en la naciente provincia de Artemisa que luego del año 2011 cuenta con una estructura, una organización y composición de los órganos de gobierno que le pertenecen con el propósito de lograr de este modo un trabajo mucho más eficiente del Estado a través de la Administración Provincial (AP).

El Gobierno de la nueva provincia Artemisa es el responsable de preparar y proponer la política integral del Estado en el Consejo de la Administración Pública, esta política deberá ser aprobada y una vez hecho esto la Administración Provincial es la responsable de dirigir y ejecutar, así como coordinar y controlar su cumplimiento sobre la base de las estrategias de desarrollo que se proponga el país.

Para realizar estas tareas cuenta con 32 direcciones principales abarcando la esfera social, económica, cultural y deportiva que poseen misiones y funciones específicas en dependencia del área que abarquen las mismas almacenando información relevante del funcionamiento de la provincia.

Las direcciones de la AP gestionan su información de forma independiente, provocando que la misma se maneje en el Centro de Información sin la integridad y la organización requerida, imposibilitando el orden, la calidad y la claridad de la información y afectando a su vez la toma de decisiones en tiempo oportuno; provocando así pérdida de la misma por falta de seguridad y control de los datos que se manejan por las diferentes direcciones que componen la AP, ya que el formato en el cual se hace entrega de la información o se envía no es el idóneo.

El manejo de los datos se realiza de forma manual, provocando inconsistencia en el trabajo, así como ineficiencia y pérdida de tiempo. Todo esto trae consigo la dificultad en los reportes inmediatos, así como la búsqueda ante grandes volúmenes de información relacionada con el trabajo de las direcciones realizándose en múltiples ocasiones duplicado de la información y pérdida de los datos provenientes de cada dirección o fuente de información que brinda la ejecución de los procesos.

Se evidencia la falta de un control de acceso a los datos adecuado a la necesidad de proteger la información confidencial, debido a que múltiples fuentes pueden acceder a la misma posibilitando de este modo causar daños a la integridad y a la

seguridad de los datos, es posible, entonces, advertir la existencia de una **situación problemática**.

Por lo tanto se define como **problema de investigación** ¿cómo perfeccionar el proceso de gestión de la información de las direcciones de la Administración Provincial de Artemisa, que permita elevar los grados de centralidad, integridad y seguridad de los datos manejados en el mismo?

El **objeto de estudio** de este trabajo de diploma es el proceso de gestión de la información, identificándose como **campo de acción** los sistemas informativos gubernamentales.

Para resolver dicho problema se traza como **objetivo general**: Desarrollar un sistema informático de gestión de la información que contribuya a elevar los grados de centralidad, seguridad e integridad de la información en el proceso de gestión de la misma en la Administración Provincial de Artemisa.

Para contribuir al correcto desarrollo del objetivo general se trazaron los siguientes **objetivos específicos**:

1. Elaborar la Fundamentación Teórica de la investigación.
2. Realizar el análisis y el diseño de la solución propuesta para el Sistema Informativo de la AP.
3. Implementar la solución propuesta para el Sistema Informativo de la AP.
4. Validar mediante pruebas funcionales los resultados obtenidos con la solución.

Luego de trazarse los objetivos específicos se plantea como **hipótesis** la siguiente: Si se introduce en el proceso de gestión de la información de la Administración Provincial de Artemisa el sistema informático entonces se contribuirá a elevar los grados de centralidad, integridad y seguridad de la información.

Para darle cumplimiento a los objetivos específicos propuestos anteriormente se propone cumplir con las siguientes **Tareas de la Investigación**:

1. Definición del marco teórico, metodología, herramientas y tecnologías de la investigación.
2. Análisis y diseño de la solución de acuerdo a los modelos realizados y los requerimientos identificados.
3. Definición e implementación de las vistas y los controladores que tributen a los requerimientos funcionales de la solución de software propuesta.
4. Definición e implementación de los eventos (metainformación) y de Plugins en el servidor que dan soporte para atender los requerimientos de la aplicación para la gestión de la información del Sistema Informativo de la AP.
5. Ejecución de las técnicas de prueba para asegurar la calidad del resultado.

Para la realización de este trabajo de diploma se hace uso de los **métodos teóricos y empíricos**.

Como métodos teóricos se utilizaron:

- Análisis histórico - lógico: Se evidencia en el análisis de cómo los sistemas informativos han venido comportándose históricamente, sus características, objetivos, importancia, entre otros aspectos. Además de usarlo para investigar sobre las aplicaciones informáticas de este tipo implementadas en Cuba y el resto del mundo.
- Analítico - sintético: Se emplea en la definición del marco teórico, en la elaboración del estado del arte y en el análisis y síntesis de la documentación utilizada para la bibliografía.

- **Modelación:** Este método se emplea para la realización de diagramas teniendo en cuenta la metodología de desarrollo de software que se seleccionó.
- **Enfoque de Sistemas:** Este enfoque se emplea como parte de la metodología sistémica para estudiar, en forma colegiada, todos los elementos del sistema a desarrollar y sus interrelaciones, para la construcción de modelos y la elaboración de simulaciones con ellos permitiendo así tomar las decisiones óptimas para mejorar los escenarios futuros del Sistema Informativo de la AP.

Como métodos científicos empíricos se emplearon:

- **Observación:** Se emplea para identificar algunas características en el proceso de gestión de información como la forma de realización, quienes intervienen, que utilizan y se utiliza para la recopilación de la información necesaria acerca de los procesos en que se gestiona información en la AP. Ver Anexo #1
- **Entrevista:** Se realizaron entrevistas a especialistas de las direcciones de la AP para identificar subprocesos automatizables, y conocer el negocio. También se utiliza este método para definir el alcance de esta investigación y tener claro el dominio del problema.
- **Análisis de documentos:** Se emplea fundamentalmente para el análisis de documentos ya existentes que contengan información de la AP.

Como **Población** se identifica a todos los trabajadores de la AP, como **muestra** a los 2 administradores de red, el director de la Dirección de Organización, Planificación e Información (DOPI) y el Director del Centro de Información que representan un **12%** del total, con el uso del **muestreo intencional**.

Dada la hipótesis se logra identificar como:

Variable Independiente: Sistema Informativo de la Administración Provincial de Artemisa (SIGOB).

Variable Dependiente: Grados de centralidad, seguridad e integridad de la información en el proceso de gestión de la misma en la Administración Provincial de Artemisa.

Una vez llevadas a cabo las tareas investigativas se espera como **aporte práctico** el Sistema Informativo de la Administración Provincial de Artemisa (SIGOB).

Estructura del Documento:

El trabajo de diploma está estructurado en tres capítulos:

En el **Capítulo 1** Fundamentación Teórica: se hace un análisis del estado del arte del objeto de estudio, se investiga acerca de los sistemas informáticos vinculados al campo de acción, se fundamentan las metodologías, tecnologías y herramientas utilizadas para el desarrollo del sistema de gestión.

En el **Capítulo 2** Características, Análisis y Diseño del Sistema: se define el negocio y se describe la solución propuesta para la situación problemática. Se presentan las características y funcionalidades del sistema a partir de los requisitos funcionales y no funcionales capturados. Se realiza el análisis y diseño del sistema.

En el **Capítulo 3** Implementación y Validación del Sistema: incluye la programación realizada a partir de los requerimientos y los diagramas del diseño elaborados, así como las métricas y pruebas utilizadas para la validación de la misma.

Capítulo 1. Fundamentación Teórica

Introducción

En el presente capítulo se brinda una descripción acerca de la gestión de la información en conjunto con los sistemas que gestionan la misma, además de una breve panorámica sobre las aplicaciones existentes a nivel internacional, nacional y de universidad que se relacionan con este trabajo de diploma. Se realiza un análisis de la necesidad de crear un sistema de gestión de información para de la AP además de fundamentar la utilización de la metodología seleccionada, las herramientas y arquitectura utilizadas para desarrollar la aplicación.

1.1. Conceptos asociados al dominio del problema

Existe una gran diversidad de conceptos que pueden contribuir al mejor entendimiento del problema en cuestión pero en el presente trabajo de diploma se abordarán algunos de los más relevantes, tales como gestión de la información, sistema de información y sistema de gestión de información.

La **Gestión de información**: Conjunto de instancias responsables por la identificación de políticas y acciones en la organización con relación a lo siguiente:

- La forma en cómo la información se adquiere, registra y guarda.
- El cómo la información se usa y acumula.
- La manera como las personas manejan la información, aplican sus habilidades y cooperan entre ellas.
- La efectividad con que las actividades relacionadas con la información contribuyen al logro de los objetivos de las organizaciones y los individuos.
- La forma como se usan las tecnologías de la información en todas estas actividades.
- Los costos y beneficios que conllevan las actividades de información. (Clavero Quintana & Díaz Padilla, 2008)

En fin se puede afirmar que la gestión de la información es un proceso imprescindible en cualquier organización de este siglo XXI debido a que incluye operaciones básicas para el manejo de los datos, tales como: la extracción, manipulación, tratamiento, depuración, acceso y colaboración de la información adquirida, entre otros. O sea, actualmente una buena gestión de los datos de una institución determinada pudiera determinar el éxito o el fracaso de los procesos o negocios de la misma ya que mejora la toma de decisiones en todas las esferas.

Un **Sistema de Información** es un conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de dicha organización. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general, los cuales procesan la información y la distribuyen de manera adecuada, buscando satisfacer las necesidades de la organización. El objetivo primordial de un sistema de información es apoyar la toma de decisiones y controlar todo lo que en ella ocurre. (Definicion.de, 2008-2012)

En fin un sistema de información es aquel que permite agrupar determinados elementos dígame: personas, datos, actividades, etc. En este sistema dichos elementos realizan una interacción entre sí posibilitando de este modo el procesamiento de información y lograr una correcta distribución de la misma de modo que responda a los objetivos de la organización en que sea utilizado.

Una vez analizados los conceptos de gestión y sistema de información mencionados con anterioridad la autora de la presente tesis de grado ofrece la definición operacional de sistema de gestión de información que se muestra a continuación:

Un **Sistema de Gestión Información** no es más que un grupo de componentes o funciones que se relacionan entre sí para procesar, distribuir y almacenar los datos

de una organización en específico de modo que estos procesos contribuyan significativamente a mejorar la toma de decisiones y propiciar un mejor control de la información.

1.2. Estado del Arte de Soluciones Similares

La sociedad actual debe hacer frente a las necesidades de información, debido a que esta se ha convertido prácticamente en un elemento vital brindando un soporte de transmisión de conocimientos. Los principales intereses que se persiguen al poseerla son los de desarrollar necesidades a futuro; determinando de este modo que la misma no se encuentre aislada ya que siempre se encontrará ligada a un entorno social y cultural. Es por ello que la gestión de la información se centra en la tecnología para transmitirla.

En la actualidad las empresas e instituciones del mundo funcionan como una unidad que comparte una visión en específico. Esto engloba de por sí la información que se comparte, el trabajo en equipo y lograr un funcionamiento acorde con los principios establecidos por la misma. El uso de un sistema de gestión de la información permite lograr los objetivos de una organización determinada a través de la creación de una serie de estrategias, entre las que figuran la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

Los sistemas de gestión de información para la gestión del conocimiento o dígame también de la información son hoy imprescindibles en cada organización. Estos permiten operar casi todos los activos tangibles e intangibles de una institución determinada convirtiéndose en una herramienta de gerencia cotizada debido a que pudiera garantizar alcanzar exitosamente los resultados propuestos por la organización.

1.2.1. Sistemas Informativos Gubernamentales en el mundo

Existen en el mundo múltiples sistemas informativos para gestionar los procesos de instituciones gubernamentales. A continuación se mencionarán algunos de estos sistemas.

En Ecuador la Subsecretaría de Informática presentó Sistemas Gubernamentales. Estos sistemas informáticos integrados en este Gobierno han agilizado la gestión de recursos financieros, humanos y materiales, buscando el ahorro de tiempo en los procesos administrativos.

Entre estos sistemas se destacan: Integrado Gubernamental, conocido como ERP o GRP; de Gestión Documental, denominado Quipux; y el Portal Único de Trámites Ciudadanos.

ERP: Permite unificar y simplificar los procesos administrativos de las instituciones públicas, y se adapta a la mayoría de entidades. El programa está elaborado en un 90% en Software Libre y permite innovaciones tecnológicas, como firma electrónica, chip inteligente, Sistema de Posicionamiento Global (GPS), entre otros. La idea es eliminar el papel, el documento y realizar la transacción a través de la red para contar con un archivo digital, ayudando a simplificar el trabajo administrativo, e incluso se puede contar con bases de datos para la elaboración de reportes gerenciales y de planificación.

Quipux: Es un sistema de autoservicio para los funcionarios y público en general. Se puede acceder a él con un usuario y contraseña de hasta ocho dígitos. El acceso es por Internet y el servicio es permanente durante los 365 días del año y las 24 horas del día. Con este sistema se puede contar con un archivo documental que permite organizar la correspondencia interna y externa. Un total de 26 instituciones públicas cuentan con este servicio en Ecuador.

Portal Único de Trámites Ciudadanos: Se conoció que este sitio, además de informar a la población de cómo realizar un trámite, sirve de nexo para facilitar la comunicación con la institución requerida. El portal registra al momento 539 trámites ciudadanos.

En México el Gobierno Federal y, en especial, la Secretaría de la Función Pública están realizando en materia de Tecnologías de Información y Comunicación (TICs) y gobierno electrónico un gran trabajo. Entre algunos de los sistemas de gestión gubernamental que están desarrollando se encuentran:

El nuevo Portal Ciudadano: El Gobierno Federal, a través de la Secretaría de la Función Pública, está trabajando en la construcción de un nuevo Portal Ciudadano. Se espera que dicho Portal se convierta en el principal buscador de información gubernamental en México y en la ventanilla virtual de servicios gubernamentales para el ciudadano. El nuevo Portal será la base para la construcción de una nube de servicios gubernamentales con la que: se detonará la automatización de trámites y servicios, se mejorará la estructura, significado y relevancia de la información de los Portales gubernamentales y se multiplicará el valor de las inversiones de TICs del Gobierno.

1.2.2. Sistemas Informativos Gubernamentales en Cuba

Cuba es uno de los países de Latinoamérica que se ha sumado a la tendencia de informatizar sus instituciones gubernamentales creando sistemas informativos que gestionen los principales procesos de las mismas. A continuación se mencionarán algunos de los sistemas existentes en el país.

Portal del Ciudadano: permite encontrar, de una forma cómoda y eficaz, toda la información que se necesita conocer para iniciar un Trámite que requiera realizar en las diferentes entidades de la administración pública y el gobierno.

Sistema de Información del Gobierno Centralizado: dependerá del presidente y sistematizará todos los datos relevantes para el funcionamiento del país. Todos los niveles de administración en los ámbitos económico, social, demográfico, geográfico, medioambiental, de funcionamiento de sus órganos y otros, tributarán con información a este nuevo sistema. El Presidente del Consejo de Ministros ejerce la dirección general del Sistema de Información del Gobierno, atribución que puede delegar en un Vicepresidente del Consejo de Ministros.

Una vez analizados algunos sistemas informativos gubernamentales existentes en el mundo y en Cuba se ha identificado que todos emplean el Protocolo de Transferencia de Hipertexto (HTTP) para enviar y recibir los datos, no existe una comunicación en tiempo real que permita una actualización constante de los cambios realizados además de que se sobrecarga el ancho de banda debido a las grandes cantidades de información enviadas y recibidas.

Debido a este análisis exhaustivo fue posible identificar que en la actualidad no existen sistemas informativos para la gestión de la información de los gobiernos provinciales en Cuba puesto que estos órganos de dirección solamente están establecidos en las provincias de Artemisa y Mayabeque que aún no han informatizado sus procesos.

1.3. Estado del Arte de Metodologías

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen a través de un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales centradas específicamente en el control del proceso. Estas han

demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos).

Sin embargo la experiencia ha demostrado que las metodologías tradicionales no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, porque no están pensadas para trabajar con incertidumbre.

Como respuesta a los problemas aplicando metodologías tradicionales surgieron otras metodologías que tratan de adaptarse a la realidad del desarrollo de software. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientadas al código: siguiendo un camino donde la parte importante de la documentación es el código fuente.

1.3.1. Metodologías Pesadas: Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

Características principales de RUP

- “Centrado en los modelos: Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.”
- “Guiado por los Casos de Uso: Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.”
- “Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.”

- “Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.” (ChristianMania, 2011)

1.3.2. Metodologías Ágiles: SXP

SXP está compuesta por las metodologías SCRUM y XP, ofreciendo una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva. Esta metodología fomenta el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SXP consta de 4 fases principales:

- **Planificación-Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** es la puesta en marcha.
- **Mantenimiento:** es la fase donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos con pequeños equipos de trabajo, un constante cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Fomenta el trabajo en equipo, con un objetivo claro, permitiendo el seguimiento y

control de las tareas a realizar. (SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE, 2010)

Fundamentos de la selección

Actualmente existen diversas tecnologías que ayudan a crear software, pero aún no se cuenta con una en específico que se ajuste sin importar el tamaño o las características del mismo. Es por ello que cuando se toma la decisión de construir un producto determinado se deben tener en cuenta las características, el nivel de complejidad y el tipo de contrato al que está sujeto el mismo. En el caso de este producto en particular se propone, debido a las ventajas que ofrece en proyectos de este tipo, el uso de metodologías ágiles y dentro de ellas en específico SXP.

1.4. Estado del Arte de Herramientas, Lenguajes y Tecnologías a utilizar

1.4.1. Lenguajes de Programación y Modelado

A nivel mundial se han ido desarrollando diversos lenguajes de programación y modelado que han elevado su nivel desde la programación estructurada hasta la orientada a objetos, etc. De este modo se evidencia la evolución del pensamiento humano al modelar la vida real de una forma abstracta y así poder gestionar de forma más eficiente la información de la que se dispone.



Lenguaje Modelado Unificado

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90's.

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. Una de las metas principales de UML es avanzar en el estado de la integración institucional proporcionando herramientas de

interoperabilidad para el modelado visual de objetos. Sin embargo para lograr un intercambio exitoso de modelos de información entre herramientas, se requirió definir a UML una semántica y una notación. (González Cornejo, 2009)

Se puede decir que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad y se centra en la representación gráfica de un sistema. Este lenguaje permite:

- **Visualizar:** permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflows) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.



Sun Microsystems desarrolló, en 1991, el lenguaje de programación orientado a objetos que se conoce como Java. La intención de Sun era crear un lenguaje con una estructura y una sintaxis similar a C y C++, aunque con un modelo de objetos más simple y eliminando las herramientas de bajo nivel.

Los pilares en los que se sustenta Java son cinco: la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en diversos sistemas operativos, la inclusión por defecto de soporte para trabajo en red, la opción de ejecutar del código en sistemas remotos de manera segura y la facilidad de uso. (Definicion.de, 2008-2011)

Es un lenguaje semicompilado que funciona con una máquina virtual, la cual se conoce como la Java Virtual Machine, o JVM. La máquina virtual de Java es necesaria para poder ejecutar los programas escritos en este lenguaje, y se encuentra disponible para diversos sistemas operativos, como son OS X, Windows, y diversas distribuciones de Linux. Esta máquina virtual lo que hace es interpretar los programas de Java, transformarlos a lenguaje máquina para la PC, y así la PC puede ejecutar el programa.

Java tiene muchas ventajas, entre estas tenemos que es un lenguaje multiplataforma, lo que quiere decir que se ejecuta en la mayoría de los sistemas operativos, inclusive en sistemas operativos móviles. Otra ventaja es que Java es un software de distribución libre, no es necesario pagar una licencia para poder comenzar a desarrollar en este lenguaje. Así mismo es un lenguaje muy completo y poderoso, se pueden realizar muchas tareas con él, pues posee una librería y utilidades muy completas que facilitan la programación. (codigoprogramacion.com, 2010-2012)

Java es:

- Basado en Clases. Los objetos son divididos en clases e instancias con toda la herencia a través de la jerarquía de clases. Las clases e instancias no pueden tener propiedades o métodos añadidos dinámicamente.
- Los tipos de datos variables deben ser declarados (definición estática de tipos).
- No puede escribir automáticamente en el disco duro. (© 2012 Mozilla Developer Network, 2012)



JavaScript

JavaScript es un lenguaje de script multiplataforma [cross-platform] orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

El núcleo de JavaScript contiene un conjunto central de objetos, tales como Array (arreglos), Date (fechas) y Math (objetos matemáticos), además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias. El núcleo de JavaScript puede ser extendido para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

- **JavaScript del lado Cliente:** extiende el núcleo del lenguaje proporcionando objetos para el control del navegador (Navigator o cualquier Web browser) y su Modelo Objeto Documento [Document Object Model] (DOM). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario HTML y responder a los eventos de usuario tales como los clics del mouse, entradas del formulario y navegación de páginas.
- **JavaScript del lado Servidor:** extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comuniqué con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor.

JavaScript comparado con Java es un lenguaje en un formato mucho más liberado. No tiene que declarar todas las variables, clases y métodos. No debe preocuparse

si sus métodos son públicos, privados o protegidos y no tiene que implementar sus interfaces. Los tipos de variables, parámetros y funciones de retorno no son explícitamente definidos.

1.4.2. Otros Lenguajes



HTML es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla de HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. EL HTML permite describir la estructura y el contenido en forma de texto, además de complementar el texto con objetos tales como imágenes. Este lenguaje se escribe mediante etiquetas, que aparecen especificadas por corchetes angulares (< y >).

Por otra parte, permite incluir scripts (por ejemplo, de Javascript), códigos que pueden modificar el comportamiento de los navegadores web y de otros procesadores de HTML. Los archivos de formato HTML utilizan la extensión .htm o .html. Entre los componentes del HTML, aparecen los elementos y sus atributos, los tipos de data y la declaración de tipo de documento. Los elementos son la estructura básica de este lenguaje, ya que tienen dos propiedades: atributos y contenido. (Definicion.de, 2008-2011)



Hojas de Estilo en Cascada - CSS (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Una vez creados los contenidos de una página web, se utiliza el

lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros. CSS es el lenguaje utilizado para controlar el aspecto de las interfaces o vistas de la solución propuesta. (Eguíliz Pérez, 2009)

Fundamentos de la Selección

Existen mundialmente múltiples lenguajes de programación y modelado que poseen características para la construcción de un sistema como el que se propone en el presente trabajo de diploma tales como PHP, ASP.NET, entre otros. Los lenguajes seleccionados son los mencionados anteriormente debido a que son los estipulados por la dirección de la FR Artemisa para el desarrollo del sistema y debido a las características que presentan.

1.4.3. Tecnologías usadas para el desarrollo de la solución

Marcos de trabajo

En términos generales, un marco de trabajo (framework) es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework es una 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. (Díaz, 2010)

Existen diversos frameworks con diferentes características, orientados a lenguajes en específico, es por ello que elegir uno puede resultar una tarea ardua. Para el

desarrollo de este trabajo de diploma se toma como base el concepto de framework definido por Ana Rosa Díaz.

A continuación se describirán las principales características de algunos marcos de trabajo conocidos a nivel mundial tanto para el lado del servidor como del cliente.

Marco de Trabajo del lado del Servidor



Marco de Trabajo jWebSocket

JWebSocket es un marco de trabajo de código abierto para el desarrollo de aplicaciones web estacionarias y móviles basado en Java en el lado del servidor y en JavaScript del lado del cliente. JWebSocket establece un modelo de token. Los tokens son datos abstractos que a través de una estructura jerárquica y una API proporcionan métodos de acceso a los contenidos. Con el objetivo de realizar una abstracción en la manipulación de los diferentes formatos, el marco de trabajo convierte los paquetes de datos entrantes y salientes en tokens. El cliente nativo soporta el intercambio de paquetes en los formatos JSON, XML y CSV, que en entornos específicos se pueden utilizar sin la necesidad de manejarlos a través de tokens. El cliente jWebSocket tiene una arquitectura de plugins que permite aumentar con facilidad sus funcionalidades.

El servidor jWebSocket está diseñado para funcionar como servidor de comunicaciones o como servidor web, brindando total flexibilidad. En la primera opción jWebSocket proporciona un archivo .jar, ofreciendo la ventaja de ejecutarse fácilmente desde una línea de comandos e integrarse a la biblioteca de una aplicación existente de Java. En la actualidad hay algunos servidores que ya soportan Websockets y otros que no, por lo que jWebSocket se integra a servidores como Tomcat o Apache para lograr una comunicación Websockets. En caso de que los servidores soporten de manera nativa Websockets, como el caso de Jetty o GlassFish, se incluyen las funciones de comunicación del marco de trabajo

jQuery, pero los motores internos se apagan y el anfitrión se utiliza. Esto asegura que no haya mecanismos de seguridad adicionales.

jQuery como servidor web proporciona un conjunto importantes funcionalidades y su arquitectura extensible mediante plugins permite añadir fácilmente características adicionales a un sistema independiente. Por otra parte los administradores pueden configurar el servidor exactamente como sea necesario y dejar a un lado todos los módulos que no necesiten. En un clúster los plugins se pueden utilizar como servicios, por lo que jQuery perfectamente es compatible con SOA (Service Oriented Architectures) en un entorno totalmente basado en eventos. Estas características muestran la fortaleza y flexibilidad del marco de trabajo para el desarrollo de aplicaciones web estacionarias y móviles, multiplataforma, multisectorial y compatible con todos los navegadores. (Framework Approach for WebSockets, 2011)

Fundamentos de la selección

El framework jQuery fue seleccionado para desarrollar esta aplicación porque es la tecnología implantada en la Facultad Regional de la UCI “Mártires de Artemisa”. Está basado en Java y utiliza el protocolo WebSocket, además en el centro existe un grupo de desarrollo de este marco de trabajo que provee soporte y ayuda en cuanto al mismo.

Marcos de Trabajo del lado del Cliente



jQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. (Murphey, 2010)

Consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

jQuery implementa una serie de clases que nos permiten programar sin preocuparnos del navegador con que trabaja el usuario, ya que funciona de la misma forma en todas las plataformas. Así pues, este marco de trabajo JavaScript, ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente.

Ventajas

- Como buen JavaScript se puede trabajar separado del html, haciéndolo más fácil de mantener y enriquecer, aumentando la productividad del proyecto.
- Se considera su sintaxis liviana, teniendo en cuenta que la sencillez y poca extensión de código es fundamental para los desarrollos.
- Funciona independientemente del navegador sobre el que se visualiza el sitio. Al servir para todos, el desarrollador tiene menos trabajo, menos código, menos espacio, menos problemas.
- Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del marco de trabajo.
- jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones.
- La comunidad de creadores de plugins o componentes es muy amplia, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar interfaces de usuario, galerías, votaciones, efectos diversos, entre otros. (Qué

es jQuery, para qué sirve y qué ventajas tiene el utilizar este framework Javascript, 2009) (jQuery, la librería Javascript por excelencia. Un framework Javascript lleno de ventajas, 2011)

jQuery es una herramienta imprescindible para desarrollar efectos dinámicos, Ajax, interacción, interfaces de usuario avanzadas, entre otros; sin tener que complicarse con los niveles más bajos del desarrollo, ya que muchas funcionalidades ya están implementadas, o bien las librerías del marco de trabajo te permitirán realizar la programación mucho más rápida y libre de errores. (Qué es jQuery, para qué sirve y qué ventajas tiene el utilizar este framework Javascript, 2009)



JavascriptMVC

Es una colección de las mejores prácticas y herramientas para crear aplicaciones JavaScript. Construido en la cima de jQuery, que consta de los componentes independientes siguientes:

- StealJS - La gestión de dependencias y una herramienta de construcción estúpidamente fácil.
- FuncUnit - No hay una solución mejor prueba funcional disponible. Escribir pruebas en la sintaxis de jQuery, que se ejecutan a través de línea de comandos o el navegador.
- jQueryMX - Estos plugins son los componentes básicos de las aplicaciones de gran jQuery: modelo, vista, controlador, de clase, accesorios y más.
- DocumentJS - Una herramienta compatible JSDoc que convierte a los comentarios en una aplicación de búsqueda de documentación. (Jupiter Consulting - JavaScriptMVC Training and Support, 2012)

JavaScriptMVC es un completo y robusto framework para crear aplicaciones en Javascript. Utiliza JQuery para las funcionalidades en Ajax y el manejo del DOM.

Más que una simple librería, es un conjunto de scripts en donde está contemplado el Testing y la documentación de la aplicación.

Este framework es algo ya más bien parecido a una plataforma para desarrollar y utiliza el patrón Model-View-Controller, lo que explica el nombre. Entre las razones de por qué usarlo, destacan: Refuerza las “Buenas Prácticas”, aumenta la mantenibilidad y reduce el tiempo de desarrollo. JavaScriptMVC suple las siguientes herramientas:

- Generador de Código.
- Manejador de Actualización y Dependencias.
- Suite de Pruebas.
- Comprensión.
- Documentación. (NubeHost.mx, 2011)

Fundamentos de la Selección

En el mundo existen muchos marcos de trabajo que permiten trabajar en el lado del cliente tales como: Moo Tools, EXTJS, Prototype, Yahoo User Interface (YUI), etc. Las tecnologías seleccionadas para utilizar en el lado del cliente son JavascriptMVC y JQuery debido a que son las estipuladas por la dirección de la Facultad Regional “Mártires de Artemisa” para realizar este proyecto, además de las ventajas que ofrecen las mismas mencionadas anteriormente.

Marcos de Trabajo para la realización de Pruebas Unitarias

JU JUnit

JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que como su nombre indica se encarga de crear pruebas unitarias de aplicaciones Java. Contiene un conjunto de clases (*framework*) que permiten realizar la ejecución de clases Java de manera controlada, para poder evaluar si el

funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

JUnit también permite controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el Nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación. (Sáez Bertrand, Rodríguez Acero, & Albert, 2009-2010)

Fundamentos de la Selección

Existen diversos frameworks que permiten realizar pruebas unitarias al código en dependencia del lenguaje que emplea la solución, entre ellos figuran para PHP: Symfony, Lime, SimpleTest, PHPUnit, etc.; para C/C++ CPPUNIT; para la plataforma .NET NUnit; para Microsoft Visual Fox Pro FoxUnit; para Python PyUnit; NUnit para C# y para Java JUnit. Debido a que el IDE de desarrollo seleccionado para el presente trabajo de diploma posee un plugin que soporta la integración con el framework JUnit, a que el lenguaje de programación en el lado del servidor es Java, y a que es la herramienta estipulada por la dirección de la Facultad Regional "Mártires de Artemisa" para la realización de pruebas unitarias, se elige a JUnit.

1.4.4. Herramientas a emplear para el desarrollo de la solución

Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) propician un conjunto de métodos y técnicas automatizadas que brindan ayuda y dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todo el ciclo de vida del desarrollo de un software, reduciendo el esfuerzo, el costo y el tiempo.

Dichas herramientas se encuentran en una continua evolución, por lo que existe una gran variedad de proveedores y productos, cada uno de ellos con diferentes aplicaciones y especificaciones. A continuación se hace una caracterización de algunas de ellas que permitirá adquirir los elementos necesarios para determinar cuál es la más idónea para especificar y diseñar la solución propuesta. (Herramientas Case, 2011)



Visual Paradigm es una poderosa herramienta CASE que hace uso del lenguaje modelado unificado (UML) con soporte multiplataforma, que proporciona un ciclo de vida completo del desarrollo de software, excelentes facilidades de interoperabilidad con otras aplicaciones, compatibilidad entre versiones, así como dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Dicha herramienta brinda una serie de facilidades que se mencionan a continuación:

- Soporta un conjunto de estándares entre los que se encuentran UML, SysML, BPMN, XML y XMI.
- Soporte de modelado UML, modelado de procesos de negocios y un generador de mapeo de objetos-relacionales para los lenguajes de programación Java, .NET y PHP.
- Integración con herramientas Java (Eclipse/IBM WebSphere, JBuilder, NetBeans IDE, Oracle JDeveloper y BEA Weblogic).
- Permite la generación de código y la ingeniería inversa para un conjunto de lenguajes entre los que se encuentran Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. Cualquiera de los cambios en el código existente puede reflejarse en el modelo y viceversa.
- Ofrece herramientas para la generación de reportes en formatos html, pdf y doc.

- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Interoperabilidad e integración. Permite la integración con un conjunto de herramientas (Visio drawing, Rational Rose, Erwin Data Modeler Project, Microsoft Excel y Microsoft Word document) e intercambiar diagramas UML y modelos usando representaciones industriales comunes.
- Modelado de base de datos. Proporciona una mayor documentación de la base de datos y diagramas de mapeo de relación de objetos.
- Integración con herramientas para el control de versiones.
- Diseño de prototipo de Interfaz de Usuario. Permite insertar información adicional a los diagramas mediante notas y comentarios para describir sus elementos lo que facilita la revisión de los prototipos así como el trabajo en equipo. (Visual Paradigm, 2012)

Fundamentos de la selección

Existe una gran diversidad de herramientas CASE desarrolladas en todo el mundo, entre ellas además del Visual Paradigm se encuentran el Rational Rose, entre otras. Luego de analizar algunas de estas y analizar que el presente trabajo de diploma será desarrollado respetando las políticas de software libre se seleccionó a Visual Paradigm. Esta herramienta posee una licencia comercial que nuestra universidad posee aunque el software no es libre.

Herramientas de Control de Versiones

Un sistema de control de versiones (o sistema de control de revisiones) es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web. (producingoss.com)

Un sistema de control de versiones no es más que un sistema que permite gestionar archivos y directorios manteniendo un historial de las modificaciones que se han realizado sobre los mismos. Estos guardan toda la información que manejan

en un repositorio central al que se puede acceder mediante la red propiciando de este modo que los desarrolladores trabajen en un entorno colaborativo al mismo tiempo que provee mayor seguridad y disponibilidad de los datos.

Es por ello que al desarrollar un proyecto de esta magnitud es necesaria la utilización de un sistema para el control de versiones, algunos de estos son:



Subversion

Subversion es un sistema de control de versiones completamente equipado que fue originalmente diseñado para reemplazar a CVS. Desde entonces se ha expandido más allá de su objetivo original, pero su modelo básico, el diseño y la interfaz fueron fuertemente influenciados por CVS por lo que debido a estas particularidades los usuarios de CVS se sienten muy cómodos al interactuar con Subversion. (Apache Software Foundation, 2011)

Características Generales:

- Directorios son versionados.
- Permite resolver conflictos interactivamente.
- Permite gestionar eficazmente archivos binarios.
- Permite realizar bloqueo de archivos.
- Es posible vincularlo con diferentes lenguajes de programación.
- Permite vincular diferentes repositorios.
- Da soporte a desarrolladores.
- Permite desarrollo paralelo.

Fundamentos de la selección

Una vez analizados los sistemas de control de versiones existentes entre los cuales además de Subversion figura Git, entre otros, se procede a elegir a Subversion para ser empleado en el desarrollo de la aplicación debido a que permite integración con NetBeans y la posibilidad de mantener un desarrollo en paralelo. Otro de los motivos por los que se selecciona Subversion es por la experiencia de

desarrollo que poseen algunos miembros del equipo, además de que es fácil de usar.

Herramientas del Cliente de Control de Versiones

Los clientes de control de versiones permiten la gestión de cambios que se realizan sobre los elementos de algún producto. Permitiendo una conexión entre él como cliente y el servidor, para un mejor manejo de los archivos locales.



RapidSVN

Es una plataforma de interfaz gráfica de usuario, para el sistema de revisión de Subversion. Este proyecto también incluye un cliente de Subversion C + + API. RapidSVN está licenciado bajo la v3 de GNU General Public License. Utiliza las mejores características de los clientes de otras arquitecturas de control de versiones. Si bien es bastante fácil para los nuevos usuarios de Subversion trabajar con él, también debe ser lo suficientemente potente como para que los usuarios con experiencia sean aún más productivos. (RapidSVN, 2011)

Características generales:

- **Simple:** proporciona una interfaz fácil de usar para las características de Subversion.
- **Eficiente:** simple para los principiantes pero lo suficientemente flexible como para aumentar la productividad para los usuarios de Subversion con experiencia.
- **Portable:** se ejecuta en cualquier plataforma en la que Subversion y wxWidgets puede ejecutar: Linux, Windows, Mac OS / X, Solaris, etc.
- **Rápido:** completamente escrito en C + +.
- **Multilingüe:** que ha sido traducido a muchos idiomas ya: alemán, francés, italiano, portugués, ruso, ucraniano, chino simplificado, japonés.
- **Soporte completo** para Unicode.

Fundamentos de la selección

Luego de analizar las características de algunas herramientas existentes como el TortoiseSVN, el RapidSVN entre otras, se selecciona como herramienta para el control de versiones del presente trabajo de diploma a RapidSVN pues su licencia libre, fácil de usar y algunos miembros del equipo poseen experiencia trabajando con la misma.

Entorno integrado de desarrollo - IDE

Un IDE (acrónimo en inglés de Integrated Development Environment) es un entorno de programación que integra varias herramientas con el objetivo de facilitar el desarrollo de software sobre uno o varios lenguajes de programación. La mayoría de los IDEs cuentan con herramientas tales como: editor de código, herramientas para el rastreo de código, compilador, depurador y constructor de interfaz gráfica (Nourie, 2005).



NetBeans

NetBeans IDE es una herramienta desarrollada por Sun Microsystems. Está completamente escrito en Java, por lo que puede ser utilizado desde cualquier sistema operativo compatible con la máquina virtual de Java. Permite el desarrollo de aplicaciones de escritorio, web y móviles. Brinda soporte a varios lenguajes de programación como Java, PHP, C/C++, Groovy, Python, JavaScript, entre otros. Es un producto libre y gratuito sin restricciones de uso. Este entorno integrado de desarrollo es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Su misión consiste en evitar tareas repetitivas, facilitar la escritura correcta de código, disminuir el tiempo de depuración e incrementar la productividad del desarrollador. Cuenta con un depurador, perfilador de integración, herramientas para refactorizaciones, completamiento de código y control de versiones de archivos. (NetBeans, 2011).

Fundamentos de la selección

Aunque existen otros IDEs en el mundo como Eclipse, etc., el IDE seleccionado para el desarrollo de la solución propuesta es NetBeans por las ventajas mencionadas anteriormente, además de que es una herramienta con la que el equipo de desarrollo está familiarizado.

Además su versión 7.0.1 introduce un soporte para el desarrollo con la especificación JavaSE7 (Java Standard Edition) con las características de JDK7 (Java Development Kit). Otros puntos destacados incluyen soporte para Maven3 y HTML5, así como mejoras en el editor de Java. (NetBeans, 2011).

Sistema Gestor de Base de Datos

“Un Sistema de Gestión de Bases de Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una BD nunca se accede o manipula directamente sino a través del SGBD. Se puede considerar al SGBD como la interfaz entre el usuario y la BD.” (Nc, 2011)

Existen sistemas gestores de bases de datos relacionales y documentales (o no relacionales). En este trabajo de diploma se abordarán los relacionales debido a que la base de datos que utilizará la aplicación será relacional.



PostgreSQL

PostgreSQL es un potente sistema de bases de datos de código abierto del objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y la corrección.

Una base de datos de clase empresarial, PostgreSQL cuenta con características avanzadas tales como control de concurrencia multi-versión (MVCC), punto en el tiempo de recuperación, tablespaces, replicación asíncrona, transacciones anidadas (puntos de retorno), en línea / copias de seguridad en caliente, un planificador de consulta sofisticada / optimizador, y escribir por delante de registro para la tolerancia a fallos. Es compatible con conjuntos de caracteres internacionales, codificaciones de caracteres multibyte, Unicode, y es la configuración regional para la clasificación, el caso de la sensibilidad-, y el formato. Es altamente escalable, tanto en la gran cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede acomodar. Hay sistemas activos de PostgreSQL en entornos de producción que manejan más de 4 terabytes de datos. (PostgreSQL Global Development Group , 1996-2012)

Fundamentos de la selección

Una vez analizados los sistemas gestores de bases de datos existentes en el mundo entre los que figuran Oracle, MySQL, entre otros, se selecciona a PostgreSQL debido a que el equipo de desarrollo está más familiarizado con el mismo y responde a las necesidades de la facultad de utilizar software libre, además por todas las ventajas mencionadas anteriormente.

Maven



Apache Maven es un software de gestión de proyectos y una herramienta de comprensión. Basado en el concepto de un modelo de objeto del proyecto (POM), Maven se puede gestionar un proyecto de construcción, elaboración de informes y la documentación de una pieza central de la información. A primera vista Maven puede parecer muchas cosas, pero en pocas palabras Maven es un intento de aplicar los patrones de construcción de la infraestructura de un proyecto con el fin de promover la comprensión y la productividad al proporcionar un camino claro en el uso de mejores prácticas. (Apache Maven, 2012)

Permite manejar las dependencias de un proyecto de modo que se sepa en todo momento las librerías y versiones requeridas para que se ejecute y una vez hecho esto lo ejecuta, lo compila, lo empaqueta y le hace pruebas. Posee un repositorio local que utiliza como caché para evitar descargar nuevamente todas las librerías reduciendo de este modo tiempo de carga del proyecto.

1.5. Conclusiones del capítulo

Una vez analizados los conceptos principales que se relacionan de forma estrecha con el objeto de estudio del presente trabajo de diploma ha sido posible comprender íntegramente el mismo. Al analizar cada una de las herramientas y lenguajes existentes que permiten desarrollar aplicaciones como la que se propone ha sido posible determinar cuáles son las óptimas para utilizar en el desarrollo de la presente tesis.

Al concluir este capítulo se pueden apreciar los resultados de la investigación realizada mediante el análisis, la valoración, la síntesis y la toma de posición en el trabajo realizado a través de las múltiples fuentes consultadas.



Capítulo 2. Características, Análisis y Diseño del Sistema

Introducción

De modo que sea posible dirigir el desarrollo del SIGOB hacia un sistema que cumpla con lo establecido es necesario identificar las características que debe poseer para lograr una perfecta comprensión del contexto del mismo. En este capítulo se realiza el modelo de dominio para definir las características del sistema a implementar y se especifican los Requisitos Funcionales y No Funcionales. Además se realiza la propuesta del sistema, se elaboran las historias de usuarios y las tareas de ingeniería asociadas a las mismas y se describe la arquitectura empleada en el desarrollo de la solución.

2.1. Concepción inicial

Actualmente el SIGOB no cuenta con una integración que garantice la interoperabilidad de cada una de las direcciones que comprende de modo que funcionen correctamente como un todo, es por ello que el objetivo de este trabajo de diploma es precisamente integrar todos los módulos del sistema para el Gobierno de Artemisa de modo que funcionen como una sola aplicación.

2.2. Levantamiento de requisitos

Para realizar el levantamiento de requisitos se respetaron las pautas establecidas por la metodología seleccionada para guiar el desarrollo de la implementación elaborando el artefacto Lista de Reserva del Producto (LRP).

Este artefacto posibilita recopilar en lenguaje natural todas las funcionalidades que el sistema debe cumplir una vez culminado el proceso de levantamiento de información conformando una lista con prioridad para organizar el trabajo a realizar. El objetivo es asegurar que el producto definido al terminar la lista es correcto, útil y competitivo adaptándose a los cambios en el entorno del mismo.

Esta lista puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas.

LRP asociada al presente trabajo de diploma: Ver Anexo #2

2.3. Planificación del proyecto por roles

Rol	Responsabilidad	Nombre
Gerente	Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la definición de objetivos y requerimientos, así como en la selección de los miembros del Equipo del Proyecto. Tiene la responsabilidad de controlar el progreso del software y trabaja junto con el Líder de Proyecto en la reducción de la Lista de reserva del producto, así como en la de riesgos.	Dania Fernández Aguilar
Cliente	Es el encargado de facilitar toda la información necesaria para el desarrollo del sistema.	Administración Provincial de Artemisa
Miembros del Equipo		
Líder de Proyecto, Programador y Analista	Es la encargada de integrar el sistema tanto para el lado del cliente como del servidor así como de implementar componentes reutilizables para todo el proyecto.	Guelmis L. Martínez Ramírez

Líder de Proyecto, Diseñador de Base de Datos	Es el encargado de integrar los esquemas independientes de cada una de las direcciones del Gobierno en una sola base de datos, garantizar que la misma esté normalizada y que funcione correctamente.	Edel Pedrol Alvarez
---	---	---------------------

2.4. Descripción de la arquitectura

La arquitectura empleada en el desarrollo de la solución es la Arquitectura en Capas. Esta arquitectura se utiliza cuando existe un sistema de grandes dimensiones que requiere ser descompuesto en grupos de subtareas o componentes tales que cada grupo de subtareas está en un nivel determinado de abstracción. Esta arquitectura divide la aplicación en 4 capas lógicas que poseen un grupo de interfaces definidas para cada una de ellas:

La primera capa es conocida como la de **Presentación**, es la empleada por el cliente web. Contiene el patrón Modelo Vista Controlador (MVC) que permite separar la lógica de la interfaz externa del usuario, facilitando su desarrollo de modo independiente.

- **Modelo:** Gestiona el proceder de la aplicación con respecto a los datos manejados en la misma respondiendo a las peticiones realizadas sobre las vistas permitiendo su actualización desde el controlador.
- **Vista:** Interpreta las acciones del usuario con respecto a las operaciones del negocio y modificando la navegación entre las vistas según los resultados obtenidos.
- **Controlador:** Responde a eventos, comúnmente acciones realizadas por el usuario, invoca peticiones al modelo y también a la vista.



La segunda es conocida como capa **Intermedia** o capa de **Servicios de Negocio**. Conocida también como capa Lógica de la Aplicación, recibe la entrada de la capa de presentación, interactúa con los servicios de datos para ejecutar las operaciones de negocio automatizadas por la aplicación y devuelve el resultado procesado a la capa de presentación.

La tercera es la capa de **Acceso a Datos** es una porción de código donde se accede a los datos.

La cuarta capa es la de **Persistencia** que posee los datos necesarios para la aplicación. Es la encargada de almacenar, recuperar y mantener los datos conjuntamente con su integridad.

El **Diseño con Metáforas** describe cómo debería funcionar el sistema, es un diseño de solución más simple que pueda funcionar y ser implementado en un momento dado. En la siguiente figura se muestra el diseño correspondiente a la arquitectura del sistema.

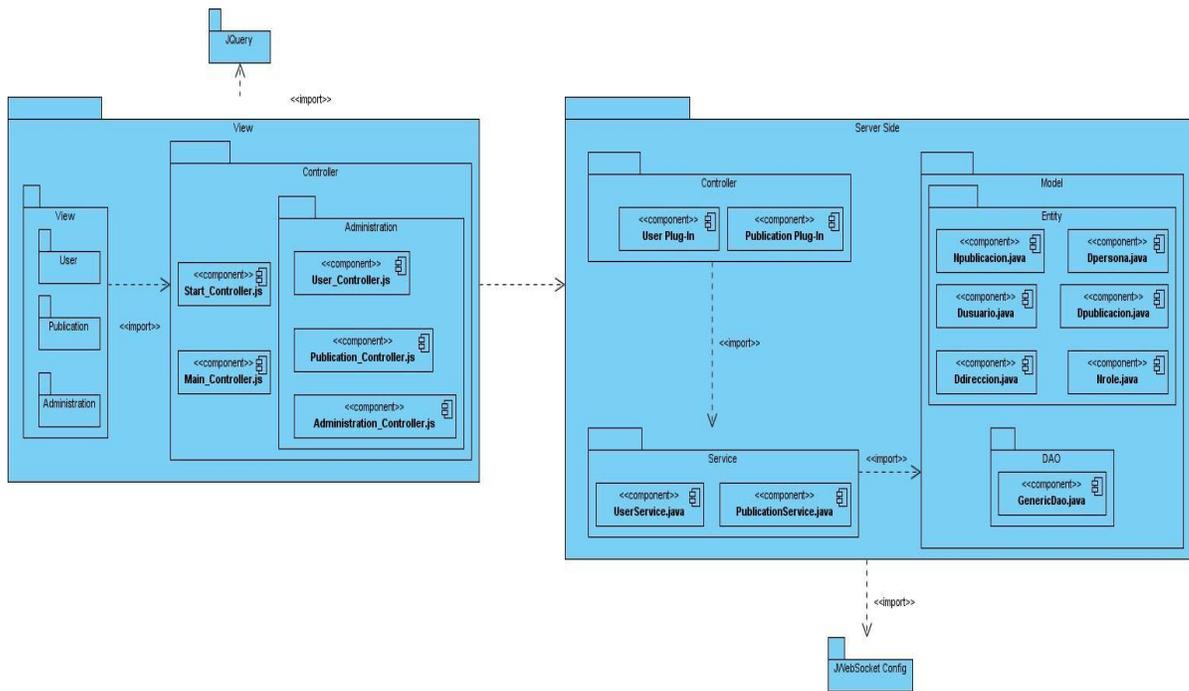


Fig #1. Diseño con Metáforas para el módulo de Administración del SIGOB

- El paquete de presentación contiene las clases correspondientes a la recepción y devolución de datos de la aplicación a los clientes según los requisitos funcionales detectados en el negocio.
- El paquete Controlador contiene los PlugIns y los Servicios, en estos se manejan y procesan los datos de la aplicación acorde a los requisitos funcionales.
- El paquete Modelo contiene las principales entidades mapeadas que permitirán almacenar y procesar la información que será gestionada.

2.5. Modelo de Dominio

La metodología SXP define una serie de actividades para guiar el proceso de desarrollo de un software, entre una de las más importantes figura la definición del Modelo de Historias de Usuario del Negocio. Este modelo permite realizar una



descripción detallada del negocio manejado, pero si el mismo no está bien definido entre los clientes y los desarrolladores se le llama Modelo de Dominio.

El diagrama de dominio propuesto para la solución a implementar es el siguiente:

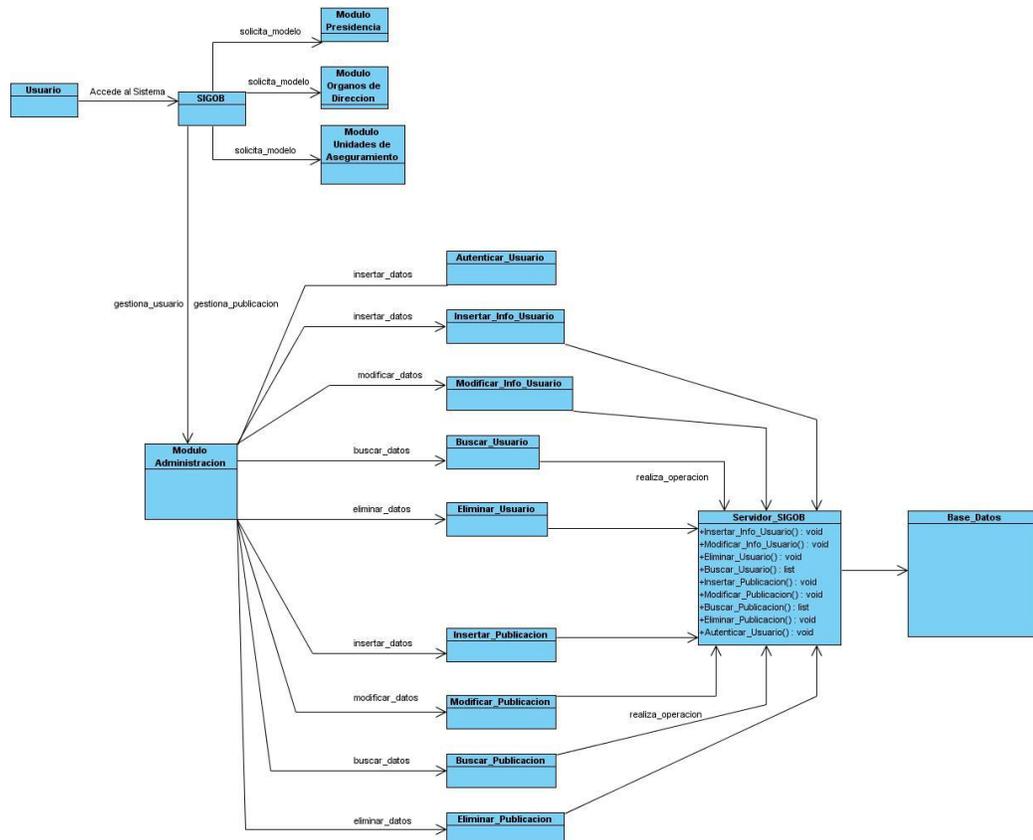


Fig #2. Modelo de Dominio

2.6. Historias de Usuario y Tareas de Ingeniería

Las Historias de Usuario (HU) son la que tienen como principal función la de describir las tareas a realizar por el sistema basándose fundamentalmente en las especificaciones dadas por el cliente del sistema. Estas HU son escritas en lenguaje natural y de forma concisa, de modo que sirvan como guía en la construcción de las pruebas de aceptación que se desarrollará más adelante.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Guelmis L. Martínez Ramírez	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 3 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 3 semanas
Descripción: La presente historia de usuario tiene como objetivo insertar, modificar, listar y eliminar toda la información referente a los usuarios del CAP.	
Observaciones: Se necesita estar autenticado como administrador de la aplicación para poder acceder a esta HU.	
Prototipo de interfaz: Autenticar Administrador: Esta interfaz permite que el administrador del sistema se autentique. Ver Anexo #3	
Vista Inicial de Administración: una vez que el administrador se autentique en el sistema accederá a esta vista principal donde realizará todas las funciones de administración definidas para el mismo. Ver Anexo #4	
Insertar Usuario: Ver Anexo #5	
Buscar Usuario: Esta interfaz es empleada antes de modificar o eliminar un usuario, permite buscar el usuario que cumpla con un criterio y seleccionar el que se quiere modificar o eliminar. Ver Anexo #6	
Modificar Usuario: Esta interfaz permite listar los usuarios existentes una	

vez que se haya encontrado al menos una coincidencia en la búsqueda realizada anteriormente y seleccionar el que se desea modificar. Ver Anexo #7

Una vez seleccionado el usuario a modificar se muestra una interfaz con los datos llenos para su posterior modificación: Ver Anexo #8

Eliminar Usuario: Esta interfaz permite listar los usuarios existentes una vez que se haya encontrado al menos una coincidencia en la búsqueda realizada anteriormente y seleccionar el que se quiere eliminar. Ver Anexo #9

Tareas de Ingeniería asociadas a la HU_1: Ver Anexo #10

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Gestionar la información de las publicaciones generadas en el CAP.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Guelmis L. Martínez Ramírez	Iteración Asignada: 3
Prioridad en Negocio: Medio	Puntos Estimados: 2 semanas y 2 días
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas y 2 días
Descripción: La presente historia de usuario tiene como objetivo añadir, modificar, eliminar y buscar toda la información referente a las publicaciones generadas en el CAP.	
Observaciones: Se necesita estar autenticado como especialista de alguna dirección en específico o como administrador de la aplicación para poder ejecutar esta HU. Las publicaciones permitidas son Reflexiones de Fidel,	

Convocatorias y Reglamentos.

Prototipo de interfaz: Insertar Publicación: Ver Anexo #11

Buscar Publicación: Esta interfaz es empleada antes de modificar una publicación, permite buscar la publicación que cumpla con un criterio y seleccionar la que se quiere modificar. Ver Anexo #12

Modificar Publicación: Esta interfaz permite listar las publicaciones existentes una vez que se haya encontrado al menos una coincidencia en la búsqueda realizada anteriormente y seleccionar la que se desea modificar. Ver Anexo #13

Una vez seleccionada la publicación a modificar se muestra una interfaz con los datos llenos para su posterior modificación: Ver Anexo #14

Eliminar Publicación: Esta interfaz permite listar las publicaciones existentes una vez que se haya encontrado al menos una coincidencia en la búsqueda realizada anteriormente y seleccionar la que se quiere eliminar: Ver Anexo #15

Tareas de Ingeniería asociadas a la HU_2: Ver Anexo #16

2.7. Plan de Releases

El Plan de Releases permite definir el número de iteraciones en que se desarrollará el sistema de modo que se puedan definir con exactitud las entregas inmediatas y la entrega final. Este artefacto posee una estrecha relación con las Historias de Usuario anteriormente detalladas, ya que se tiene en cuenta la prioridad definida por el cliente para cada historia y se colocan en un orden con prioridad. Al definir las prioridades se cada una de estas historias se arribó a la siguiente planificación:

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
2	En esta iteración se implementarán las HU que permitirán garantizar un funcionamiento básico de la aplicación, dígase la correcta gestión de los usuarios y de las publicaciones generadas en el CAP.	HU_1 HU_2	33 días

2.8. Propuesta de Solución

Se propone desarrollar el Sistema Informativo del Gobierno (SIGOB), este debe permitir al usuario final gestionar la información de los procesos manejados en cada una de las direcciones de la Administración Provincial de Artemisa. La interfaz principal del SIGOB estará diseñada para que cualquier usuario la encuentre amigable aunque no tenga conocimientos avanzados de computación.

2.9. Conclusiones del capítulo

Al concluir el desarrollo de este capítulo se obtuvo como resultado una mejor definición del modelo conceptual sobre el que se basa el presente trabajo de diploma contribuyendo así a un mejor dominio del problema en cuestión. Se establecieron y aprobaron los requisitos funcionales y no funcionales necesarios para lograr que el producto resultante cumpla con todas las expectativas del cliente y posea la calidad requerida en todos los productos de la Universidad de las Ciencias Informáticas. Se obtiene la aplicación SIGOB como propuesta de solución para gestionar los procesos de las direcciones de la AP.

Capítulo 3. Implementación y Validación del Sistema

Introducción

En el presente capítulo queda documentada la implementación de la solución propuesta a través de la elaboración del diagrama de despliegue, el estándar de código empleado, la programación realizada a partir de los requerimientos y la exposición de las pruebas que se le realizaron al sistema para darle seguimiento a los cambios en aras de garantizar la calidad del producto final. Esto fue posible gracias a la elaboración de los casos de pruebas, también conocidos como pruebas de aceptación y las pruebas unitarias. Además se exponen los resultados obtenidos hasta el momento así como su aporte social y económico.

3.1. Diagrama de Despliegue

El diagrama de despliegue permite modelar la arquitectura en tiempo de ejecución de un sistema ya que sitúa el software en el hardware que lo contiene. A continuación se muestra el diagrama de despliegue correspondiente al sistema SIGOB donde cada nodo representa un recurso de cómputo necesario para el despliegue del sistema:

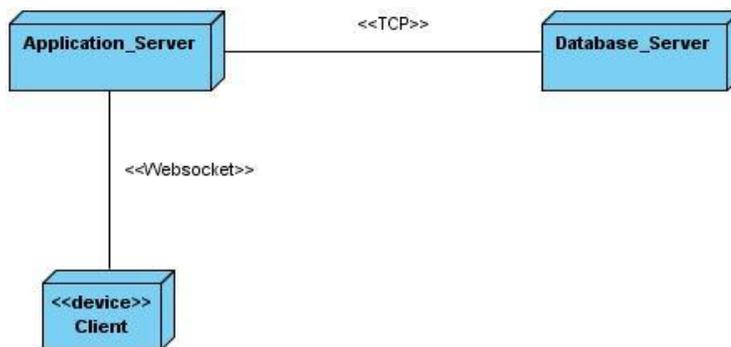


Fig #3. Diagrama de Despliegue

- **Cliente:** Este se conecta a través del protocolo WebSocket al Servidor jWebSocket utilizando cualquier navegador que soporte este protocolo independientemente del sistema operativo instalado.

- **Servidor jWebSocket:** El servidor de jWebSocket se conectará al servidor de base de datos de PostgreSQL a través del protocolo TCP/IP para persistir o consultar los datos.

3.2. Estándar de Código

Un estándar de codificación abarca todo lo referente al proceso de generación de código. Establecer el estándar es necesario al iniciar cualquier proyecto de software para asegurar que todos los miembros del equipo de desarrollo trabajen de forma similar.

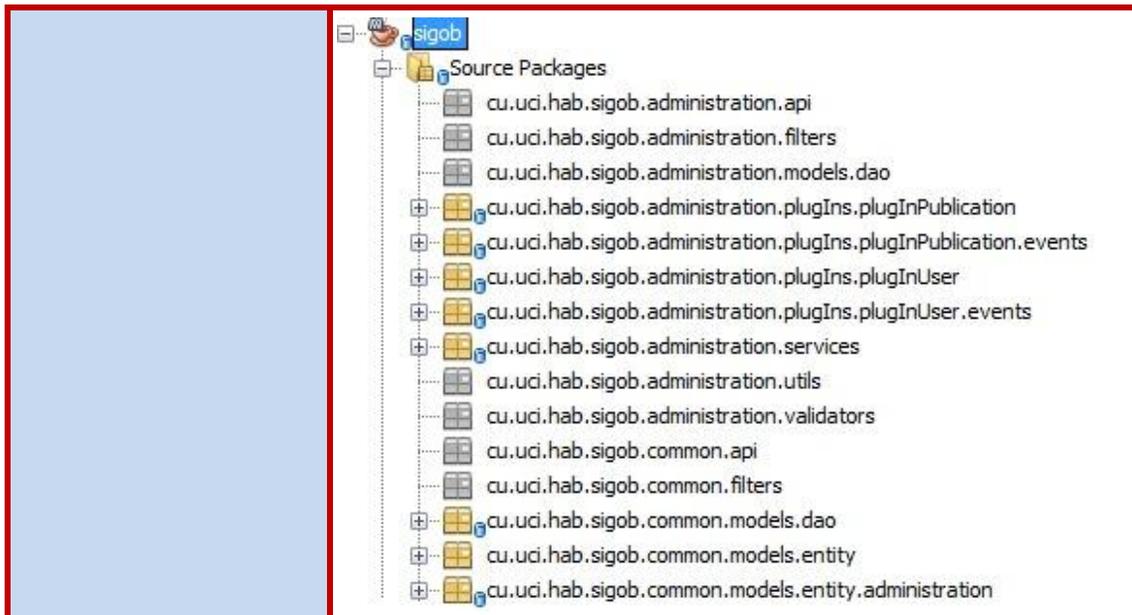
Para el desarrollo del Sistema Informativo del Gobierno se utilizaron varias reglas a seguir, entre ellas:

- Para la definición de los nombres de los identificadores se empleó el convenio establecido por CamelCase.
- Para las clases, enumeradores e interfaces se hizo uso de la variante UpperCamelCase.
- Se usaron 4 espacios de indentación en el código.
- Cada línea debe contener hasta 80 caracteres.
- Los atributos de las clases se declararon al inicio de las mismas exceptuando los iteradores.
- Se utilizaron comentarios en bloque al comienzo de cada clase o interfaz y antes de cada método.

3.3. Estructura de la Implementación

3.3.1. En el lado del Servidor

Estructura de los paquetes:	Espacio de nombre base: cu.uci.hab.sigob.administration cu.uci.hab.sigob.common.models.entity.administration
-----------------------------	--



cu.uci.hab.sigob.administration:

Contiene las clases principales del módulo que permiten la gestión de usuarios y publicaciones, o sea las funcionalidades ejecutadas por el administrador del sistema.

cu.uci.hab.sigob.administration.plugIns.pluginPublication.events:

cu.uci.hab.sigob.administration.plugIns.pluginUser.events:

Eventos: Son las acciones ejecutadas sobre la aplicación por el usuario. Al producirse un evento en el cliente este dispara la funcionalidad asociada al mismo en el servidor. Es un mensaje de software que indica que algo ha ocurrido.

cu.uci.hab.sigob.administration.plugIns.pluginUser:

cu.uci.hab.sigob.administration.plugIns.pluginPublication:

Plugins: Extensiones que permiten proporcionar soporte a los acontecimientos ocurridos en el lado del servidor, básicamente son los

listeners' para notificar a los eventos relacionados con el mismo.

cu.uci.hab.sigob.administration.services

Servicios: Posibilitan que se realicen las operaciones solicitadas por el cliente pues ordenan la ejecución de consultas en la bases de datos mediante el uso del GenericDao.

cu.uci.hab.sigob.common.models.entity.administration:

Contiene las clases principales mapeadas por Hibernate para el esquema de administración.

A continuación se exponen algunos ejemplos de elementos que componen la integración del sistema:

```
public class DeleteUser extends C2SEvent {  
  
    private List<Integer> usersId;  
  
    /**  
     * @return the user  
     */  
    public List<Integer> getUsersId() {  
        return usersId;  
    }  
  
    /**  
     * @param user the user to set  
     */  
    @ImportFromToken  
    public void setUsersId(List<Integer> usersId) {  
        this.usersId = usersId;  
    }  
}
```

Fig #4. Evento DeleteUser del PlugIn Usuario

```
public class UserPlugIn extends EventModelPlugIn {

    UserServiceImpl service;

    /**
     * Event LoadDirections
     * @param aEvent, aResponseEvent
     * @return
     */
    public void processEvent(LoadDirections aEvent,
        C2SResponseEvent aResponseEvent) {
        try {
            List<Ddireccion> directions = service.getDirectionsEntity();
            aResponseEvent.getArgs().setList("list", directions);
        } catch (Exception e) {
            aResponseEvent.getArgs().setString("error", e.getMessage());
        }
    }

    /**
     * Event LoadRoles
     * @param aEvent, aResponseEvent
     * @return
     */
    public void processEvent(LoadRoles aEvent,
        C2SResponseEvent aResponseEvent) {
        try {
            List<Nrol> roles = service.getNrolesEntiy();
            aResponseEvent.getArgs().setList("list", roles);
        } catch (Exception e) {
            aResponseEvent.getArgs().setString("error", e.getMessage());
        }
    }
}
```

Fig #5. PlugIn Usuario

```
public class UserServiceImpl {

    private GenericDao dao;

    /**
     * Esta funcion adiciona un usuario a la base de datos
     * @param addUser
     */
    @Transactional
    public void addUser(AddOrUpdateUser addUser) throws Exception {
        Dpersona personToSave = new Dpersona();
        personToSave.setName(addUser.getName());
        personToSave.setLastName(addUser.getLastName());
        personToSave.setNci(addUser.getNci());
        personToSave.setPersonalAddress(addUser.getPersonalAddress());
        personToSave.setSex(addUser.getSex());
        dao.save(personToSave);

        Dusuario userToSave = new Dusuario();
        userToSave.setUserName(addUser.getUserName());
        String password = Tools.getMd5(addUser.getPassword());
        userToSave.setPassword(password);
        userToSave.setDpersona(personToSave);
        Ddireccion direction = (Ddireccion) dao.findByPK(Ddireccion.class,
            addUser.getDirectionId().longValue());
        userToSave.setDirection(direction);
        Nrol role = (Nrol) dao.findByPK(Nrol.class,
            addUser.getRoleId().longValue());
        userToSave.setNrole(role);
        dao.save(userToSave);
    }
}
```

Fig #6. Servicio para el PlugIn Usuario

3.3.2. En el lado del Cliente

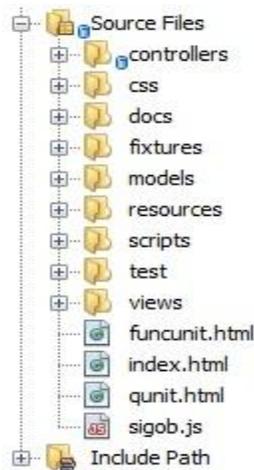


Fig #7. Directorios asociados a la aplicación cliente

Descripción de los elementos:

A continuación se describirán los elementos utilizados en el desarrollo de esta aplicación. Aunque el marco de trabajo cuenta con muchas más características en el presente sistema solo se hacen uso de las siguientes:

- controllers: Este directorio contiene todos los controladores de la aplicación. Estos permiten organizar los manejadores de eventos de forma muy fácil y eficiente.
- resources: Este directorio contiene todas las librerías, plugIns y widgets utilizados para el desarrollo de la aplicación.
- scripts: Este directorio contiene los scripts necesarios para el funcionamiento básico de cualquier aplicación realizada sobre este marco de trabajo.
- views: Este directorio contiene todas las vistas asociadas al sistema ordenadas acorde a la estructura interna del organismo propietario de la aplicación.
- index.html: Es un archivo de hipertexto. Estos suelen ser de contenido estático, no dinámico. Incluye el HTML de la aplicación.

- sigob.js: Este archivo es fundamental en la aplicación ya que es el encargado de cargar todos los elementos necesarios para hacer que la aplicación funcione, dígame, controladores, modelos, plugins, etc.

A continuación se muestra un controlador de ejemplo perteneciente al módulo de administración:

```
$.Controller('users',{
  init:function(){
    controller = this;
    $.existsUser = -1;
  },
  "history.administration.addUser subscribe" : function(){
    $.Administration.Handler.showView("management/user/addUser");
    $.Administration.Commons.loadRoles("userRol");
    $.Administration.Commons.loadDirections("directionAdd");
  },
  "history.administration.searchUser subscribe" : function(){
    $.Administration.Handler.showView("management/user/searchUser");
    $.Administration.Commons.loadRoles("searchRol");
    $.Administration.Commons.loadDirections("searchDirection");
  },
});
```

Fig #8. Controlador Usuario

3.4. Diseño de casos de prueba

En la etapa de prueba del software se crean una serie de casos de prueba que intentan "destruir" el software desarrollado. La prueba requiere que se descarten ideas preconcebidas sobre la "calidad o corrección" del software desarrollado. Este proceso consiste en diseñar pruebas tratando de encontrar errores con la mínima cantidad de tiempo y esfuerzo. Para realizar pruebas a un software pueden utilizarse dos modos:

- **Pruebas de caja blanca:** consiste en desarrollar pruebas que aseguren que la operación interna está acorde a las especificaciones establecidos y que cada componente interno se ha probado adecuadamente.
- **Pruebas de caja negra:** consiste en realizar una serie de pruebas que permitan comprobar que cada función es operativa, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

A continuación se muestran algunos de los tipos de pruebas de caja blanca y negra que existen:

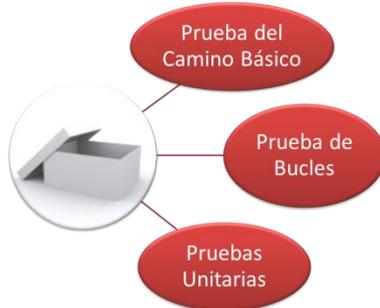


Fig #9. Tipos de Prueba de Caja Blanca

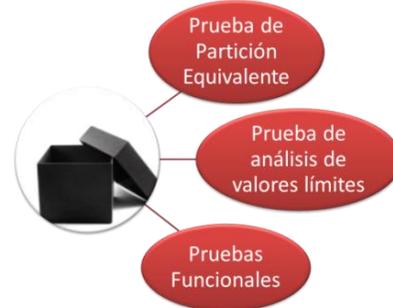


Fig #10. Tipos de Prueba de Caja Negra

3.5. Pruebas de Caja Blanca realizadas al SIGOB

Las pruebas unitarias son pruebas de caja blanca que permiten realizar comprobaciones al software para garantizar el correcto funcionamiento de las clases así como de sus principales funcionalidades durante la implementación. Este tipo de prueba tiene como objetivo principal garantizar que cada método desarrollado sea un elemento funcional para garantizar el paso a una nueva etapa sin errores anteriores.

Se diseñaron los casos de prueba para cada servicio con el framework JUnit de modo que se lograra garantizar la correcta funcionalidad del sistema. Se le realizaron pruebas a cada método, pero en el presente trabajo de diploma solamente se abordarán dos de ellos. A continuación se muestran las clases que genera el framework por cada servicio.



Fig #11. Clases de Prueba generadas por el framework JUnit para el módulo de Administración

A continuación se muestran los métodos generados para probar algunas de las funcionalidades.

```
/**
 * Test of verifyUser method, of class UserServiceImpl.
 */
public void testVerifyUser() throws Exception {
    VerifyUser userToVerify = new VerifyUser();
    userToVerify.setUserName("glmartinez");
    userToVerify.setNci("88061740813");
    String nci = "";
    String userName = "";
    UserServiceImpl instance = new UserServiceImpl();
    Integer expectedResult = 1;
    Integer result = instance.verifyUser(userToVerify);
    assertEquals(expectedResult, result);
    System.out.println("verifyUser");
    System.out.println(result);
}
```

Fig #12. Método VerifyUser de la clase UserServiceImplTest

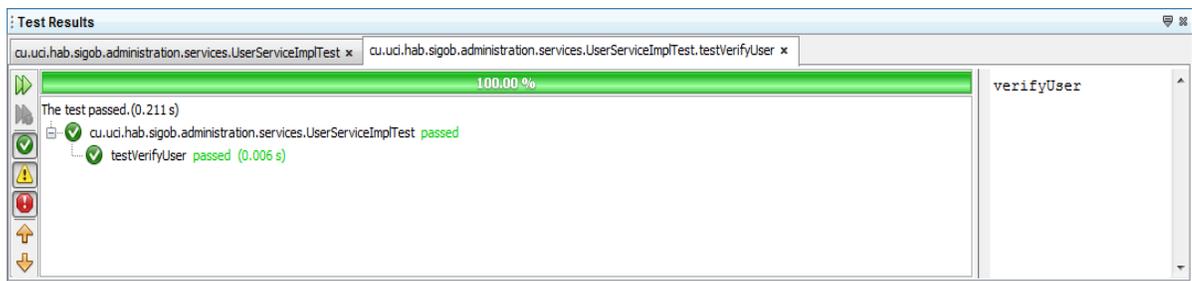


Fig #13. Resultados Obtenidos luego de realizar la prueba al método VerifyUser

```
/**
 * Test of addUser method, of class UserServiceImpl.
 */
public void testAddUser() throws Exception {
    AddOrUpdateUser userToAdd = new AddOrUpdateUser();
    userToAdd.setName("Guelmis");
    userToAdd.setLastName("Martinez");
    userToAdd.setPersonalAddress("Ciudad de la Habana");
    userToAdd.setSex("Femenino");
    userToAdd.setDirectionId(1);
    userToAdd.setRoleId(1);
    userToAdd.setUserName("glmartinez");
    userToAdd.setPassword("123");
    userToAdd.setNci("88061740813");
    userToAdd.setUserId(1);
    UserServiceImpl instance = new UserServiceImpl();
    boolean expectedResult = true;
    boolean result = instance.addUser(userToAdd);
    assertEquals(expectedResult, result);
    System.out.println("addUser");
    System.out.println(result);
}
```

Fig #14. Método AddUser de la clase UserServiceImplTest

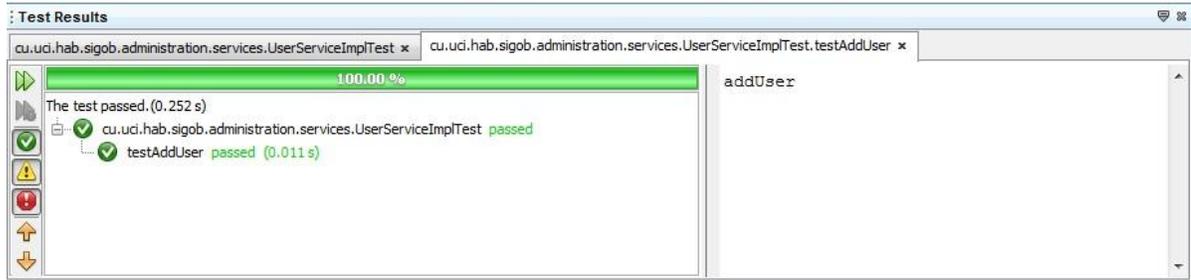


Fig #15. Resultados Obtenidos luego de realizar la prueba al método AddUser

3.6. Pruebas de Caja Negra realizadas al SIGOB

Casos de prueba de Aceptación

La ejecución y aprobación de los casos de prueba de aceptación son definidas por el cliente de la aplicación pero es el equipo de desarrollo el máximo responsable de su preparación, ya que les permite medir cualitativamente el trabajo realizado. Para cada historia de usuario existente se definió un caso de prueba, a continuación se muestran las pruebas realizadas.

3.6.1. Casos de Prueba para la HU Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-1-1	Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se autentique un usuario.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario exista en la base de datos del sistema.	

Entrada / Pasos de ejecución: Para autenticarse en el sistema debe dar clic en la opción Autenticar del menú de navegación ubicado en la parte superior de la interfaz de la aplicación, luego debe insertar su nombre de usuario y contraseña personal y luego debe hacer clic en el botón Aceptar. En caso de que no quiera autenticarse debe hacer clic en el botón Cancelar y todos los datos insertados serán borrados.

Resultado Esperado: El usuario podrá acceder a los módulos de la aplicación acorde al nivel de acceso definido en sus credenciales.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación

Código Caso de Prueba: sigob-1-2

Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).

Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez

Descripción de la Prueba: Esta prueba consiste en verificar que se autentique un usuario como administrador del sistema y pueda acceder a la vista principal de administración.

Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario exista en la base de datos del sistema.

Entrada / Pasos de ejecución: Para autenticarse en el sistema debe dar clic en la opción Autenticar del menú de navegación ubicado en la parte superior de la interfaz de la aplicación, luego debe insertar su nombre de usuario y contraseña personal y luego debe hacer clic en el botón Aceptar. En caso de que no quiera autenticarse debe hacer clic en el botón Cancelar y todos los datos insertados serán borrados.

Resultado Esperado: El usuario podrá acceder a la vista principal de administración.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-1-3	Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda insertar un usuario.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.	
Entrada / Pasos de ejecución: Para insertar un nuevo usuario en el sistema debe mover el mouse sobre la opción del menú <u>Gestionar Usuario</u> y dar clic en la opción <u>Adicionar Usuario</u> , se mostrará un formulario con los datos requeridos por el sistema para adicionar al mismo a la base de datos de la aplicación y luego debe hacer clic en el botón Aceptar. En caso de que no quiera adicionar el usuario debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.	
Resultado Esperado: El usuario quedará registrado en la base de datos y será reconocido como usuario del sistema con sus credenciales de acceso en dependencia del rol asignado por el administrador del mismo.	
Evaluación de la Prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-1-4	Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda buscar uno o varios usuarios que coincidan con el criterio de búsqueda especificado y mostrarlos al	

administrador del sistema ya sea para su modificación o eliminación.

Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.

Entrada / Pasos de ejecución: Para buscar un usuario en el sistema debe mover el mouse sobre la opción del menú Gestionar Usuario y dar clic en la opción Buscar Usuario, se mostrará un formulario con los datos requeridos por el sistema para buscar al mismo en la base de datos de la aplicación, debe seleccionar si desea buscar usuarios para modificarlos o eliminarlos y luego debe hacer clic en el botón Aceptar. En caso de que no quiera buscar usuarios debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.

Resultado Esperado: En caso de encontrar usuarios que correspondan a los criterios seleccionados se mostrará una lista con los datos de los mismos, en caso de que la búsqueda fuera para modificar solo se podrá seleccionar uno y en caso de eliminación se podrán seleccionar todos los que se deseen. Si no se encuentra ningún usuario que corresponda a los criterios de búsqueda insertados se mostrará un mensaje notificando que no se encontraron coincidencias.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación

Código Caso de Prueba: sigob-1-5

Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).

Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez

Descripción de la Prueba: Esta prueba consiste en verificar que se puedan modificar los datos de un usuario seleccionado.

Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.

Entrada / Pasos de ejecución: Para modificar un usuario en el sistema debe mover el mouse sobre la opción del menú Gestionar Usuario y dar clic en la opción Buscar Usuario, se mostrará un formulario con los datos requeridos por el sistema para buscar al mismo en la base de datos de la aplicación. Debe seleccionar la opción Para Modificar y luego debe hacer clic en el botón Aceptar, se mostrará una lista con las coincidencias encontradas, luego se selecciona el usuario que se desea modificar y luego debe hacer clic en el botón Aceptar. Se mostrará una interfaz con los datos del usuario seleccionado que podrán ser modificados según se desee y luego debe hacer clic en el botón Aceptar. En caso de que no quiera modificar el usuario seleccionado debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.

Resultado Esperado: Se modificarán los datos del usuario seleccionado.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-1-6	Nombre Historia de Usuario: Gestionar la información de los usuarios de la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda eliminar uno o varios usuarios seleccionados.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.	
Entrada / Pasos de ejecución: Para modificar un usuario en el sistema debe mover el	

mouse sobre la opción del menú Gestionar Usuario y dar clic en la opción Buscar Usuario, se mostrará un formulario con los datos requeridos por el sistema para buscar al mismo en la base de datos de la aplicación. Debe seleccionar la opción Para Eliminar y luego debe hacer clic en el botón Aceptar, se mostrará una lista con las coincidencias encontradas, luego se selecciona el o los usuarios que se desea eliminar y luego debe hacer clic en el botón Aceptar. En caso de que no quiera eliminar el o los usuarios seleccionados debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.

Resultado Esperado: Se eliminará el o los usuarios seleccionados.

Evaluación de la Prueba: Satisfactorio

3.6.2. Casos de Prueba para la HU Gestionar la información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-2-1	Nombre Historia de Usuario: Gestionar la información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda insertar una publicación.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.	
Entrada / Pasos de ejecución: Para insertar una nueva publicación en el sistema debe mover el mouse sobre la opción del menú <u>Gestionar Publicación</u> y dar clic en la opción <u>Adicionar Publicación</u> , se mostrará un formulario con los datos requeridos por el sistema para adicionar la mismo a la base de datos de la aplicación y luego debe hacer clic en el botón Aceptar. En caso de que no quiera adicionar la publicación debe hacer clic en el	

botón Limpiar y todos los datos insertados serán borrados.

Resultado Esperado: La publicación quedará registrada en la base de datos.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación

Código Caso de Prueba: sigob-2-2

Nombre Historia de Usuario: Gestionar la información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).

Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez

Descripción de la Prueba: Esta prueba consiste en verificar que se pueda buscar una o varias publicaciones que coincidan con el criterio de búsqueda especificado y mostrarlos al administrador del sistema ya sea para su modificación o eliminación.

Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.

Entrada / Pasos de ejecución: Para buscar una publicación en el sistema debe mover el mouse sobre la opción del menú Gestionar Publicación y dar clic en la opción Buscar Publicación, se mostrará un formulario con los datos requeridos por el sistema para buscar la misma en la base de datos de la aplicación, debe seleccionar si desea buscar publicaciones para modificarlas o eliminarlas y luego debe hacer clic en el botón Aceptar. En caso de que no quiera buscar publicaciones debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.

Resultado Esperado: En caso de encontrar publicaciones que correspondan a los criterios seleccionados se mostrará una lista con los datos de las mismas, en caso de que la búsqueda fuera para modificar solo se podrá seleccionar una y en caso de eliminación se podrán seleccionar todas las que se deseen. Si no se encuentra ninguna publicación que corresponda a los criterios de búsqueda insertados se mostrará un mensaje notificando que no se encontraron coincidencias.

Evaluación de la Prueba: Satisfactorio

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-2-3	Nombre Historia de Usuario: Gestionar la información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se puedan modificar los datos de una publicación seleccionada.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.	
Entrada / Pasos de ejecución: Para modificar una publicación en el sistema debe mover el mouse sobre la opción del menú <u>Gestionar Publicación</u> y dar clic en la opción <u>Buscar Publicación</u> , se mostrará un formulario con los datos requeridos por el sistema para buscar la misma en la base de datos de la aplicación. Debe seleccionar la opción <u>Para Modificar</u> y luego debe hacer clic en el botón Aceptar, se mostrará una lista con las coincidencias encontradas, luego se selecciona la publicación que se desea modificar y luego debe hacer clic en el botón Aceptar. Se mostrará una interfaz con los datos de la publicación seleccionada que podrán ser modificados según se desee y luego debe hacer clic en el botón Aceptar. En caso de que no quiera modificar la publicación seleccionada debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.	
Resultado Esperado: Se modificarán los datos de la publicación seleccionada.	
Evaluación de la Prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código Caso de Prueba: sigob-2-4	Nombre Historia de Usuario: Gestionar la

	información de las publicaciones generadas en la Administración Provincial de Artemisa (AP).
Nombre de la persona que realiza la prueba: Guelmis L. Martínez Ramírez	
Descripción de la Prueba: Esta prueba consiste en verificar que se pueda eliminar una o varias publicaciones seleccionadas.	
Condiciones de Ejecución: Para poder realizar la prueba se tiene que utilizar un navegador que soporte el protocolo WebSocket, el servidor de la aplicación tiene que estar ejecutándose y se necesita que el usuario autenticado posea el rol de administrador.	
Entrada / Pasos de ejecución: Para modificar una publicación en el sistema debe mover el mouse sobre la opción del menú <u>Gestionar Publicación</u> y dar clic en la opción <u>Buscar Publicación</u> , se mostrará un formulario con los datos requeridos por el sistema para buscar la misma en la base de datos de la aplicación. Debe seleccionar la opción <u>Para Eliminar</u> y luego debe hacer clic en el botón Aceptar, se mostrará una lista con las coincidencias encontradas, luego se selecciona la o las publicaciones que se desea eliminar y luego debe hacer clic en el botón Aceptar. En caso de que no quiera eliminar la o las publicaciones seleccionadas debe hacer clic en el botón Limpiar y todos los datos insertados serán borrados.	
Resultado Esperado: Se eliminará la o las publicaciones seleccionadas.	
Evaluación de la Prueba: Satisfactorio	

3.7. Resultados obtenidos

Al concluir el presente trabajo se obtiene como resultado el SIGOB en su versión 1.0 que permite informatizar los principales procesos de la Administración Provincial de Artemisa.

3.8. Funcionalidades Obtenidas

Entre las principales funcionalidades y características que posee SIGOB en su versión 1.0 se pueden mencionar:

- Es capaz de adicionar, modificar, buscar y eliminar un usuario.
- Es capaz de adicionar, modificar, buscar y eliminar una publicación.
- Provee información de cada uno de los municipios que componen la provincia.
- Gestiona todos los modelos con información relevante de la provincia en cada una de las direcciones de la AP.

3.9. Aporte social y económico.

SIGOB es una aplicación desarrollada con el marco de trabajo jWebSocket. Esta mejorará significativamente el flujo de trabajo del Gobierno de Artemisa debido a que automatizará gran parte de los procesos que allí se ejecutan actualmente de forma manual.

Esta aplicación brindará un gran potencial al trabajo de este organismo contribuyendo a la rapidez en la entrega y recepción de informes de cada uno de los municipios que es uno de los principales procesos ejecutados en la entidad.

Otro aspecto importante es que la informatización de la AP forma parte del proceso de informatización de las empresas que se está llevando a cabo en nuestro país contribuyendo de este modo al perfeccionamiento empresarial y al ahorro de tiempo y esfuerzo innecesario del personal del centro llegando finalmente a la satisfacción del usuario como cliente final.

3.10. Conclusiones del capítulo

Al concluir este capítulo culmina a su vez el desarrollo de las pruebas al sistema demostrando con el análisis de los resultados obtenidos, que las funcionalidades alcanzadas por SIGOB, se han desarrollado de acuerdo a los requerimientos definidos en la etapa inicial y en el período establecido. Se realizó el diseño de metáforas y los casos de pruebas en conjunto con las pruebas unitarias que guiaron la calidad de la aplicación desarrollada mostrando que las funciones de la

aplicación son operativas y que se produce un resultado correcto. Demostrándose así que el sistema se encuentra en óptimas condiciones para su despliegue.

Conclusiones Generales

- Se creó un sistema que eleva los grados de centralidad, seguridad e integridad de la información en el proceso de gestión de la misma en la Administración Provincial de Artemisa.
- Se logró una mejor organización; disminuyendo la complejidad de implementación de las historias de usuarios.
- Se dio respuesta a los requisitos funcionales propuestos para el desarrollo del SIGOB.
- Se obtuvo SIGOB en su versión 1.0 como producto que permite informatizar los principales procesos de la Administración Provincial de Artemisa.

Recomendaciones

Desarrollar la segunda versión de SIGOB, adicionándole nuevas funcionalidades tales como:

- Importar Modelos Excel.
- Exportar a PDF los reportes generados.
- Validar ataques a través de inyecciones SQL al sistema.

Referencias Bibliográficas

- © 2012 Mozilla Developer Network. (2012). *Mozilla Developer Network*. (© 2012 Mozilla Developer Network) Recuperado el 17 de Enero de 2012, de https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript
- Apache Maven. (2012). *Apache Maven*. Recuperado el 16 de Mayo de 2012, de <http://maven.apache.org/>
- ChristianMania. (23 de Octubre de 2011). *Christian Mania Desarrollo de Sistema & Soluciones Web*. (Copyright © 2008 - christianmania.com) Recuperado el 17 de Enero de 2012, de <http://www.christianmania.com/2011/10/metodologias-de-desarrollo-de-software-%C2%BFcual-es-el-camino/>
- Clavero Quintana, L. I., & Díaz Padilla, L. (8 de Septiembre de 2008). Recuperado el 18 de Enero de 2012, de Revista Electrónica editada y publicada por el CDIP del Universidad de Ciencias Pedagógicas "Conrado Benítez García": <http://conrado.revistas.rimed.cu/articulo/mostrar/la-gesti-n-de-informaci-n-una-herramienta-para-el-bibliotecario-escolar>
- codigoprogramacion.com. (2010-2012). *Codigoprogramacion.com*. Recuperado el 30 de Enero de 2012, de <http://codigoprogramacion.com/java/47-introjjava.html>
- Definición.de. (2008-2011). *Definición.de*. Recuperado el 31 de Enero de 2012, de <http://definicion.de/java/>
- Definición.de. (2008-2011). *Definición.de*. (Definición.de) Recuperado el 17 de Enero de 2012, de <http://definicion.de/html/>
- Definición.de. (2008-2012). *Definición.de*. (Definición.de) Recuperado el 18 de Mayo de 2012, de <http://definicion.de/sistema-de-informacion/>
- Díaz, Ana Rosa. (16 de Mayo de 2010). Sociedad de Desarrollo en Videojuegos, SODVI. Recuperado el 17 de Enero de 2012, de <http://sodvi.info/blog/es-MX/entry/1711516073/Diferenciacion-de-conceptos-en-el-area-del-desarrollo-de-aplicaciones>

- González Cornejo, J. E. (2009). *DocIRS*. (Document Information Retrieval Systems, DocIRS) Recuperado el 17 de Enero de 2012, de <http://www.docirs.cl/uml.htm>
- Jupiter Consulting - JavaScriptMVC Training and Support. (2012). *JavascriptMVC*. Recuperado el 31 de Enero de 2012, de <http://javascriptmvc.com/>
- NetBeans. (2011). (Oracle Corporation) Obtenido de NetBeans: <http://netbeans.org/features/index.html>
- Nourie, D. (24 de Mayo de 2005). (Oracle Corporation) Recuperado el 2011, de Oracle Sun Developer Network (SDN): <http://java.sun.com/developer/technicalArticles/tools/intro.html>
- NubeHost.mx. (12 de Mayo de 2011). *NubeHost.mx*. Recuperado el 1 de Febrero de 2012, de Web hosting en la nube: <http://nubehost.mx/blog/tag/framework/>
- PostgreSQL Global Development Group. (1996-2012). *PostgreSQL*. Recuperado el 5 de Febrero de 2012, de <http://www.postgresql.org/about/advantages/>
- producingoss.com. (s.f.). *Capítulo 3. Infraestructura Técnica*. Recuperado el 1 de Febrero de 2012, de <http://producingoss.com/es/vc.html>
- RapidSVN. (2011). *RapidSVN*. Recuperado el 13 de 12 de 2011, de <http://www.rapidsvn.org/>
- Sáez Bertrand, J., Rodríguez Acero, F., & Albert, D. (2009-2010). Universidad Complutense de Madrid. Recuperado el 17 de Mayo de 2012, de http://eprints.ucm.es/11272/1/Generador_automatico_de_casos_de_prueba_unitarios_para_Junit.pdf
- *Visual Paradigm*. (2012). (s.f.). Recuperado el 11 de Diciembre de 2011, de <http://www.visual-paradigm.com>

Bibliografía

- Alfaro, F. M. (2011). Herramientas Case. *COLECCION CULTURA INFORMATICA*.
- Alvarez, M. A. (2009). Qué es jQuery, para qué sirve y qué ventajas tiene el utilizar este framework Javascript.
- Alvarez, M. A. (2010). YUI es un framework para desarrollo de webs que contiene librerías Javascript y CSS para crear interfaces de usuario típicas de los sitios de contenido enriquecido.
- Apache Software Foundation. (10 de 12 de 2011). *Apache Subversion*. Recuperado el 10 de 12 de 2011, de Apache Subversion: <http://subversion.apache.org/>
- CAVSI. (2011). *CAVSI*. Recuperado el 2 de Febrero de 2012, de <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>
- Chacon, S. (2012). *Git te fasto version control system*. Recuperado el 8 de Febrero de 2012, de <http://git-scm.com/about>
- Chiavenato, I. (2006). *Introducción a la Teoría General de la Administración* (Séptima Edición ed.). McGraw-Hill Interamericana.
- Curto, J. (28 de Noviembre de 2006). *Information Management*. Recuperado el 13 de Enero de 2012, de Reflexiones sobre las tecnologías de la información: <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>
- Definición.de. (2008-2011). *Definición.de*. Recuperado el 13 de Diciembre de 2011, de <http://definicion.de/gestion/>
- Definición.de. (2008-2011). *Definición.de*. Recuperado el 31 de Enero de 2012, de <http://definicion.de/sistema/>
- Definición.de. (2008-2011). *Definición.de*. Recuperado el 14 de Diciembre de 2011, de <http://definicion.de/sistema-de-informacion/>

- Definición.de. (2008-2011). *Definición.de*. Recuperado el 12 de Diciembre de 2011, de <http://definicion.de/modelo-de-gestion/>
- Definición.de. (2008-2011). *Definición.de*. (Definición.de) Recuperado el 13 de Enero de 2012, de <http://definicion.de/informacion/>
- Díaz, A. R. (16 de Mayo de 2010). *Sociedad de Desarrollo en Videojuegos, SODVI*. (Niuware.com) Recuperado el 17 de Enero de 2012, de <http://sodvi.info/blog/es-MX/entry/1711516073/Diferenciacion-de-conceptos-en-el-area-del-desarrollo-de-aplicaciones>
- Eclipse. (2011). (Eclipse Foundation) Obtenido de The Eclipse Foundation Open Source Community Website: <http://www.eclipse.org/>
- Eguíluz Pérez, J. (2009). *CSS avanzado*. http://www.librosweb.es/css_avanzado/capitulo5/el_framework_yui.html.
- ExtJS. (2009). *www.extjs.es*. Recuperado el 10 de 12 de 2011, de <http://www.extjs.es>
- García, P. (2009). ¿Qué es Mootools?
- JavaTM. (s.f.). *JavaTM*. Recuperado el 31 de Enero de 2012, de http://www.java.com/es/download/faq/whatis_java.xml
- JavaTM. (s.f.). *JavaTM*. Recuperado el 31 de Enero de 2012, de <http://java.com/es/about/>
- Murphey, R. (2010). *jQuery Fundamentals*. Autoedición.
- Nc, J. (2011). *SlideShare Inc*. Recuperado el 1 de Diciembre de 2011, de <http://www.slideshare.net/jeissonlarry/sistema-gestin-de-bases-de-datos-2657624>
- Oracle Corporation and/or its affiliates. (2012). *MySQL The world's most popular open source database*. Recuperado el 30 de Enero de 2012, de <http://www.mysql.com/why-mysql/>
- Peñalver, G., Meneses, A., & García, S. (2010). *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE*. Chile: 1er congreso Iberoamericano de Ingeniería de Proyectos.

- Programming4us. (2012). *Programming4us*. Recuperado el 4 de Febrero de 2012, de <http://mscerts.programming4.us/es/20552.aspx>
- Proietti, V. (2009). Recuperado el 14 de 12 de 2011, de Mootools: <http://mootools.net/>
- Prototype Core Team. (2007). *Prototype*. Recuperado el 14 de 12 de 2011, de <http://www.prototypejs.org/>
- Proyecto jQuery. (2010). *License*. Recuperado el 8 de 12 de 2011, de jQuery: <http://jquery.org/license/>
- Rational. (s.f.). *Grupo Soluciones Innova*. Recuperado el 11 de 12 de 2011, de <http://www.rational.com.ar/herramientas/roseenterprise.html>
- Rational. (s.f.). *Grupo Soluciones Innova*. Recuperado el 11 de 12 de 2011, de <http://www.rational.com.ar/herramientas/roseenterprise.html>
- Real Academia Española. (s.f.). *Diccionario de la Lengua Española Vigésima Segunda Edición*. Recuperado el 31 de Enero de 2012, de REAL ACADEMIA ESPAÑOLA:
http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=g%C3%A9nesis
- Sánchez, J. (2006). ¿Qué es un 'framework'?
- Schulze, A. (2011). Framework Approach for WebSockets.
- Sencha. (2011). *Ext JS 4 JavaScript Framework for Rich Apps in Every Browser*. Recuperado el 14 de 12 de 2011, de Sencha: <http://www.sencha.com/products/extjs/>
- Soriano, J. (2011). jQuery, la librería Javascript por excelencia. Un framework Javascript lleno de ventajas.
- Tavárez, D. (2009). Comparación de Frameworks en Javascript.
- team, T. (2011). *TortoiseSVN*. Recuperado el 13 de 12 de 2011, de <http://tortoisesvn.net/>
- Thompson, I. (Octubre de 2008). © 2011 *Promonegocios.net: Portal de Mercadotecnia con Artículos, Directorios y Foros*. Recuperado el 31 de Enero de 2012, de <http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>

- TortoiseSVN. (2012). *TortoiseSVN*. Recuperado el 6 de Febrero de 2012, de http://tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-preface.html
- Villa, C. (2011). Carga de clases dinámicamente con ExtJS 4.
- Wordpress.com. (06 de 08 de 2007). *Librería ExtJS*. Recuperado el 13 de 12 de 2011, de Desarrollo WEB y otras Hierbas: <http://vargasti.wordpress.com/2007/08/06/libreria-extjs/>
- Yahoo! Inc. (2011). *YUI Library*. Recuperado el 14 de 12 de 2011, de Yahoo!: <http://developer.yahoo.com/yui/>