

Universidad de las Ciencias Informáticas

Facultad Regional “Mártires de Artemisa”



Título: Módulo de la Dirección de Educación de la Administración
Provincial de Artemisa

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

Autor: Johanna Carbonell Parra

Tutor: Ing. Anaibis Alvarez Morales

Cotutor: MsC. Olga Lidia Rodríguez Martínez

Artemisa, Junio 2012

“Año 54 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

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

Autor: Johanna Carbonell Parra

Tutor: Ing. Anaibis Alvarez Morales

CoTutor: MsC. Olga Lidia Rodríguez Martínez

FRASE



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

Ernesto Che Guevara



AGRADECIMIENTOS

A mi mamá por apoyarme siempre desde el primer momento, dándome todo su cariño y buenos consejos en los momentos más difíciles.

A mi novio por ser mi sostén, mi compañero, mi amigo, por ser comprensivo y tenerme paciencia.

A mi papá, Pedro y a mi familia en general.

A mi hermana y a mi cuñado por darme siempre aliento, amor, y ganas de salir adelante.

A mis amigas Adilen, Evelyn, y Yuslenis por estar conmigo desde el principio y compartir esta experiencia.

A todos aquellos que han contribuido a desarrollarme profesionalmente y me han ayudado a llegar hasta aquí.

A todos muchas gracias.





DEDICATORIA

Dedico este trabajo a mi mamá, compañera incondicional e incansable, por creer siempre en mí, en la posibilidad de poder lograr cualquier meta que me proponga en la vida, y por demostrarme siempre lo orgullosa que se siente de tenerme como hija. Mami, esto es por ti.



RESUMEN

En la actualidad las instituciones necesitan gestionar la información que procesan de forma rápida y eficaz mediante el uso de sistemas informáticos. Una de estas instituciones es la Administración Provincial de Artemisa la cual cuenta con 32 direcciones, siendo una de ellas la Dirección de Educación.

El objetivo principal de esta investigación es contribuir a la eficacia y confiabilidad en los procesos de gestión de información desarrollados por la Dirección de Educación. Este trabajo se basa en la implementación de un módulo capaz de responder a las necesidades de los trabajadores de la Dirección optimizando los procesos de búsqueda y control de la información almacenada, así como la creación de todos aquellos reportes que se requieran, potenciando de manera sustancial los procesos fundamentales que allí se realizan.

Se realizó un estudio de los sistemas informáticos empleados por otras instituciones en Cuba y el mundo para la gestión de la información. Se identificaron las tecnologías y herramientas adecuadas para implementar la propuesta de solución, así como las pruebas pertinentes que describe la metodología de desarrollo aplicada. Fueron implementados todos los requerimientos funcionales deseados obteniendo resultados satisfactorios, además de la documentación pertinente obtenida a lo largo del trabajo.

Palabras Claves: procesos, gestión de información, AP, Educación, sistemas.

ÍNDICE

| | |
|--|-----------|
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... | 9 |
| 1.1 Conceptos asociados al dominio del problema..... | 9 |
| 1.2 Sistemas de Gestión de Información. Tendencias Actuales..... | 14 |
| 1.3 Metodologías de Desarrollo de <i>Software</i> | 19 |
| 1.4 Herramientas y Tecnologías Actuales..... | 24 |
| 1.4.1 Lenguaje de Modelado..... | 24 |
| 1.4.2 Herramientas CASE..... | 25 |
| 1.4.3 Tecnologías del lado del servidor..... | 27 |
| 1.4.4 Entorno de Desarrollo Integrado..... | 29 |
| 1.4.5 <i>Frameworks de desarrollo</i> | 31 |
| 1.4.6 Herramientas para el control de versiones..... | 34 |
| 1.4.7 Herramienta para la gestión de proyectos java..... | 35 |
| CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA..... | 37 |
| 2.1 Procesos de gestión de información a automatizar..... | 37 |
| 2.2 Propuesta de solución..... | 38 |
| 2.3 Planificación del Proyecto por Roles..... | 39 |
| 2.4 Modelo de Dominio..... | 39 |
| 2.5 Lista de Reserva del Producto (LRP)..... | 40 |
| 2.6 Historias de Usuario y Tareas de Ingeniería..... | 40 |
| 2.7 Plan de Release..... | 43 |
| 2.8 Arquitectura de Software..... | 44 |
| 2.9 Patrones de Diseño..... | 45 |

| | |
|--|-----------|
| 2.10 Diagrama con Metáfora | 48 |
| 2.11 Diagrama de Componentes | 49 |
| CAPÍTULO 3: ADQUISICIÓN Y VALIDACIÓN DE LOS RESULTADOS DEL SISTEMA .. | 51 |
| 3.1 Estándares de Códificación | 51 |
| 3.2 Implementación del lado del servidor | 52 |
| 3.2.1 Creación de los eventos de la aplicación..... | 52 |
| 3.2.1 Creación de las clases services de la aplicación. | 53 |
| 3.1.3 Creación de las clases plugin de la aplicación. | 55 |
| 3.3 Casos de Prueba | 56 |
| 3.3.1 Pruebas de caja blanca:..... | 56 |
| 3.3.1 Pruebas Unitarias | 57 |
| 3.4 Resultados Obtenidos..... | 61 |
| 3.5 Diagrama de Despliegue | 61 |
| 3.6 Funcionalidades Obtenidas | 62 |
| 3.7 Aporte Social y Económico..... | 63 |
| BIBLIOGRAFÍA REFERENCIADA..... | 66 |
| BIBLIOGRAFÍA CONSULTADA | 71 |

INTRODUCCIÓN

El hombre ha necesitado a lo largo de la historia almacenar información en cualquier actividad productiva en la que esté inmerso, convirtiéndose la misma en el mediador principal de todo tipo de actividades y en uno de los recursos más valiosos con los que cuenta una organización en sus diferentes procesos, contribuyendo al desarrollo adecuado de los mismos. A diario se toman decisiones que indican el curso de las empresas o proyectos en los cuales se trabaja, por lo que es necesario que la información posea la calidad requerida.

Con el vertiginoso desarrollo de la informática y las telecomunicaciones, la cantidad de datos manejados ha aumentado considerablemente con el decursar de los años, dando lugar a la utilización de las Tecnologías de la Información y las Comunicaciones (TIC). El surgimiento de estas tecnologías ha tenido un impacto significativo en prácticamente todas las esferas de la sociedad, llegando a convertirse en un instrumento indispensable que posibilita la informatización de los procesos que se desarrollan dentro de las organizaciones, haciendo uso de sistemas que permiten gestionar la información de acuerdo a las exigencias demandadas por los usuarios.

La gestión de la información a nivel mundial se ha convertido en un proceso importante en las instituciones y empresas, debido a que permite organizar y poner en uso los recursos de información de la organización, además de ganar competitividad, eficiencia y eficacia, permitiéndoles obtener posiciones privilegiadas dentro del mercado. Se utiliza para mejorar el capital empresarial y para tener control de los recursos materiales.

Cuba está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos. En los últimos años, dentro de los lineamientos económicos, políticos y sociales, se ha establecido como reto el desarrollo tecnológico de todos los sectores de la

sociedad, auspiciado por la máxima dirección del estado. Uno de estos sectores, priorizado por la Revolución Cubana, es la Educación, con la valiosa misión de formar y preparar a las nuevas generaciones.

La Universidad de las Ciencias Informáticas (UCI) surgió como parte del proyecto de informatización que asume el país. Tiene entre sus principales objetivos la formación de ingenieros informáticos altamente calificados y la vinculación directa de los estudiantes a la producción de software. Presenta un modelo de integración docencia-producción e investigación, que le ha permitido a convertirse en una universidad de excelencia.

La UCI está integrada por diversas facultades, entre ellas se encuentran las facultades regionales ubicadas en Artemisa, Ciego de Ávila y Granma. La Facultad Regional “Mártires de Artemisa” asume las estrategias de trabajo de la universidad, desarrollando *software* para diferentes entidades nacionales entre las que se destaca la Administración Provincial (AP) de Artemisa. Entre las 32 direcciones de la AP se encuentra la Dirección de Educación, que tiene como misión dirigir, ejecutar y controlar la aplicación de la Política del Estado y el Gobierno en la actividad educacional de la provincia, excepto en la Educación Superior.

Actualmente la Dirección de Educación se encarga de tramitar toda la información referente al control y ejecución de las actividades educacionales en la provincia, las cuales son rectoradas por el Ministerio de Educación. La información gestionada en la Dirección procede de cada uno de los municipios, la misma está relacionada con las altas y bajas estudiantiles, disponibilidad de profesionales, matrículas de estudiantes, entrega de materiales docentes, proyecciones de cursos escolares por cada centro educacional, tecnologías educativas, resultados de las evaluaciones de los estudiantes y profesores según los distintos niveles de escolaridad, salud y promoción escolar, así como la capacitación y superación de los profesores. Dicha información se recibe en diferentes períodos de tiempo ya sea mensual, trimestral, semestral o anualmente.

Con el transcurso de la investigación se detectó la siguiente **situación problemática**: Toda la gestión de la información de los procesos llevados a cabo en la Dirección se realiza de forma manual tramitada por diversas vías: telefónica, correo electrónico o mediante mensajeros que se encargan de su entrega tanto en formato duro como digital.

Debido a la manipulación y organización de la información de forma manual y al gran cúmulo de datos generados en la Dirección, el proceso de búsqueda y consulta de la información resulta lento al no obtener la información consultada con la inmediatez requerida. La información que se necesita se encuentra almacenada en documentos Word y Excel, los cuales no son flexibles ni eficaces a la hora de realizar búsquedas avanzadas.

La información no se encuentra centralizada, por tanto no existe la posibilidad de que varios usuarios puedan acceder a la misma de forma simultánea y rápida, dada la necesidad de recopilar información proveniente de diferentes fuentes. Esto trae como consecuencia que el trabajo de las personas que laboran en la Dirección resulte engorroso, ya que la realización y control de todos los documentos generados requieren más tiempo del esperado, viéndose afectado el análisis de los mismos de manera eficaz.

Además de los problemas planteados surgen deficiencias que comprometen la confiabilidad de los datos con la ocurrencia de posibles errores humanos que se pueden producir durante el proceso, lo que trae consigo en muchas ocasiones duplicidad de la información. Los factores antes mencionados pueden repercutir de manera directa en la planificación educativa de la provincia, permitiendo que no se tomen en cuenta distintos indicadores para medir el trabajo educativo realizado por parte del personal docente tanto para realizar reestructuraciones en cuanto al incremento de profesores como las horas-clases a impartir.

Por todo lo planteado anteriormente el proceso de gestión de la información llevado a cabo en la Dirección de Educación carece de la eficacia y confiabilidad necesarias para la generación de reportes con calidad y en el tiempo requerido resultando poco fiable la información registrada en los mismos.

Debido a la situación problemática planteada se traza el siguiente **problema de investigación**: ¿Cómo contribuir al proceso de gestión de la información de la Dirección de Educación que permita elevar los niveles de eficacia y confiabilidad de la información que maneja la AP de Artemisa?

Definiendo como **objeto de estudio**: El proceso de gestión de información cuyo **campo de acción** se enmarca en: El proceso de gestión de información en el sector de Educación.

Se identificó como **objetivo general**: Desarrollar el Módulo de la Dirección de Educación que contribuya a elevar los niveles de eficacia y confiabilidad en el proceso de gestión de información de la Dirección de Educación de la AP de Artemisa.

Partiendo del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Establecer los fundamentos teórico-metodológicos para el desarrollo de los procesos de gestión de información.
2. Realizar el análisis y diseño de la solución de software propuesta para la Dirección de Educación de la AP de Artemisa.
3. Implementar el soporte para dar solución a los requerimientos de las aplicaciones clientes para la gestión de la información de la Dirección de Educación de la AP de Artemisa.
4. Validar mediante pruebas funcionales los resultados obtenidos con la solución.

Como **idea a defender** se establece que: El desarrollo del Módulo de la Dirección de Educación contribuirá a elevar los niveles de eficacia y confiabilidad en el proceso de gestión de información de la AP de Artemisa.

Para dar cumplimiento a los objetivos enunciados anteriormente se definen las siguientes **tareas de la investigación**:

- ✓ Elaboración del marco teórico y del estado del arte de la investigación.
- ✓ Determinación de la metodología a utilizar para el desarrollo de la solución propuesta.
- ✓ Especificación de las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.
- ✓ Realización del modelado del negocio a través del modelo de dominio de la solución de *software* propuesta.
- ✓ Identificación de los procesos de gestión de información de la Dirección de Educación de la AP de Artemisa.
- ✓ Identificación de los requerimientos de la solución de *software* propuesta.
- ✓ Elaboración del diseño de la solución de acuerdo a los modelos realizados y los requerimientos identificados.
- ✓ Definición e implementación de los eventos (metainformación) y de PlugIns en el servidor que dan soporte para atender los requerimientos de las aplicaciones clientes para la gestión de la información de la Dirección de Educación de la AP de Artemisa.
- ✓ Ejecución de las técnicas de prueba para la validación de la contribución lograda a partir del cumplimiento de los requisitos propuestos.

Durante el desarrollo de la investigación se utilizaron diversos métodos teóricos y empíricos tales como:

Métodos Teóricos

- ✓ **Histórico-Lógico:** permitió establecer la necesaria correspondencia entre los elementos lógicos e históricos, con el fin de analizar la evolución histórica de las soluciones informáticas existentes para la gestión de la información y conocer las tendencias actuales de las mismas.
- ✓ **Analítico-Sintético:** permitió analizar, comparar y confrontar las diferentes fuentes bibliográficas consultadas, posibilitando el análisis detallado y pormenorizado de las aplicaciones existentes en cuanto a la gestión de la información y las funcionalidades que estas brindan para llevarlas a cabo.
- ✓ **Modelación:** permitió modelar el proceso de diseño e implementación del sistema a través de la creación de modelos, con el objetivo de explicar cómo funcionan los procesos en la Dirección de Educación de la AP de Artemisa.

Métodos empíricos:

- ✓ **Entrevista:** permitió obtener a través del personal que labora en la Dirección, información acerca de cómo ocurren los procesos de almacenamiento, gestión y búsqueda de información en la Dirección de Educación de la AP de Artemisa. Se entrevistaron especialistas, jefes de departamento y directivos, con posibilidades de brindar mayor información. El 100 % del personal entrevistado coincide en la existencia de bajos niveles de eficacia y confiabilidad en los procesos de gestión de información desarrollados. **(Ver Anexo.1)**
- ✓ **Análisis Documental:** se utilizó para el análisis de toda la documentación existente en la Dirección de Educación, la misma aportó mayor información sobre los procesos que allí se desarrollan.

Población y Muestra: La población definida en la investigación está constituida por 30 trabajadores de la Dirección de Educación de la AP de Artemisa con la selección

de una muestra de 10 personas representando el 33 % de la población; la técnica seleccionada fue el muestreo intencional.

Variables dependientes: Eficacia y Confiabilidad.

Variable Independiente: Módulo de la Dirección de Educación de la AP de Artemisa.

Actualidad y Necesidad

La presente investigación se lleva a cabo por la necesidad que tiene la Dirección de Educación de la AP de Artemisa de elevar los niveles de eficacia y confiabilidad en el proceso de gestión de información que allí se desarrolla. En la actualidad la Dirección no cuenta con un sistema informático que controle toda la información generada en cuanto al control y ejecución de las actividades educacionales. Esto trae como consecuencia que el trabajo de las personas que laboran en la Dirección se vea limitado en cuanto a la eficacia, lo que provoca insatisfacción a aquellos que puedan estar involucrados, molestias innecesarias, pérdida de tiempo y aumento de la dificultad a la hora de realizar el trabajo. Debido a los motivos expuestos, es que surge la necesidad de desarrollar un Módulo de la Dirección de Educación de la AP de Artemisa que permita gestionar toda la información tramitada en la Dirección.

Los sistemas de gestión de información existentes a nivel mundial a pesar de tener características similares con la solución a implementar contienen marcadas diferencias ,pues responden solamente a los intereses trazados en dichas instituciones de acuerdo a sus necesidades, además la mayoría son desarrollados bajo licencias de *software* propietario. Por lo expresado anteriormente sus experiencias no son aplicables al entorno y la realidad del problema que se desea resolver, de esta manera se evidencia la novedad y el aporte del trabajo actual en el marco de la problemática a solucionar, adecuado a las condiciones de Cuba para implementar *software*.

Aporte Práctico

Módulo de la Dirección de Educación que contribuye a elevar los niveles de eficacia y confiabilidad en el proceso de gestión de información de la Dirección de Educación de la AP de Artemisa.

Estructura del Documento:

El trabajo de diploma se encuentra estructurado en tres capítulos:

Capítulo 1: Fundamentación Teórica.

Este capítulo aborda lo referente a la investigación de diferentes aplicaciones que manejan la gestión de información. Se realiza un estudio de las metodologías, herramientas y tecnologías más utilizadas en el desarrollo de *software*, para la fundamentación de las seleccionadas en la propuesta de solución.

Capítulo 2: Características, Análisis y Diseño del sistema.

En este capítulo se realiza una propuesta del sistema, se describe cómo debe funcionar y se destacan sus características distintivas; además, se especifican sus Requisitos Funcionales y No Funcionales y se muestran los modelos necesarios para la creación del diseño de la herramienta a realizar.

Capítulo 3: Implementación y Validación del Sistema.

Este capítulo incluye la programación realizada a partir de los requerimientos y los diagramas de diseño elaborados. Además, se valida la solución propuesta mediante los casos de pruebas unitarias, garantizando el correcto funcionamiento del módulo para suplir las necesidades de los usuarios de la Dirección de Educación de la AP de Artemisa.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

Este capítulo es el resultado de la búsqueda y análisis de la información relacionada con el objeto de estudio, se describen los conceptos fundamentales más relevantes, se realiza el análisis del estado del arte del tema tratado, de las tendencias, tecnologías, metodologías y herramientas usadas en la actualidad o de las que sirven de apoyo para la solución del problema a resolver.

1.1 Conceptos asociados al dominio del problema

Proceso

Ponjuán Dante (2004) señala que un proceso es entendido como “el suministro de insumos, la transformación de éstos (agregación de valor) y la obtención de un producto final para el usuario/cliente. Sin procesos no existirá transformación en un sistema”. Los procesos contribuyen a crear compromisos, generan recursos, proyectan una situación, solucionan conflictos y facilitan el logro de objetivos y metas.

El Diccionario de la Real Academia Española (2009) define proceso como un “conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial, también como un conjunto de operaciones lógicas y aritméticas ordenadas cuyo fin es la obtención de unos resultados determinados.”

A partir del análisis de las definiciones anteriores la autora de esta investigación se acoge a la definición realizada por la Dra. Ponjuán Dante ya que el contenido de la misma se ajusta a las características y propósitos del módulo a desarrollar.

Información

La información es un cúmulo de datos relacionados entre sí que tiene forma y significado en todo momento pero su valor está determinado a partir de que es poseída, concienciada y convertida en conocimiento por alguna entidad (DELGADO, 2006a). Permite resolver problemas y tomar decisiones, ya que su uso racional es la base del conocimiento.

Gestión

La gestión es el conjunto de acciones relativas a la planificación, la organización, la instrumentación, el direccionamiento y la supervisión del trabajo requerido para llevar estratégicamente a cabo una misión. (URDANETA, 1992)

Gestión de la información

Actualmente la sociedad presenta organizaciones basadas en el aprendizaje, donde el capital máspreciado es el ser humano, sustentándose en un desarrollo tecnológico constantemente en evolución, en el cual las grandes compañías planifican sus productos en función de la gestión de la información y de la viabilidad para su obtención.

Para tener una aproximación a los basamentos teóricos de la gestión de información, se han considerado los conceptos y definiciones de varios autores

- ✓ Páez Urdaneta (1992) afirma que “la gestión de información es el manejo de la inteligencia corporativa de una organización a objeto de incrementar sus niveles de eficacia, eficiencia y efectividad en el cumplimiento de sus metas”.
- ✓ Rodríguez y Domínguez (2005) definen la gestión de la información, como “aquel proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de la organización”.

✓ Bartle (2011) considera que” la gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas”.

A partir del análisis de los conceptos antes mencionados sobre la gestión de la información la autora de esta investigación se acoge a la definición realizada por Phil Bartle ya que el contenido de la misma se ajusta a las características y propósitos del módulo a desarrollar.

Proceso de gestión de la información

El proceso de gestión de información no es más que una determinada cantidad de actividades previstas en un orden lógico, que definen el tratamiento de la información. Permite relacionar el conocimiento, organiza y disemina los datos claves y la información necesaria para ayudar a la organización a responder a las necesidades de los clientes, busca la perpetuación y la materialización del potencial de las organizaciones. Además, permite que se identifiquen la innovación, la capacidad de respuesta y la productividad. (CABALLERO, 2004)

Sistema de gestión de la información

Los SGI son un conjunto de elementos relacionados y ordenados según ciertas reglas que aportan al sistema objeto, es decir, a la organización a la que sirven, y que marcan sus directrices de funcionamiento así como la información necesaria para el cumplimiento de sus fines, para ello debe recoger, procesar y almacenar datos procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. (DÍAZ; AMADOR, 2007)

La implementación de un sistema de gestión eficaz puede ayudar a:

- ✓ Gestionar los riesgos sociales, medioambientales y financieros.
- ✓ Mejorar la efectividad operativa.

- ✓ Reducir costos.
- ✓ Aumentar la satisfacción de clientes y partes interesadas.
- ✓ Proteger la marca y la reputación.
- ✓ Lograr mejoras continuas.

Servidor

Un servidor es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. (VEASUIP, 2010). Generalmente gestiona la mayor parte o la (totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación.

Un servidor es todo proceso que proporciona un servicio a otros siendo el “proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. (SIEMENS, 2011).

La autora de esta investigación asume como servidor un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta.

Funciones de un servidor en los sistemas de gestión.

- ✓ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ✓ Procesar requerimientos de bases de datos.
- ✓ Brindar la información requerida a los clientes.
- ✓ Procesar la lógica de la aplicación.
- ✓ Procesamiento de los datos entrados por el cliente.
- ✓ Brindar servicios de forma ágil a las solicitudes clientes.
- ✓ Funciona como intercambiador de mensajes entre las diferentes capas de la aplicación.
- ✓ Funciona como fachada entre la capa de presentación y la de datos, de forma que el interfaz se mantenga ignorante y por tanto independiente de la estructura interna de los datos.

Módulo

Un módulo es un fragmento de un programa que se desarrolla de forma independiente del resto del programa general. Esta independencia hace posible un mecanismo de compilación por separado que limita la complejidad del programa desarrollado. Al compilarse el módulo por separado se hace más fácil la detección y corrección de errores ya que el programa está segmentado y si el error está en un módulo determinado el programador solo revisa ese módulo sin modificar los demás. (GALLARDO, 2006)

Educación

La educación es un proceso de socialización y endoculturación de las personas a través del cual se desarrollan capacidades físicas e intelectuales, habilidades, destrezas, técnicas de estudio y formas de comportamiento ordenadas con un fin social. (ECURED, 2012)

Eficacia

Oliveira (2002) plantea que la eficacia "está relacionada con el logro de los objetivos/resultados propuestos, es decir con la realización de actividades que permitan alcanzar las metas establecidas. La eficacia es la medida en que alcanzamos el objetivo o resultado".

Thompson (2008) define la eficacia como " hacer lo necesario para alcanzar o lograr los objetivos deseados o propuestos"

La autora de esta investigación asume que el concepto más apropiado de eficacia es "hacer lo necesario para alcanzar o lograr los objetivos deseados o propuestos" definido por Thompson.

Confiabilidad

La confiabilidad se refiere a la seguridad y estabilidad en los resultados al realizar un proceso de obtención de datos garantizando la consistencia de los mismos, permitiendo así una correcta realización de las operaciones y mantenimiento de las funcionalidades previstas en un tiempo determinado.”(SUEIRO, 2009)

1.2 Sistemas de Gestión de Información. Tendencias Actuales

Debido al gran avance de las tecnologías de la informática y las comunicaciones, en el mundo existe una proliferación de sistemas de gestión, entre ellos los vinculados al sector educacional, la mayoría con universidades, pues ésta es la institución con más nivel tanto tecnológico como intelectual e inmersa en la búsqueda de soluciones a sus necesidades.

En el ámbito Internacional:

En el plano internacional existen diversos sistemas de gestión de información que se han destacado por su eficiencia y facilidad de uso.

SIGA: Sistema Integrado de Gestión Académica

SIGA es un sistema de gestión desarrollado en España. Es una aplicación que puede ser utilizada en muchos centros educacionales, no solo universitarios, sino también en conservatorios, academias, colegios, centros de formación de empresas, maestrías, postgrados, entre otros. Está compuesto por 13 módulos que cubren en su mayoría las necesidades de cualquier institución. Dichos módulos son: Alumnos, Opciones de alumnos, Diplomas, Profesores, Horario, Inventario, Ingreso-Gastos, Utilidades, Generación de listados, Económico, Automatización de exámenes, Automatización de encuestas, y Automatización de matriculación de alumnos. (DARA, 2009)

UNIVERSITAS XXI-ACADÉMICO es un sistema desarrollado con el fin de facilitar las gestiones académicas para los alumnos universitarios de Iberoamérica. Facilita la participación de profesores, estudiantes y personal de administración en todos

los procesos relacionados con la trayectoria universitaria del estudiante, así como en las gestiones administrativas de planificación.

Las áreas funcionales y de gestión que cubre UNIVERSITAS XXI - ACADÉMICO son la actividad académica del alumno, la planificación de recursos docentes y el acceso al campus virtual. La gestión del alumnado facilita la participación de profesores, estudiantes y personal de administración en los procesos relacionados con la trayectoria académica del alumno, desde su ingreso hasta la obtención del título. En este sentido, asume toda la gestión de alumnado de forma integral, y se orienta hacia la máxima eficiencia y seguridad. Este sistema ha sido implantado exitosamente en universidades de España, Colombia, Chile, Venezuela, Ecuador y República Dominicana. (CORBAL, 2008)

Los módulos que componen el sistema son: Planes de estudios, Recursos docentes, Gestión económica, Matrícula, Expediente, Actas, Gestión de propuestas de programas, Acceso, Becas, Doctorado, Estudios propios, Títulos, Estadísticas, Horarios, Gestión de bolsa de empleo, Relaciones internacionales, Tablas Generales, Administración del sistema, Comunicados, Generador de informes, Generador de cartas y etiquetas, Servicios telemáticos, Automatrícula WEB y Calificación de actas WEB (CORBAL, 2008)

La universidad de Chile ha implantado con éxito un **Sistema de Información de Gestión Académica**, el cual recopila y trata la información de distintos niveles de institución, por lo que se ha establecido como una herramienta fiable para la constatación, medición de procesos y acciones de la universidad, a través de indicadores de gestión medibles y comparables en el tiempo que conducen a un verdadero análisis institucional.

Los módulos con que cuenta el sistema son: Personal Académico, Autoridades, Personal Universitario, Asignaturas Pregrado, Programación Académica, Resumen de Certificación, Programa de Postgrado, Programa de Postítulos, Área de

Proyecto, Creación Artística y Cultural, Áreas Informes, Premios y Distinciones, Área de Docencia, Carreras y Programas académicos, Transferencia Tecnológica, Área de Extensión y Difusión, Extensión y Difusión, Área de Cooperación, Proyectos, Publicaciones ,Convenios, Estadías Nacionales e Internacionales, Área y Ventana de Noticias. (CHILE, 2005)

SIU-Guaraní: Sistema que administra la gestión académica desde que los alumnos ingresan como aspirantes hasta que obtienen el diploma. El sistema administra la gestión de alumnos en forma segura e íntegra, brindando información consistente y oportuna para la toma de decisiones.

El SIU-Guaraní brinda servicios a los alumnos, docentes, personal administrativo y autoridades de todo el sistema universitario. Las principales prestaciones que se le brindan a los usuarios que interactúan con el sistema son: Gestión de carreras y planes, Planificación, Gestión de matrícula, Gestión de cursado, Gestión de aulas, Gestión de exámenes, Gestión de equivalencias, Gestión de egresados, Administración y emisión de certificados, Mensajería a casillas de e-mail o celulares (docentes/alumnos/autoridades) y Gestión de encuestas para alumnos. (NORDESTE, 2006)

En el ámbito Nacional:

El desarrollo informático que ha obtenido el país en los últimos años ha evolucionado el pensamiento de la sociedad cubana en busca de nuevas soluciones para los problemas presentados en numerosas ramas, los centros educacionales no están ajenos a esta nueva revolución que ha generado el sector de la informática y es por eso que automatizan los procesos que realizan como búsqueda de solución a diversos problemas. Una de las actividades realizadas y que cuenta con varios sistemas desarrollados es lo referente a la gestión académica.

Actualmente en la Universidad de Matanzas Camilo Cienfuegos se encuentra implantado un sistema que ha demostrado la eficiencia y eficacia en la manipulación y procesamiento de la información académica. El sistema es capaz de realizar diversas acciones y brindar numerosos reportes como son:

- ✓ Búsqueda de un alumno.
- ✓ Listado de estudiantes por grupo.
- ✓ Reportes dinámicos de la información existente.
- ✓ Reporte de notas por asignatura y grupo.
- ✓ Tabla con los resultados docentes de un grupo en un semestre.
- ✓ Reporte de los resultados académicos de un estudiante en toda su carrera.
- ✓ Dar baja a un estudiante.

El sistema cuenta con varios módulos:

- ✓ Módulo de Administración para la gestión de las tablas del sistema vía Web.
- ✓ Módulo Web para las Secretarías Docentes para la Gestión de Estudiantes.
- ✓ Un Módulo Web para los Jefes de Departamentos docentes.
- ✓ Módulo Web para la Gestión de la Matrícula.
- ✓ Módulo Web para los Profesores. (DELGADO, 2006b)

El Instituto Superior Politécnico “José Antonio Hechevarría ” (CUJAE) ha creado un **sistema de gestión académica (SDI)** , desarrollado por el Grupo de Gestión Universitaria Digital dirigido a aumentar la eficiencia de la gestión universitaria, específicamente el control de los procesos docentes, su planificación y resultados.(ORQUÍN; BARRIOS, 2006)

SDI está conformado actualmente por 8 módulos y se trabaja para aumentar sus funcionalidades. Los módulos que están diseñados son: Secretaría de Facultades, Matrícula, Estadística, Archivo Histórico, Recuperación de Información, Pago de Estipendios, Planificación Docente y Postgrado.

De forma general el objetivo del sistema es registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado, como de postgrado, de forma automatizada. Entre las funciones más importantes que automatiza se encuentran: (ORQUÍN; BARRIOS, 2006)

- ✓ Promoción de los estudiantes de año, según su situación escolar.
- ✓ Conformación de grupos de estudiantes.
- ✓ Recuperación de información.
- ✓ Hoja Académica del estudiante.

La eficiencia en el proceso de gestión académica en la Universidad de las Ciencias Informáticas (UCI) se logra a través de **Akademios**, el mismo es un sistema automatizado para la gestión académica desarrollado por un equipo de trabajo de la Dirección de Informatización de la Universidad.

Akademios surge con el objetivo de permitir la gestión automatizada de los elementos que intervienen en la labor académica de un centro de estudios, que pueda enfrentar los cambios de forma natural, adaptándose a las nuevas condiciones y formas de hacer con el menor costo. El mismo se encuentra dividido en varios subsistemas que relacionados entre sí, automatizan las tareas del sistema, estos son: Matrícula, Estudiante, Registro, Profesor, Plan de Estudio, Seguridad y Reportes.

La mayoría de estos sistemas en su conjunto gestionan solamente información de los estudiantes, incorporando en muchas ocasiones módulos de atención de los mismos que requieren de mucha tecnología y son demasiado caros para ser implantados en cualquier centro estudiantil.

Además de los motivos expuestos anteriormente, por problemas de seguridad no se encuentra disponible el código fuente de los sistemas mencionados resultando

imposible realizar un estudio acerca de las funcionalidades de los mismos así como la reutilización del código perteneciente a la capa de lógica del negocio.

Después del análisis de las propuestas anteriores se decide que no es conveniente adoptar ninguno de estos sistemas debido a que la información procesada en la capa de procesamiento de datos de los mismos resulta demasiado específica, comparada con la información gestionada en la Dirección de Educación, resultando ineficientes dadas las necesidades actuales de la AP de Artemisa. Por lo tanto se hace necesario el desarrollo de un sistema que permita informatizar todos los procesos que estén vinculados a la gestión de la información de la Dirección de Educación de la AP de Artemisa.

1.3 Metodologías de Desarrollo de Software

Una metodología de desarrollo de *software* es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software, la cual puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. Estas tienen como principales objetivos garantizar la eficiencia, estructurar, planificar, minimizar las pérdidas de tiempo y controlar el proceso de desarrollo en sistemas de información. Actualmente se categorizan en dos grupos: Metodologías Ágiles y Tradicionales.

Metodologías Ágiles/Ligeras

Las metodologías ágiles o ligeras son metodologías orientadas a la interacción con el cliente y el desarrollo incremental del *software* mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando.(PÉREZ., 2008)

Entre las metodologías ágiles más destacadas se encuentran: XP¹ , Scrum, *Crystal Clear*, DSDM (*Dynamic Systems Development Method*), FDD (*Feature Driven Development*), ASD (*Adaptive Software Development*), Xbreed , *Extreme Modeling*. (ACUÑA, 1990)

Metodologías Tradicionales/Robustas

Las metodologías tradicionales o robustas son aquellas metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. (PÉREZ, 2008)

Entre las metodologías tradicionales más destacadas se encuentran: RUP (*Rational Unified Procces*), MSF (*Microsoft Solution Framework*), *Win-Win Spiral Model*, *Iconix*. (ACUÑA, 1990). A continuación se describen algunas de las metodologías de desarrollo de *software* más utilizadas en el mundo.

RUP²

Es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema *software*. Sin embargo, RUP es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas *software*, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. (JACOBSON,2000)

El ciclo de vida de RUP se caracteriza por ser: Dirigido por Casos de Uso, Centrado en la arquitectura e Iterativo e Incremental.

RUP define entre sus principales elementos a los trabajadores, actividades, artefactos y flujo de actividades, los cuales contribuyen a un proceso de desarrollo de *software* exitoso, además se caracteriza por dividir el ciclo de vida de la producción del *software* en 4 fases: Inicio, Elaboración, Construcción y Transición.

¹ Extreme Programming o Programación Extrema

² Rational Unified Process o Proceso Unificado de Desarrollo

RUP proporciona calidad, rendimiento, reutilización, seguridad y mantenimiento del *software* mediante una gestión sistemática de los riesgos. Es flexible, se puede ajustar a las necesidades del proyecto, brinda facilidades relacionadas a la organización y genera gran documentación, lo que contribuye a un mejor entendimiento del equipo de trabajo.

XP

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos, muy cambiantes, y donde existe un alto riesgo técnico. (TORRES; LÓPEZ, 2003)

Características de la metodología de desarrollo XP: (SÁNCHEZ, 2004)

- ✓ Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelante en algo hacia el futuro y se hagan pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- ✓ Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no hace en ese momento.

XP tiene como ventaja la comunicación que existe entre sus desarrolladores, además de la programación en pareja que permite que constantemente se esté

revisando el código. Esta metodología integra un cliente al equipo, esta característica permite que las dudas que les urjan a los desarrolladores sean resueltas al momento.

SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de *software* se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Principios de SCRUM

- ✓ Colaboración estrecha con el cliente.
- ✓ Predisposición y respuesta al cambio.
- ✓ Desarrollo incremental con entregas funcionales frecuentes.
- ✓ Comunicación verbal directa entre los implicados en el proyecto.
- ✓ Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- ✓ Simplicidad. Supresión de artefactos innecesarios en la gestión del proyecto.

SXP

SXP es una metodología de desarrollo de *software* compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del

equipo, ayudando al líder del proyecto a tener un mejor control del mismo. 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, puesta en marcha.
- ✓ Mantenimiento, 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, lo que permite mejorar el diseño cada vez que se le agregue una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes puedan ver día a día cómo progresa el trabajo. (ROMERO, 2008)

Metodología seleccionada para el desarrollo de la investigación

Luego de realizar un estudio de algunas metodologías de desarrollo de *software* existentes en la actualidad, se llega a la conclusión de que la metodología SXP es la más indicada para guiar el proceso de desarrollo del sistema que se desea implementar debido a que provee un marco de proceso adaptable a las necesidades y características del *software*.

1.4 Herramientas y Tecnologías Actuales

Para desempeñar un proyecto es necesario investigar las tecnologías y herramientas que se pueden utilizar, caracterizar a cada una, para así poder escoger basándose en las ventajas de su aplicación las que se utilizarán, teniendo en cuenta las características del entorno donde se desplegará el sistema a desarrollar. A continuación se define el conjunto de tecnologías y herramientas a utilizar.

1.4.1 Lenguaje de Modelado

En la construcción de un *software* los lenguajes y notación de modelado juegan un papel primordial. Mediante la utilización de los mismos se pueden crear distintos modelos y diagramas que permitan un mayor entendimiento de cómo debe funcionar el sistema por parte de todo el equipo de desarrollo del *software*. Algunos de los lenguajes que permiten el modelado de procesos de una entidad son: *Unified Modeling Language* (UML) y *Business Process Modeling Notation* (BPMN).

Lenguaje de Modelado: UML

UML se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Dicho lenguaje es una notación unificada con la que se permite lograr un entendimiento que propicie el intercambio entre los usuarios y los desarrolladores. Se ha convertido en un estándar de la industria del *software*. El UML estándar está compuesto por tres partes: elementos de construcción (tales como clases, objetos y mensajes), relaciones entre los bloques tales como: asociación, generalización y diagramas.

Los principales beneficios de UML son:

- ✓ Mejores tiempos totales de desarrollo.
- ✓ Permite la modelación de sistemas utilizando conceptos orientados a objetos.
- ✓ Establecer conceptos y artefactos ejecutables.

- ✓ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ✓ Mejor soporte a la planeación y al control de proyectos.

BPMN

Business Process Modeling Notation (Notación de Modelado de Procesos de Negocio) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. Proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. (BIZAGI, 2009)

Selección del lenguaje de modelado a utilizar

Después de realizar el estudio de algunos de los lenguajes de modelado existentes se decidió utilizar UML por ser un lenguaje de uso universal, consolidado, fácil de aprender y usar. Es una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático y prácticamente todas las herramientas CASE y de desarrollo lo han adaptado como lenguaje de modelado. Con el uso de UML se obtiene una notable efectividad y productividad en labores de diseño arquitectónico y mantenimiento.

1.4.2 Herramientas CASE

Las Herramientas CASE (*Computer Aided Software Engineering*) (Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática,

documentación o detección de errores. A continuación se identifican algunas de las herramientas CASE más empleadas a nivel internacional.

Visual Paradigm for UML

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

Visual Paradigm entre sus principales ventajas tiene su disponibilidad de integrarse con los principales entornos de desarrollo integrados, la total compatibilidad entre sus ediciones y su facilidad para instalarse y actualizarse. Además, cuenta con una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un *software*, lo cual garantiza la calidad del producto final.

Rational Rose

Rational Rose Data Modeler es una herramienta de modelado visual que posibilita que los diseñadores de base de datos, analistas, arquitectos, desarrolladores y todos los demás miembros del equipo de desarrollo trabajen juntos, capturando y compartiendo los requerimientos de negocio, dándole seguimiento a medida que cambian a través del proceso, es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML (INNOVA, 2009). Permite la realización de ingeniería inversa y es capaz de generar documentación automáticamente.

Selección de la herramienta CASE

Teniendo en cuenta las características antes expuestas de las herramientas analizadas y mediante un profundo estudio de las mismas, se llega a la conclusión

que se utilizará como herramienta de desarrollo de *software* Visual Paradigm para UML por soportar múltiples plataformas y permitir un diseño de mejores aplicaciones a un menor costo. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas, además de su robustez, usabilidad y portabilidad. Esta herramienta es de gran utilidad para el proyecto por su nivel de integración con el entorno de desarrollo y por la posibilidad de que ambos se ejecuten sobre la plataforma libre GNU/Linux, sistema operativo sobre el cual está montado el entorno de desarrollo del proyecto. Aunque Visual Paradigm es una herramienta privativa la universidad posee las licencias necesarias para su uso.

1.4.3 Tecnologías del lado del servidor

Lenguajes de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Son herramientas que permiten crear programas. Además, facilitan la tarea de programación, porque disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

Lenguajes del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a BD, conexiones en red y otras tareas para crear la vista final que se enviará al cliente, el cual solamente recibe una página con el código HTML³. Como la página resultante contiene únicamente código HTML, es compatible con la mayoría de los navegadores. Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos, que se

³ Hyper Text Markup Language o Lenguaje de Marcado de Hipertexto.

destacan por ser los más utilizados por los programadores como son PERL, ASP, PHP, Java y JSP.

Java

Java es un lenguaje orientado a objeto. Fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas web. En Java se pueden realizar los applets, que son aplicaciones especiales, los cuales se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor web, por lo general los applets son programas pequeños y de propósitos específicos.

A continuación se exponen algunas características de Java

- ✓ El polimorfismo y la encapsulación de datos son características intrínsecas.
- ✓ Incorpora un recolector automático de basura que impide incurrir en un error por inexperiencia en el manejo directo de memoria.
- ✓ Un programa creado en Java puede ser ejecutado en cualquier sistema operativo.
- ✓ Los programas escritos en Java pueden ejecutarse en cualquier arquitectura ya que genera un archivo de arquitectura neutral con códigos que no dependen de algún tipo de procesador específico. El intérprete de Java toma este archivo y lo convierte en código de máquina para la arquitectura donde se implemente.

La programación en Java permite el desarrollo de aplicaciones bajo el esquema de cliente servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente y de esta forma logra distribuir el trabajo a realizar. (PROGRAMACIÓN, 2009)

PHP

PHP⁴, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de las características más poderosas del PHP es el

⁴ HyperText Preprocessor

soporte nativo incluido para una gran cantidad de base de datos. Entre las bases de datos soportadas se encuentran: MySQL, Oracle, PostgreSQL, MSSQL Server, DBM, entre otras.

Principales ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Posee una de las más grandes comunidades en Internet.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Selección del lenguaje de programación

Partiendo de las características de los lenguajes de programación antes abordados se decide emplear Java para el desarrollo del sistema debido a que el mismo es un lenguaje de programación multiplataforma, implementa en diferentes niveles un sistema seguro de validación, logrando la seguridad de acceso a las aplicaciones web, además posee un amplio conjunto de bibliotecas estándar para trabajar con colecciones; estructuras de datos; ficheros; acceso a base de datos e interfaces gráficas, simplifica varios aspectos a la hora de programar porque realiza la gestión de memoria de forma automática, provee una sintaxis simplificada, es factible para el desarrollo de páginas web dinámicas y además es un lenguaje robusto que brinda la posibilidad de realizar verificaciones en busca de problemas en tiempo de compilación y de ejecución.

1.4.4 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE por sus siglas en inglés, *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. (OVIEDO, 2009)

Eclipse

Eclipse es una plataforma de *software* de código abierto y que ha sido muy utilizada para desarrollar IDEs, pero del mismo modo se puede usar para otros tipos de aplicaciones cliente. Lo mejor de Eclipse es que tiene una gran comunidad de usuarios extendiendo constantemente las aplicaciones. (FUNCION.COM, 2007)

Eclipse fue desarrollado originalmente por IBM Canadá. Actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como código abierto.

Netbeans

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en el 2000 y continúa siendo su patrocinador principal. NetBeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso.

NetBeans es poderoso, extensible, abierto, gratis, y además permite operar un conjunto de prácticas. Entre otras de sus características más relevantes se evidencia las mejoras en el editor de código, ya que completa el código más inteligentemente y resaltado. Su instalación y actualización es mucho más simple que otros IDE' s de desarrollo. Esta poderosa herramienta desarrolla aplicaciones colaborativamente, da la opción para la mensajería instantánea y comparte el entorno de desarrollo completo. (NETBEANS, 2010)

Selección del entorno de desarrollo integrado

A partir del análisis de los IDE que se identificaron, se propone utilizar NetBeans como entorno de desarrollo de la aplicación que se desea desarrollar debido a sus características de flexibilidad entre plataformas y su compatibilidad con el lenguaje de programación seleccionado para la implementación del sistema. NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web y control de versiones.

1.4.5 Frameworks de desarrollo

Un *framework* es una estructura de soporte definida mediante la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.

Spring

Spring es un *framework* de aplicación desarrollado por la compañía Interface, para aplicaciones escritas en el lenguaje de programación Java. Es ligero, ya que no es una aplicación que requiera de muchos recursos para su ejecución. Es un framework de código abierto, por su diseño ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fácil de usar.

Ventajas

- ✓ Proporciona la posibilidad de integrarse con otras herramientas e inclusive otros *frameworks* con el fin de obtener los beneficios que el desarrollador desea de cada una de ellas.
- ✓ Permite desarrollar aplicaciones flexibles con un bajo acoplamiento y altamente cohesivas.

- ✓ Posee una forma modular permitiendo habilitar/deshabilitar las características a utilizar según se requiera.

Symphony

Symphony es un *framework* diseñado para optimizar, gracias a sus características, el desarrollo de aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symphony posee las siguientes características:

- ✓ Fácil de instalar y configurar en la mayoría de las plataformas.
- ✓ Independiente del sistema gestor de base de datos.
- ✓ Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue las mejores prácticas y patrones de diseño para la web.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Jwebsocket

Es una tecnología orientada al desarrollo de aplicaciones basadas en websockets que gocen de altos niveles de velocidad, escalabilidad y seguridad. Jwebsocket le permite crear innovadoras aplicaciones HTML5. Los Websockets reemplazan XHR (Ajax) y Comet por una nueva tecnología de comunicación TCP basada en sockets ultra rápida para la web. La comunicación basada en websockets garantiza menos sobrecarga en la red en el orden de 400 veces y solamente 1/3 de la latencia en la red. La solución se compone en:

- ✓ Jwebsocket server: Un servidor desarrollado en Java para llamadas cliente a servidor (C2S), servidor cliente (S2C) y llamadas cliente a cliente (C2C).
- ✓ Jwebsocket clients: Clientes JavaScript y Java (SE y ME) para interacciones con el servidor jWebSocket proporcionando altos niveles de abstracción.
- ✓ Jwebsocket FlashBridge: Un adaptador basado en flash para navegadores que no son compatibles con HTML5, sin embargo requiere el plug-in the flash instalado en el navegador. (SANTAMARÍA, 2011)

JUnit

Es un *framework* para automatizar las pruebas unitarias de aplicaciones Java. Se utiliza en la fase de desarrollo mediante un conjunto de clases que permite 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.(PÉREZ, 2011). Además, permite controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple los requisitos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

Selección del *framework* de desarrollo

Partiendo de las características de los *frameworks* de desarrollo mencionados anteriormente se ha decidido utilizar los frameworks Jwebsocket, Spring y JUnit. El *framework* Jwebsocket ofrece una amplia gama de funcionalidades desde el intercambio de bajo nivel basado en Tokens hasta la sincronización de interfaz gráfica de usuario. Con Jwebsocket se puede programar rápidamente una solución independiente con un mínimo de instalación y configuración reutilizando todo el vasto cúmulo de tecnologías, librerías y aplicaciones existentes para la tecnología Java, constituyendo así una sólida base para la creación de las más diversas aplicaciones web basadas en websockets.

Spring brinda una estructura más sólida y consistente, posibilitando que cualquier persona que posea los conocimientos necesarios pueda entender con mayor facilidad la aplicación, además es portable y su mantenimiento y actualización son relativamente sencillos. En general, Spring brinda amplias funcionalidades en los aspectos de simplificación, seguridad, reutilización del código, consistencia y estructuración.

Por otro parte JUnit es simple, fácil de usar y de configurar. Está diseñado para reportar fallos o éxitos sobre código sin necesidad de interpretar el reporte. Ejecuta casos de prueba contra un conjunto de objetos y provee formas de inicializar y configurar cada caso.

1.4.6 Herramientas para el 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. (SERRADILLA, 2000)

SVN: Es un sistema de control de versiones, que mantiene los registros de todos los cambios que se han realizado a los archivos de un *software*, lo que permite el trabajo de distintos desarrolladores en un mismo proyecto. Esta herramienta es muy utilizada por los programadores de *software* libre. (SALVATORI, 2007)

RapidSVN: Es un cliente gráfico que permite manipular los repositorios de Subversión. Además, es una de las alternativas más conocidas para los sistemas GNU/Linux, muy intuitivos y fáciles de utilizar.

1.4.7 Herramienta para la gestión de proyectos java

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en el 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

Es un proceso de aplicación de patrones a la infraestructura de construcción de proyectos (Procesos de compilación, empaquetado, despliegue, testing) orientada a alcanzar una gestión coherente de los proyectos *software*. Una de las grandes ventajas de Maven son los repositorios (almacenes) de ficheros jar que se crea, con los cuales puedes incluir dependencias que posteriormente serán usadas en la construcción del proyecto.

Conclusiones

En este capítulo se realizó un estudio del estado del arte de varios sistemas de gestión de la información desarrollados tanto a nivel internacional como a nivel nacional vinculados al campo de acción. Se profundizó en el estudio y fundamentación de las metodologías de desarrollo de software más usadas, se estudiaron las tecnologías existentes del lado del servidor, y se analizaron herramientas que dan soporte al proceso de desarrollo de software. Después de tener estos elementos se arribó a las siguientes conclusiones:

Como metodología de desarrollo se decidió utilizar SXP. La Herramienta Case utilizada fue Visual Paradigm utilizando como lenguaje de modelado UML en su versión 6.4, como lenguaje de programación del lado del servidor se determinó utilizar Java, utilizando como entorno de desarrollo integrado Netbeans en su versión 7.0.1, para la gestión de los proyectos en java se utilizó Maven versión 2.2.1 ,como *frameworks* de desarrollo se utilizó Jwebsocket en su versión 1.0

Spring en su versión 3.0 y JUnit en su versión 4.5 y para el control de versiones; Subversion (SVN) y RapidSVN en su versión 0.12.0 . El avance de los medios enfocados a la Web se encuentra en un desarrollo vertiginoso y acelerado lo que permitió que se haya podido seleccionar de una gran gama de productos los más adaptables al módulo que se desea desarrollar.

CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En este capítulo se definen las principales características del sistema. Se especifican los principales artefactos generados como es el caso de las historias de usuario con sus correspondientes tareas de ingeniería, el plan de *release*, además de los requisitos funcionales y no funcionales reflejados en la lista de reserva del producto, el modelamiento del dominio, la elaboración del diagrama de componentes, el diseño con metáforas y la descripción de la arquitectura empleada en el desarrollo de la solución.

2.1 Procesos de gestión de información a automatizar

La Dirección de Educación se encarga de controlar y monitorear toda la actividad educacional ejercida en la provincia, exceptuando la Educación Superior. La misma se encuentra dividida por varios departamentos y en conjunto procesan y gestionan toda la información recibida de los municipios. Cada departamento se encarga de manejar distintos tipos de información tales como:

- ✓ Departamentos de las Enseñanzas: Dichos departamentos se encargan de gestionar toda la información relacionada con todos los niveles educacionales ya sea la enseñanza especial, primaria, adultos, preuniversitaria y secundaria básica.
- ✓ Departamento de Secretaría: El mismo se encuentra dividido a su vez en distintas áreas :
 - ✓ Salud Escolar: Área encargada de procesar toda la información referente a distintas actividades ya sean Talleres de Educación Sexual para niños, adolescentes y jóvenes, Capacitación de los trabajadores docentes y no docentes con respecto a la enfermedad del VIH-SIDA, entre otras, todas realizadas con el fin de promover, conservar,

mejorar o restaurar el estado de salud de los estudiantes en los distintos centros escolares y prevenir las enfermedades.

- ✓ Tecnologías Educativas: Área encargada de llevar el control de toda la tecnología utilizada por cada tipo de enseñanza.
- ✓ Capacitación y Superación: Área encargada de gestionar la información relacionada con la capacitación y superación de los trabajadores docentes y no docentes.
- ✓ Departamento de Planeación y Estadística: El departamento de planeación y estadística contiene a su vez distintas áreas:
 - ✓ Área de Inversión: Encargada de manejar el presupuesto monetario destinado a la construcción y reparación de obras educacionales tales como: escuelas, círculos infantiles, secundarias, primarias, y preuniversitarios de todos los municipios.
 - ✓ Área de Recursos Humanos: Encargada de manejar la información correspondiente al personal docente y no docente en cuanto al éxodo y jubilación, altas y bajas, desglose de la cobertura docente, desvinculados, licencias por maternidad y reincorporados.
 - ✓ Área de Planeación y Estadística: Encargada de manejar estadísticamente y de forma general, informaciones de todos los departamentos de enseñanzas, ya sean los resultados de las evaluaciones docentes, las matrículas estudiantiles, la continuidad de estudios de los estudiantes de sexto, noveno y doce grado, con el objetivo de realizar reportes estadísticos.

2.2 Propuesta de solución

Desarrollar un módulo que gestione la información procesada en la Dirección de Educación de la Administración Provincial de Artemisa que contribuya a elevar los niveles de eficacia y confiabilidad en el proceso de gestión de información. El módulo cuenta con las siguientes funcionalidades.

- ✓ Inserción de los datos.
- ✓ Modificación de los datos.
- ✓ Obtención de los datos.
- ✓ Eliminación de los datos.
- ✓ Generación de reportes.

2.3 Planificación del Proyecto por Roles

Algunas de las personas involucradas en un proyecto de *software* pueden ser los diseñadores, analistas y programadores. Un trabajador es el papel que un individuo puede desempeñar durante el desarrollo del *software*. El término rol se refiere a los distintos papeles que cumple un determinado trabajador. **(Ver Anexo .2)**

2.4 Modelo de Dominio

Dentro de las actividades más importantes definidas en la metodología SXP se encuentra la definición del Modelo de Historias de Usuario del Negocio, en el cual se hace una detallada descripción del negocio en cuestión. Pero si dicho negocio no está bien definido entre los clientes y los ejecutores del proyecto; entonces es generado el Modelo de Dominio. La siguiente figura muestra el diagrama de dominio que se propone para el sistema

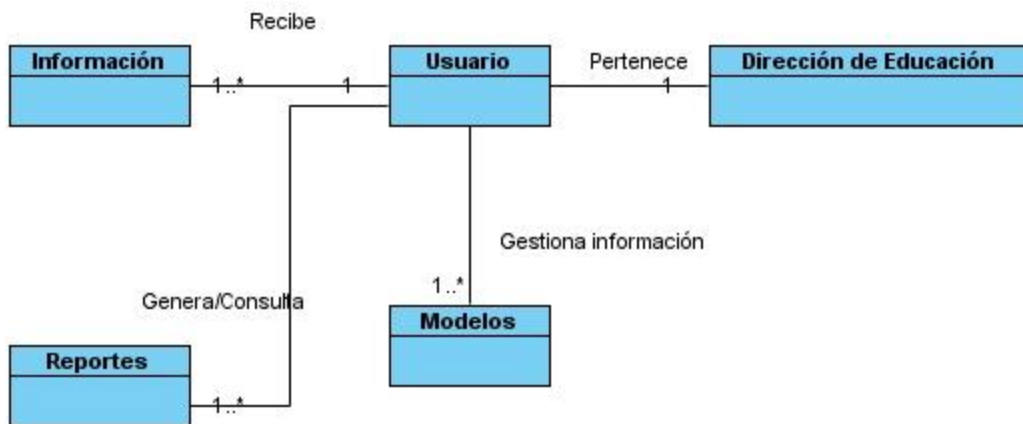


Figura 1: Modelo de Dominio

Conceptos del modelo de dominio:

Clase Información: Representa la información mensual, trimestral, semestral y anual recibida por la Dirección procedente de los municipios.

Clase Usuario: Persona encargada dentro de la Dirección de gestionar la información que recibe de los distintos centros educacionales ubicados en los diferentes municipios de la provincia, además de diseñar los modelos de recogida de datos, con una estructura y un formato previamente establecidos.

Clase Dirección de Educación: Dirección donde pertenece el usuario

Clase Modelos: Representa los modelos correspondientes a los departamentos de la Dirección.

Clase Reportes: Representa el reporte correspondiente al modelo o los modelos que ha gestionado el usuario a partir de la información recibida.

2.5 Lista de Reserva del Producto (LRP)

La Lista de reserva del producto (LRP), es el primer artefacto generado en la etapa de Captura de requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que solo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. **(Ver Anexo.3)**

2.6 Historias de Usuario y Tareas de Ingeniería

Cuando se hace uso de la metodología SXP, para especificar los casos de uso como correspondería en RUP, se hace mediante historias de usuarios (HU), descriptoras de las tareas que el sistema debe hacer, cuestión que depende en

gran medida de las especificaciones realizadas por el cliente. Se escriben con un lenguaje natural y con palabras concisas para no exceder su tamaño en unas pocas líneas de texto.

Para facilitar la definición de cada una de las actividades que describen las HU se le asignan a las mismas una o más tareas de ingeniería. Estas permiten conocer el tiempo que dura cada una de las actividades asociadas a las historias de usuario, así como el responsable de implementar dicha tarea, facilitando la realización de una buena estimación del tiempo de implementación. La siguiente tabla muestra información de una historia de usuario asociada con algunas tareas de ingeniería

Tabla 1: Historia de Usuario y Tareas de Ingeniería

| Historia de Usuario | |
|---|---|
| Número: HU_1 | Nombre Historia de Usuario: Gestionar información estadística de las enseñanzas. |
| Modificación de Historia de Usuario Número: ninguna | |
| Usuario: Johanna Carbonell Walfrido Báez | Iteración Asignada: 2 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Muy Alto | Puntos Reales: 1 |
| Descripción: La presente historia de usuario tiene como objetivo insertar, modificar, eliminar y buscar información sobre algunos aspectos generales de las enseñanzas teniendo en cuenta algunos elementos significativos de la educación especial tales como: cantidad de internos, seminternos, continuantes, nuevo ingreso, información sobre los estudiantes según las distintas modalidades , proyección del egreso, indicadores de eficiencia, por otra parte en la primaria se tienen en cuenta diversos parámetros relacionados con las escuelas puras, mixtas, urbanas, internas y rurales, además de datos del centro de diagnóstico y orientación los cuales reflejan análisis estadísticos de todas las | |

| | |
|--|---|
| enseñanzas. | |
| Observaciones: Para que esto sea posible el usuario debe estar logueado previamente para de esta forma manejar la información. | |
| Prototipo de interface: <i>Ver Anexo 4</i> | |
| Tarea de Ingeniería | |
| Número Tarea: 1.1 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Investigar cómo se desarrollan los diferentes procesos llevados a cabo en la Dirección de Educación para implementar las funcionalidades de insertar, modificar, eliminar y buscar información. | |
| Tipo de Tarea : Investigación | Puntos Estimados: 1/6 |
| Fecha Inicio: 30/01/12 | Fecha Fin: 31/01/12 |
| Programador Responsable: Johanna Carbonell Parra y Walfrido Cabrera Báez | |
| Descripción: Efectuar un estudio sobre el proceso de cómo realizar las diferentes funcionalidades de insertar, modificar, buscar y eliminar información en la Dirección de Educación. | |
| Tarea de Ingeniería | |
| Número Tarea: 1.2 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Implementación de la funcionalidad insertar situación estadística de la enseñanza especial. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1/6 |
| Fecha Inicio: 30/01/12 | Fecha Fin: 31/01/12 |
| Programador Responsable: Johanna Carbonell Parra y Walfrido Cabrera Báez | |

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

2.7 Plan de Release

El Plan de Release es definido para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente y se tiene en cuenta la prioridad que definió el cliente para ubicar las historias de usuario. Como resultado de la priorización de historias se llegó a la siguiente planificación:

| Release | Descripción de la iteración | Orden de la HU a implementar | Duración total |
|-------------|---|---|----------------|
| Iteración 2 | En esta iteración se desarrollarán las historias de usuario que tienen prioridad muy alta. | HU_1,HU_2,HU_3,HU_4,HU_5 HU_6. | 4 semanas |
| Iteración 3 | En esta iteración se desarrollarán las historias de usuario de alta prioridad y se irá integrando con las ya realizadas. | HU_7, HU_8, HU_9. | 4 semanas |
| Iteración 4 | En esta iteración se desarrollarán las historias de usuario de prioridad media y se integrarán con las historias de usuario ya implementadas. | HU_10,HU_11,HU_12,HU_13 HU_14,HU_15,HU_16,HU_17, HU_18, HU_19, HU_20, HU_21, HU_22, HU_23, HU_24, HU_25, HU_26. | 4 semanas |

Figura 2: Plan de Release

2.8 Arquitectura de Software

Lo que se conoce como arquitectura en capas es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio, capa de acceso a datos y mecanismos de almacenamiento. Para una mayor comprensión de la misma se explica cómo está constituida en la siguiente figura:

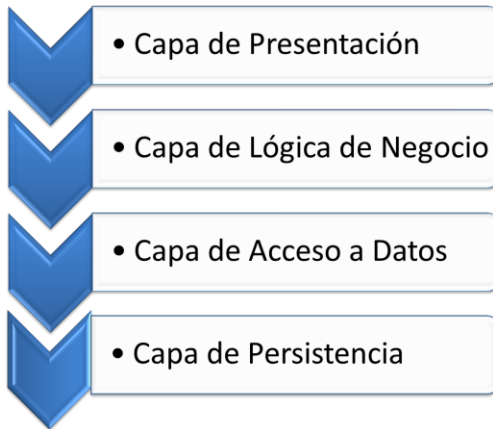


Figura 3: Representación de la Arquitectura N-Capas

Capa de presentación: Es la capa con la que interactúa el usuario directamente. Se comunica únicamente con la capa de lógica de negocio. Desde la filosofía de arquitectura en capas, esto significa que la capa de lógica de negocios presenta una "interfaz" para brindar servicios a la capa de presentación.

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

En este caso para el desarrollo del módulo se hizo especial atención sobre la capa de lógica de negocio y la capa de acceso a datos.

- ✓ **Capa Lógica del Negocio:** Se denomina capa de lógica de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse.

- ✓ Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados; y con la capa de datos para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- ✓ Es la encargada de la implementación de los eventos, los plugins y los servicios. Los servicios proporcionan todas las funcionalidades del módulo, implementadas a partir de los procesos de gestión de información identificados. Esto es posible gracias a la utilización del dao genérico, mediante el cual se realizan consultas a la base de datos.
- ✓ **Capa Acceso a Datos:** Capa que sirve como puente entre la capa lógica de negocio y el proveedor de datos. Esta capa pretende encapsular las especificidades del proveedor de datos (tales como SQL, archivos XML, texto, hojas electrónicas) a la siguiente capa, para que si cambia el proveedor de datos solo se necesite cambiar el mismo en una sola capa.

2.9 Patrones de Diseño

Los patrones de diseño permiten dar solución a problemas comunes que se presentan dentro del ciclo de vida del proceso de desarrollo de software, describiendo dichos problemas de forma sencilla e indicando los pasos necesarios para resolverlos. Se categorizan en tres grupos fundamentales, los creacionales, los estructurales y los de comportamiento. A continuación se describen algunos de los patrones utilizados en el desarrollo del Módulo de la Dirección de Educación de la AP de Artemisa.

Patrones creacionales: Abstraen la forma en la que se crean los objetos, permitiendo tratar las clases de forma genérica, dejando para más tarde la decisión de qué clases crear o cómo crearlas.

Dentro de esta clasificación, los patrones utilizados fueron los siguientes:

- ✓ **Singleton** (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

El uso de la inyección de dependencias en el *framework* Spring está dirigido a potenciar este patrón, con el uso del IoC de Spring. Con la creación de este bean se puede instanciar la clase **SpecialStatisticalSituationPlugIn** definiendo como identificador del plugIn "id", el cual será posteriormente utilizado por el cliente para generar el plugIn y llamar los eventos que este utiliza.

La property `service` del plugIn hace referencia al bean **SpecialStatisticalSituationService** el cual no es más que una instancia de la clase `SpecialStatisticalSituationService`. La siguiente figura muestra un ejemplo de lo planteado anteriormente

```
<bean id="SpecialStatisticalSituationPlugIn" class="cu.uci.hab.sigob.directorates.
education_culture_and_sport.education.plugIns.plugInSpecialDepartment.
plugInSpecialStatisticalSituation.SpecialStatisticalSituationPlugIn" parent="AbstractPlugIn">
  <property name="id" value="SpecialStatisticalSituationPlugIn" />
  <property name="service" ref="SpecialStatisticalSituationServiceImpl"/>
```

Figura 4: Representación del uso del IoC de Spring

- ✓ **Factory Method** (Método de fabricación): Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear.

Bean Factory es uno de los componentes principales del núcleo de Spring. Es una implementación del patrón de fabricación. Bean Factory es de propósito general, ya que puede crear diferentes tipos de beans. Los beans pueden ser llamados por nombre y se encargan de manejar las relaciones entre objetos. A continuación se muestra un ejemplo de un bean:

```

<bean parent="AbstractEventDefinition">
  <property name="id" value="SpecialStatisticalSituationPlugIn.add" />
  <property name="ns" value="cu.uci.hab.sigob.directorates.education_culture_and_sport.
education.plugin.pluginSpecialDepartment.pluginSpecialStatisticalSituation.events.
AddSpecialStatisticalSituationEvent" />
  <property name="responseRequired" value="true" />
  <property name="responseToOwnerConnector" value="true" />
  <!--property name="securityEnabled" value="true" /-->
  <property name="incomingArgsValidation">
    <set>
      <bean class="org.jwebsocket.eventmodel.filter.validator.Argument" >
        <property name="name" value="date" />
        <property name="type" value="string" />
        <property name="optional" value="false" />
      </bean>
    </set>
  </property>
</bean>

```

Figura 5: Ejemplo de un Bean

Como se muestra en el ejemplo existe un elemento bean, el cual puede contener dentro uno o más elementos de tipo bean, para cada una de las diferentes funciones que se requieran.

Patrones de Comportamiento: Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. Dentro de esta clasificación se encuentra el patrón Observer.

El patrón Observer define la dependencia de uno-a-muchos entre objetos, de modo que cuando el objeto cambia de estado, todos sus dependientes son notificados y actualizados automáticamente.

Los eventos de la aplicación heredan de la clase **C2SEvent** importado de la librería **org.jwebsocket.eventmodel.event.C2SEvent** de jwebsocket. Los eventos C2SEvent/S2CEvent son definiciones orientadas a objetos usado en el EventModel para el tratamiento o manejo de las variables de entrada y de salida desde el cliente o para el cliente. A continuación se muestra un ejemplo:

```
public class DeleteSpecialStatisticalSituationEvent extends C2SEvent
```

Otra observación es el uso de anotaciones como la que se encuentra encima de las properties de los eventos **@ImportFromToken** mediante la cual se logra la asignación automática de campos de los eventos, además de copiar o mover el valor de los tokens de entrada a los campos delimitados en los eventos. Esta anotación debe estar definida para cada uno de los atributos del evento. A continuación se muestra un ejemplo:

@ImportFromToken

```
public void setIdSpecialStatistical(Integer idSpecialStatistical) {  
  
this.idSpecialStatistical = idSpecialStatistical;  
  
}
```

2.10 Diagrama con Metáfora

El Diseño con metáforas es el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto; lo cual genera el artefacto conocido como Modelo de Diseño, que a su vez está compuesto por un diagrama de paquetes, el cual expone dicho diseño. A continuación se representa el diagrama de paquetes para el sistema que se propone.

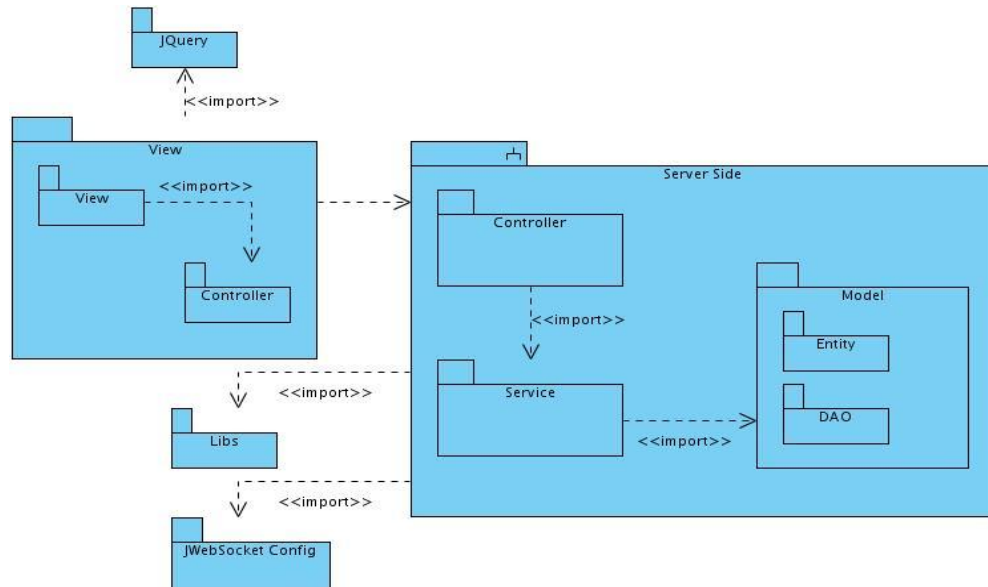


Figura 6 Diagrama de Paquetes

Descripción

- ✓ El paquete View (vista) muestra cómo se maneja la información en la capa cliente para tratar las solicitudes que realizan los clientes.
- ✓ En el paquete Controlador se encuentran los Plugins y los servicios donde se manejan y procesan los datos de la aplicación de cada uno de los requisitos funcionales observados en el negocio y las relaciones entre ellos.
- ✓ El paquete Modelo tiene las principales entidades donde se encuentra almacenada la información que será gestionada.

2.11 Diagrama de Componentes

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

A continuación se presenta el diagrama de componentes para el módulo que se propone.

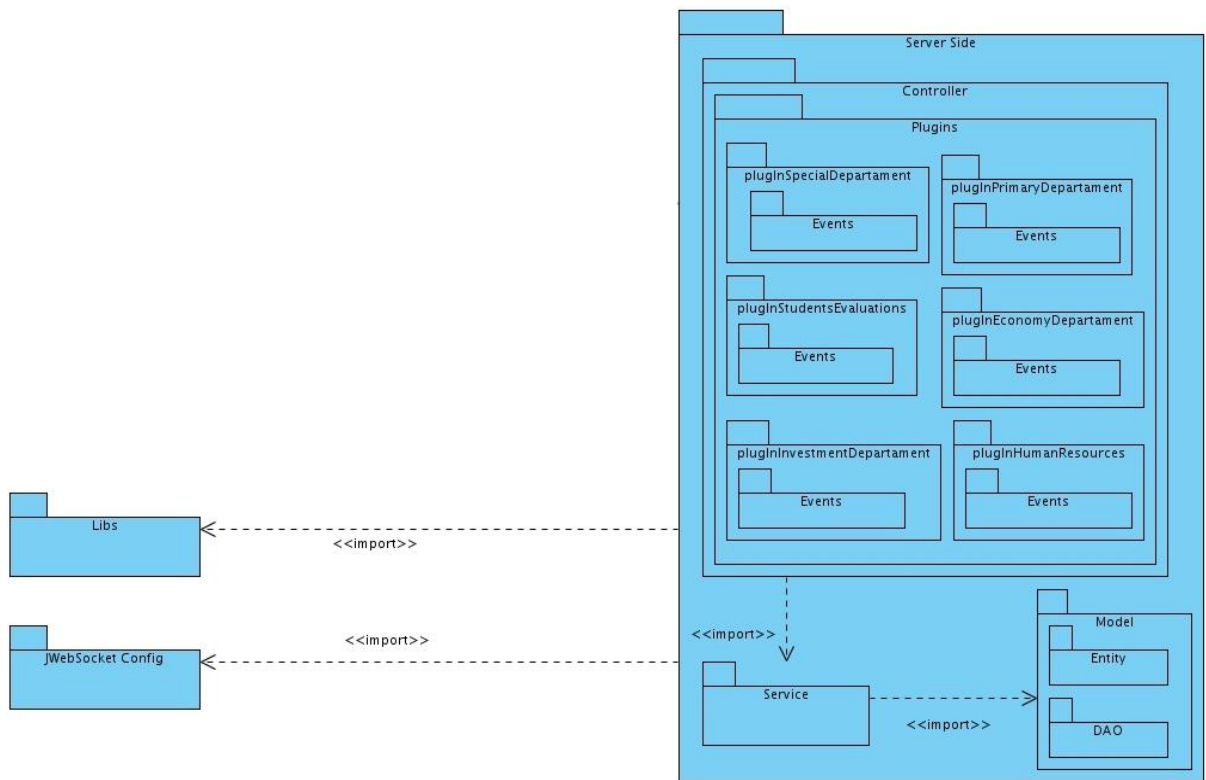


Figura 7 Diagrama de Componentes

Conclusiones del Capítulo

Con la realización del presente capítulo se definieron los principales requisitos del sistema y la duración general del desarrollo del mismo. Se dividió la aplicación en diferentes iteraciones donde en cada una de ellas se da terminación a un grupo de procesos específicos del Módulo. Se modeló de manera general e individual el diseño de la aplicación y las relaciones existentes entre los componentes que la integran obteniéndose así el Módulo para la Dirección de Educación quedando solamente realizarle las pruebas pertinentes para verificar su correcto funcionamiento.

CAPÍTULO 3: ADQUISICIÓN Y VALIDACIÓN DE LOS RESULTADOS DEL SISTEMA.

Introducción

En el presente capítulo queda documentado el estándar de código utilizado, la programación realizada a partir de los requerimientos y los diagramas de diseño elaborados. Se expone la validación de las funcionalidades realizadas al sistema mediante los casos de pruebas de caja blanca siendo estos necesarios para avanzar hacia la próxima iteración. Se dan a conocer los resultados obtenidos, el diagrama de despliegue elaborado, las funcionalidades obtenidas, además del aporte social y económico de la investigación.

3.1 Estándares de Códificación

Los estándares de codificación son pautas de programación que no están enfocados a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Establecer el estándar de código 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 Módulo de la Dirección de Educación de la AP de Artemisa se utilizaron varias reglas a seguir:

- ✓ 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**. Todas las palabras que componen a dichos identificadores empezarán con mayúscula. A continuación se muestra un ejemplos del uso de esta variante: `class AddSpecialStatisticalSituationEvent`

- ✓ Para los métodos y variables se hizo uso de la variante **LowerCamelCase**. Empiezan con minúsculas y si estos identificadores están compuestos por varias palabras las siguientes empezarán con mayúscula. A continuación se muestra ejemplos del uso de esta variante:

Integer semiInternal;

```
public String getDate()
```

- ✓ 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. A continuación se muestra un ejemplo

```
/**  
  
 * Class Description: Add event Special Statistical Situation  
  
 * @author yohana  
  
 * @version $1.0$  
  
 */
```

3.2 Implementación del lado del servidor

El servidor es el encargado de responder a las peticiones solicitadas por el cliente. Mediante el uso de eventos, plugins y servicios se obtienen los datos enviados por el cliente para ser procesados en el servidor.

3.2.1 Creación de los eventos de la aplicación

La programación orientada a eventos es muy utilizada en la construcción de interfaces gráficas de usuario.

Los 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. A través de los mismos se obtienen los datos ingresados por el cliente ya sea para adicionar, modificar, buscar, eliminar o generar reportes de la información referente a los distintos modelos correspondientes a la Dirección de Educación. A continuación se presenta la implementación de uno de los eventos de la aplicación

Evento DeleteSpecialStatisticalSituation: Este evento permite ejecutar la acción de eliminar una situación estadística de la enseñanza especial a través del identificador de la misma. La siguiente figura muestra la implementación del evento

```
/**
 * Class description: Delete event Special Statistical Situation
 * @author yohana
 * @version $1.0$
 */
public class DeleteSpecialStatisticalSituationEvent extends C2SEvent {

    private Integer idSpecialStatistical; // id

    /**
     * @return the idSpecialStatistical
     */
    public Integer getIdSpecialStatistical() {
        return idSpecialStatistical;
    }

    /**
     * @param idSpecialStatistical the idSpecialStatistical to set
     */
    @ImportFromToken
    public void setIdSpecialStatistical(Integer idSpecialStatistical) {
        this.idSpecialStatistical = idSpecialStatistical;
    }
}
```

Figura 8: Implementación del evento DeleteSpecialStatisticalSituation

3.2.1 Creación de las clases services de la aplicación.

Las clases services permiten realizar la operación solicitada por el cliente. Son las encargadas de ordenar la ejecución de consultas en la base de datos mediante el

uso del GenericDao que tiene como atributo. A continuación se muestra uno de los servicios creados de la aplicación:

Service SpecialStatisticalSituationServiceImpl: En esta clase se implementan los requisitos o funcionalidades involucradas con la situación estadística de la enseñanza especial. A continuación se muestra una tabla con algunos de los métodos implementados en la clase SpecialStatisticalSituationServiceImpl.

Tabla 2: Métodos de la clase SpecialStatisticalSituationServiceImpl

| Métodos |
|--|
| <p><u>addSpecialStatistical</u> (AddSpecialStatisticalSituationEvent event): void</p> <p>Recibe como entrada un evento y permite salvar una situación estadística en la base de datos. Verifica que la fecha ingresada por el usuario no se encuentre registrada, en caso de que exista lanza una excepción indicando que la fecha existe, en caso contrario el método se ejecuta satisfactoriamente.</p> |
| <p><u>modifySpecialStatistical</u>(ModifySpecialStatisticalSituationEvent event): void</p> <p>Recibe como entrada un evento y permite actualizar la situación estadística de la enseñanza especial en la base de datos.</p> |
| <p><u>deleteSpecialStatistical</u>(DeleteSpecialStatisticalSituationEvent event):void</p> <p>Recibe como entrada un evento y permite eliminar la situación estadística de la enseñanza especial de la base de datos.</p> |
| <p><u>findSpecialStatistical</u>(FindSpecialStatisticalSituationEvent event): List<FastMap></p> <p>Recibe como entrada un evento y permite listar las situaciones estadísticas de la enseñanza especial que se encuentren registradas en la base de datos, según el criterio de búsqueda especificado en el evento.</p> |

generateReport(ReportSpecialStatisticalSituationEvent report):List<FastMap>

Recibe como entrada un evento y permite generar el reporte de las situaciones estadísticas de la enseñanza especial que se encuentren registradas en la base de datos, según el criterio de búsqueda especificado en el evento.

3.1.3 Creación de las clases plugin de la aplicación.

Los plugins son extensiones para dar soporte personalizados (C2S/S2C) a los acontecimientos en el lado del servidor, básicamente son los listeners para escuchar o notificar a los eventos relacionados en el mismo. Cada clase plugin hereda de la clase **EventModelPlugin** del paquete de `jWebSocket.org.jwebsocket.eventmodel.plugin.EventModelPlugin` además de implementar los distintos `processEvent` que se encargarán de escuchar las notificaciones realizadas por el cliente. A continuación se muestra uno de los servicios creados de la aplicación:

Plugin SpecialStatisticalSituationPlugin: Este plugin es el encargado de manejar los distintos eventos involucrados con la situación estadística de la enseñanza especial. Esta clase hace uso del servicio `SpecialStatisticalSituationServiceImpl` definido con anterioridad. A continuación se muestra un ejemplo del plugin mencionado implementando algunos `processEvent`:

Tabla 3: Plugin SpecialStatisticalSituation

```
public class SpecialStatisticalSituationPlugin extends EventModelPlugin {  
    private SpecialStatisticalSituationServiceImpl service;  
    public void processEvent(DeleteSpecialStatisticalSituationEvent aEvent,
```

```
C2SResponseEvent aResponseEvent) throws ParseException,
    Exception { getService().deleteSpecialStatistical(aEvent);
}
public void processEvent(ListSpecialStatisticalSituationEvent aEvent,
    C2SResponseEvent aResponseEvent) throws Exception {
    aResponseEvent.getArgs().setList("list", getService().
        ListSpecialStatisticalSituationEvent());
}}
```

3.3 Casos de Prueba

Los casos de prueba tienen como objetivo validar que las operaciones del *software* cumplan con el funcionamiento esperado y permitir al cliente determinar su aceptación desde el punto de vista funcional. Permiten garantizar la entrega de un producto con calidad respondiendo a las necesidades del usuario final.

3.3.1 Pruebas de caja blanca:

Las pruebas de caja blanca tienen como objetivo asegurar que la operación interna de los métodos implementados está acorde a las especificaciones establecidas y que cada componente interno se ha probado adecuadamente. Existen varios tipos de pruebas de caja blanca:

- ✓ Prueba del camino básico.
- ✓ Pruebas unitarias.
- ✓ Pruebas de la estructura de control.
 - ✓ Prueba de condición.
 - ✓ Prueba de flujo de datos.

- ✓ Prueba de bucles

3.3.1 Pruebas Unitarias

Las pruebas unitarias son tipos de 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.

Las pruebas que se realizan al Módulo de la Dirección de Educación se desarrollaron como parte de la pruebas ejecutadas por el Procedimiento para la ejecución y documentación de pruebas unitarias en los proyectos desarrollados en java de la Facultad Regional “Mártires de Artemisa” ,mediante la utilización del *framework* JUnit. La comprobación de datos se dividió en tres iteraciones, mostrándose en cada una de ellas resultados diferentes. A continuación se muestran ejemplos de los métodos generados para probar algunas de las funcionalidades, de las iteraciones realizadas y una gráfica resumen de los resultados de las pruebas:

```
@Test
public void testAddSpecialStatistical() throws Exception {

    AddSpecialStatisticalSituationEvent addSpecialStatistical = new AddSpecialStatisticalSituationEvent();
    addSpecialStatistical.setContinuants(5);
    addSpecialStatistical.setDate(new Date("2012/5/12"));
    addSpecialStatistical.setExternal(52);
    addSpecialStatistical.setIdModel(1);
    addSpecialStatistical.setInitialRegistration(6);
    addSpecialStatistical.setInternal(2);
    addSpecialStatistical.setNewEntry(7);
    addSpecialStatistical.setSemiInternal(9);
    SpecialStatisticalSituationServiceImpI instance = new SpecialStatisticalSituationServiceImpI();
    boolean expResult = true;
    boolean result = instance.addSpecialStatistical(addSpecialStatistical);
    assertEquals(expResult, result);
}
```

Figura 9: Método AddSpecialStatistical

```

@Test
] public void testModifySpecialStatistical() throws Exception {
    System.out.println("modifySpecialStatistical");
    ModifySpecialStatisticalSituationEvent modifySpecialStatistical = new ModifySpecialStatistical();
    modifySpecialStatistical.setContinuants(23);
    modifySpecialStatistical.setDate(new Date("2012/05/05"));
    modifySpecialStatistical.setExternal(52);
    modifySpecialStatistical.setIdModel(1);
    modifySpecialStatistical.setIdSpecialStatisticalSituation(24);
    modifySpecialStatistical.setInitialRegistration(74);
    modifySpecialStatistical.setInternal(14);
    modifySpecialStatistical.setNewEntry(2300);
    modifySpecialStatistical.setSemiInternal(89);
    SpecialStatisticalSituationServiceImpl instance = new SpecialStatisticalSituationServiceImpl();
    int expectedResult = 1;
    int result = instance.modifySpecialStatistical(modifySpecialStatistical);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    System.out.println("modifySpecialStatistical");
    System.out.println(result);
}

```

Figura 10: Método ModifySpecialStatistical

```

@Test
] public void testFindSpecialStatistical() throws Exception {
    System.out.println("findSpecialStatistical");
    FindSpecialStatisticalSituationEvent findSpecialStatistical = new FindSpecialStatisticalSituationEvent();
    findSpecialStatistical.setDate(new Date("2012/05/05"));

    SpecialStatisticalSituationServiceImpl instance = new SpecialStatisticalSituationServiceImpl();
    boolean expectedResult = true;
    boolean result = instance.findSpecialStatistical(findSpecialStatistical);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    System.out.println("findSpecialStatistical");
    System.out.println(result);
}

```

Figura 11: Método FindSpecialStatistical

```

@Test
] public void testListSpecialStatistical() throws Exception {
    System.out.println("listSpecialStatistical");
    ListSpecialStatisticalSituationEvent listSpecialStatistical = new ListSpecialStatisticalSituationEvent();

    SpecialStatisticalSituationServiceImpl instance = new SpecialStatisticalSituationServiceImpl();
    boolean expectedResult = true;
    boolean result = instance.ListSpecialStatisticalSituationEvent();
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    System.out.println("listSpecialStatistical");
    System.out.println(result);
}

```

Figura 12: Método ListSpecialStatistical

```

@Test
public void testDeleteSpecialStatistical() throws Exception {
    System.out.println("deleteSpecialStatistical");
    DeleteSpecialStatisticalSituationEvent deleteSpecialStatistical = new DeleteSpecialStatistical();
    deleteSpecialStatistical.setIdSpecialStatistical(1);
    SpecialStatisticalSituationServiceServiceImpl instance = new SpecialStatisticalSituationServiceServiceImpl();
    boolean expectedResult = true;
    boolean result = instance.deleteSpecialStatistical(deleteSpecialStatistical);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    System.out.println("deleteSpecialStatistical");
    System.out.println(result);
}

```

Figura 13: Método DeleteSpecialStatistical

```

@Test
public void testReportSpecialStatistical() throws Exception {
    System.out.println("reportSpecialStatistical");
    ReportSpecialStatisticalSituationEvent reportSpecialStatistical = new ReportSpecialStatistical();
    reportSpecialStatistical.setDate(new Date("2012/05/05"));
    SpecialStatisticalSituationServiceServiceImpl instance = new SpecialStatisticalSituationServiceServiceImpl();
    boolean expectedResult = true;
    boolean result = instance.generateReport(reportSpecialStatistical);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    System.out.println("reportSpecialStatistical");
    System.out.println(result);
}

```

Figura 14: Método ReportSpecialStatistical

The screenshot shows a test runner window with two tabs: 'pruebasigobedu.SpecialStatisticalSituationServiceImplTest' and 'pruebasigobEdu'. A green progress bar indicates 66,67% completion. Below the progress bar, a summary states '4 tests passed, 2 tests caused an error.(0,314 s)'. A tree view shows the following results:

- pruebasigobedu.SpecialStatisticalSituationServiceImplTest FAILED
 - testFindSpecialStatistical passed (0,01 s)
 - testListSpecialStatistical passed (0,006 s)
 - testDeleteSpecialStatistical passed (0,004 s)
 - testReportSpecialStatistical passed (0,002 s)
 - testModifySpecialStatistical caused an ERROR: java.lang.NullPointerException
 - testAddSpecialStatistical caused an ERROR: java.lang.NullPointerException

A sidebar on the left contains icons for running, pausing, and navigating through the test results.

Figura 15: Iteración 1 de las pruebas unitarias

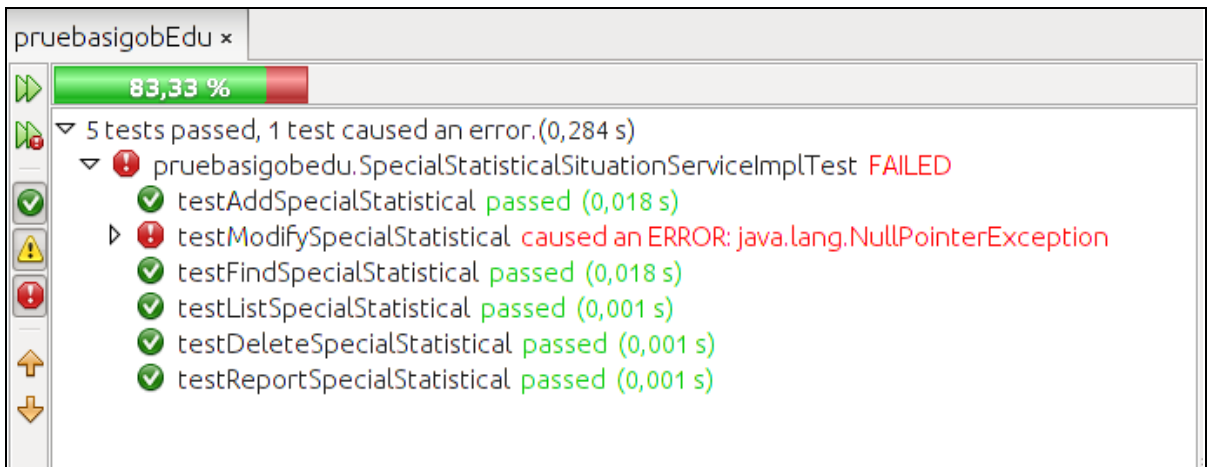


Figura 16: Iteración 2 de las pruebas unitarias

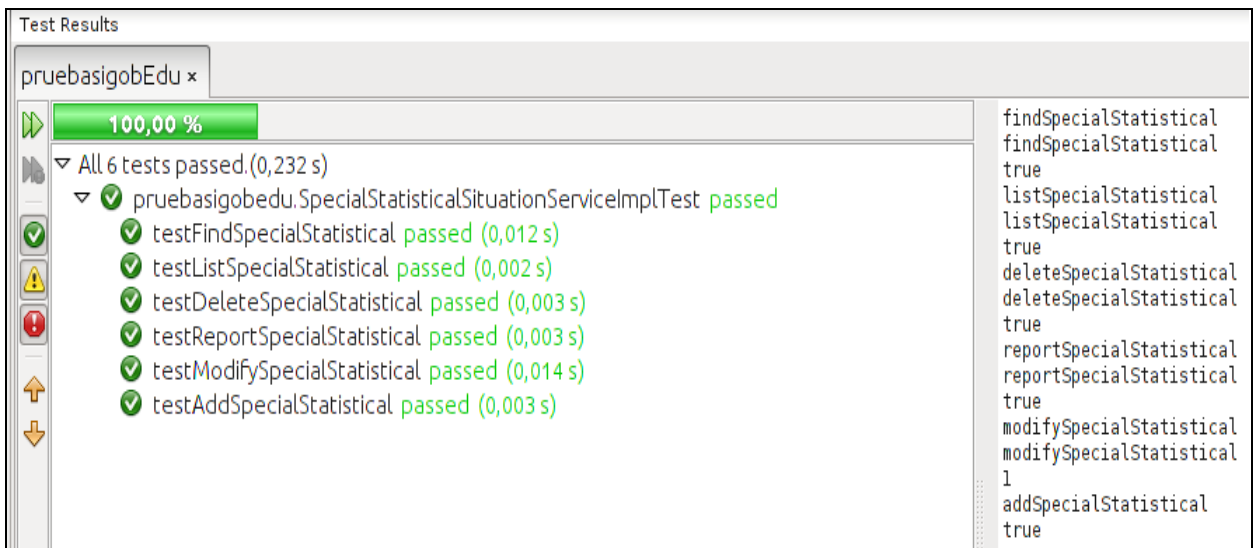


Figura 17: Iteración 3 de las pruebas unitarias

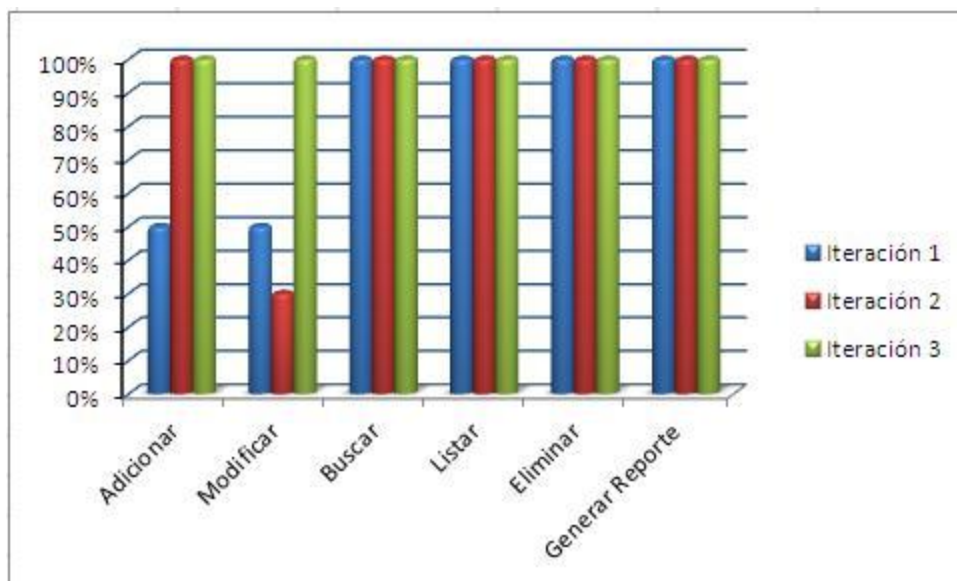


Figura 18: Gráfica Resumen del Resultado de las pruebas unitarias

3.4 Resultados Obtenidos

Como producto final del presente trabajo de diploma y teniendo en cuenta los resultados satisfactorios de los casos de pruebas realizados queda disponible el Módulo para la Dirección de Educación de la AP de Artemisa en su versión 1.0.

3.5 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. Describe la distribución física del sistema y muestra cómo están distribuidos los componentes de software entre los distintos nodos de cómputo.

Los nodos se utilizan para modelar la topología del hardware sobre el cual se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes. A continuación se presenta el diagrama de despliegue del sistema propuesto:

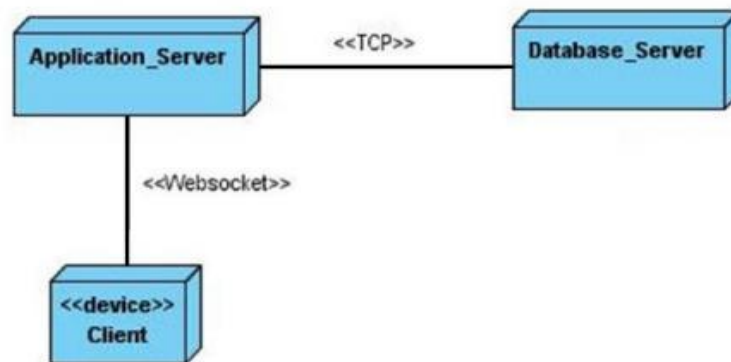


Figura 19: Diagrama de Despliegue

Descripción

- ✓ **Cliente:** Este se conecta a través del protocolo Websocket al Servidor Jwebsocket utilizando cualquier navegador que soporte dicho protocolo independientemente del sistema operativo que esté 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.

3.6 Funcionalidades Obtenidas

Entre las primeras funcionalidades que posee el Módulo para la Dirección de Educación en su versión 1.0 se pueden mencionar:

-Gestionar toda la información referente a la Enseñanza Especial ya sea insertar, modificar, buscar y eliminar la Situación Estadística, Continuidad de Estudios, Proyección y Ubicación del Tránsito, Seguimiento y Orientación , Indicadores de Eficiencia, Proyección del Egreso, Modalidades, Proyección del Egreso, Resultados de la Evaluación Profesoral, Promoción, Escuelas por Rango y Grupos de Matrículas, Personal Docente y Diagnóstico y Orientación.

-Generar reportes por cada uno de los modelos insertados con anterioridad en la base de datos.

3.7 Aporte Social y Económico

El desarrollo del Módulo para la Dirección de Educación de la AP de Artemisa posibilita la informatización de los procesos de gestión de la información desarrollados en la Dirección. La implantación del mismo trae consigo los siguientes beneficios:

- ✓ Control del recurso información y la disminución del tiempo empleado en la gestión del mismo permitiendo minimizar los riesgos a los que están expuestos
- ✓ Rapidez en la comunicación y en las búsquedas de información por parte del personal disminuyendo su carga de trabajo.
- ✓ Aumento de los niveles de eficacia y confiabilidad en el procesamiento de gestión de la información de la Dirección.
- ✓ Brinda al país un ahorro significativo, ya que tiene la ventaja económica de realizarse en Cuba, es decir no utiliza software propietario por el que haya que pagar para su uso, ni es necesario invertir dinero contratando una serie de desarrolladores para la realización del mismo.

Conclusiones del Capítulo

Al concluir el desarrollo de las pruebas al sistema se demuestra que las funcionalidades obtenidas por el Módulo, se han desarrollado de acuerdo a los requisitos establecidos en la etapa inicial del desarrollo del mismo y satisfacen las necesidades del cliente. Como resultado final de la presente investigación se obtuvo la versión 1.0 del Módulo de la Dirección de Educación de la Administración Provincial de Artemisa.

CONCLUSIONES GENERALES

- ✓ Se desarrolló el Módulo para la Dirección de Educación de la AP de Artemisa usando los estándares adecuados para su desarrollo.
- ✓ La implementación de las funcionalidades dieron respuesta a los requisitos funcionales propuestos para el desarrollo del Módulo.
- ✓ El Módulo de la Dirección de Educación de la AP de Artemisa elevó los niveles de eficacia y confiabilidad en los procesos de gestión de información desarrollados, así como la calidad del trabajo en la Dirección y finalmente la satisfacción del usuario.

RECOMENDACIONES

- ✓ Seguir optimizando las funcionalidades para mejorar la gestión de la información en la Dirección de Educación de la AP de Artemisa.
- ✓ Desplegar la aplicación en las entidades municipales para así facilitar el trabajo de los técnicos de estas instituciones.

BIBLIOGRAFÍA REFERENCIADA

ACUÑA, K. B. Selección de Metodologías de Desarrollo Para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos. En *Selección de Metodologías de Desarrollo Para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos*. 1990

BARTLE, P... *Información para la gestión y gestión de la información* Última actualización: abril 13, 2011. (Información para la Gestión y gestión de la información). Disponible en: <http://cec.vcn.bc.ca/mpfc/modules/mon-miss.htm>.

BIZAGI, C... *BPMN Business Process Modeling Notation*. Editado por publicado el: 2009 de 2009, última actualización: 2009. [Consultado el: febrero 9, 2012]. Disponible en: <http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf>.

CABALLERO, I... *Optimización del Proceso de Gestión de Información para la Mejora de la Calidad de la Información*. Editado por: publicado el: 2004 de 2004, última actualización: 2004. [Consultado el: mayo 31, 2012]. Disponible en: <http://www.grise.upm.es/rearviewmirror/conferencias/jiisic04/Papers/4.pdf>.

CARVAJAL, A. A... *Lenguaje Unificado de Modelado*. Editado por: publicado el: noviembre 4, 2008 de 2008, última actualización: noviembre 4, 2008. [Consultado el: febrero 9, 2012]. Disponible en: <http://syacomputadores.googlepages.com/LenguajeUnificadodeModeladoUML.doc>.

CHILE, U. DE. *SIGA. Sistema de Información de Gestión Académica Universidad de Chile* Última actualización: 2005. Disponible en: http://alpes.stg.uchile.cl/~srvcen01/censo/index_ie.html.

DARA, G... *S.I.G.A. Sistema Integrado de Gestión Académica*. Última actualización: 2009. Disponible en: <http://www.dara.es/siga/>.

DANTE, G. P. *Gestión de Información en la Sociedad de la Información* Última actualización: 2004. Disponible en: <http://www.abinia.org/boletin/3-1/temas.htm>.

DELGADO, M. M. *La organización de la información para la gestión del conocimiento en las empresas* Última actualización: 2006. Disponible en: <http://www.gestiopolis1.com/recursos7/Docs/ger/organizacion-de-la-informacion-para-la-gestion-del-conocimiento.htm>.

DELGADO, Y. C. *GESTACAD. SISTEMA PARA LA GESTIÓN ACADÉMICA.* Editado por: publicado el: 2006 de 2006, última actualización: 2006. [Consultado el: febrero 2, 2012]. Disponible en: www.posgrados.frc.utn.edu.ar/congreso/trabajos/7.doc.

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

ECURED. *Educación* Última actualización: 2012. Disponible en: <http://www.ecured.cu/index.php/Educaci%C3%B3n>.

ESPAÑOLA, R.A... *Diccionario de la lengua española-Vigésima segunda edición* Última actualización: 2009. Disponible en: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=gestion.

PÉREZ, H. F... *JUnit: Manual Básico* Última actualización: marzo 14, 2011. Disponible en: <http://fuenteperez.es/blog/junit-manual-basico>.

FUNCION.COM, T... *Los mejores IDEs para Php* Última actualización: diciembre 9, 2007. (Patrocinar TuFuncion.com). Disponible en: <http://www.tufuncion.com/ide-php>.

GALLARDO R. J., García López Carmen M. *Diseño modular.* Facultad de Ciencias Matemáticas. Málaga.

INNOVA, G. S... *Rational Rose Data Modeler* Última actualización: 2009. (Soluciones y Propuestas Rational, Servicios, Capacitación, Consultoría). Disponible en: <http://www.rational.com.ar/herramientas/rosedatamodeler.html>

JACOBSON, G.B. Ivar. 2006. *Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000.

NETBEANS, C. DE. *Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto*. Última actualización: 2010. (¿Qué es NetBeans?). Disponible en: http://netbeans.org/index_es.html.

NORDESTE, U. N. DEL. *SIU-Guaraní*. Editado por: publicado el: 2006 de 2006, última actualización: 2006. [Consultado el: febrero 9, 2012]. Disponible en: http://criminalisticas.unne.edu.ar/documentos/Guaraní_folleto.pdf.

Oliveira D.S, R... *Teorías de la Administración*, International Thomson Editores, S.A. de C.V., 2002, Pág. 20.

ORQUÍN, A. F.; BARRIOS, Y... *Sistema para la recuperación de información docente*. Editado por: publicado el: 2006 de 2006, última actualización: 2006. [Consultado el: febrero 2, 2012]. Disponible en: http://informaticahabana.com/evento_virtual/files/EDU089.doc

OVIEDO, U. DE I. T. I. DE. *Entornos de Desarrollo Integrado*. Editado por: publicado el: 2009 de 2009, última actualización: 2009. [Consultado el: febrero 9, 2012]. Disponible en: <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos de Desarrollo Integrado.pdf>.

PÁEZ, U.I. *Gestión de la inteligencia, aprendizaje tecnológico y modernización del trabajo intelectual. Retos y oportunidades*. Caracas: Instituto de Estudios del Conocimiento de la Universidad Simón Bolívar, 1992.

PÉREZ, I. C.; GONZÁLEZ, R. P. et al... *Metodología de Desarrollo del Software*. Editado por: publicado el: octubre 15, 2008 de 2008, última actualización: octubre 15, 2008. [Consultado el: febrero 9, 2012]. Disponible en: <http://code.google.com/p/solusoft-g11/downloads/list/Metodologías de desarrollo.pdf>.

PROGRAMACIÓN, L. DE. *Lenguajes de programación, programación Java* Última actualización: 2009. (Programación Java). Disponible en: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.

RODRÍGUEZ, Y. P.; DOMÍNGUEZ, A. C... La gestión del conocimiento: un nuevo enfoque en la gestión empresarial: septiembre 24, 2005, vol. 13, n° [Consultado el: enero 31, 2012]. Disponible en: http://bvs.sld.cu/revistas/aci/vol13_6_05/aci040605.htm.

ROMERO, G. M. P... *MA-GMPR-UR2 Metodología ágil para proyectos de software libre...* Tutor: Universidad de las Ciencias Informáticas, 2008.

SALVATORI S. *Control de Versiones SVN | Stefano Salvatori.* ed. Última actualización: 2007. (). Disponible en: <http://stefano.salvatori.cl/blog/2007/10/03/control-de-versiones-svn/>.

SÁNCHEZ, M. A. M... *Metodologías De Desarrollo De Software.* Editado por: publicado el: julio 7, 2004 de 2004, última actualización: julio 7, 2004. [Consultado el: febrero 9, 2012]. Disponible en: <http://www.willydev.net/descargas/cualmetodologia.pdf>.

SANTAMARÍA, R... *jWebSocket: «A dream is coming true...».* Editado por: publicado el: 2011 de 2011, última actualización: 2011. [Consultado el: febrero 9, 2012]. Disponible en: <http://softwarelibre.hab.uci.cu/jwsblog/?p=1>.

SERRADILLA J.L. Sección de Metodología, Normalización y Calidad del Software. 2000, p.100

SIEMENS. *Siemens Industry* Última actualización: 2010. Disponible en: <https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo2&aktprim=99&lang=es>.

SUEIRO, G... *¿Qué es la confiabilidad?* Editado por: publicado el: 2009 de 2009, última actualización: 2009. [Consultado el: junio 8, 2012]. Disponible en: www.lezgon.com/pdf/IB00000020/41.pdf.

THOMPSON, I... *Definición de Eficacia* Última actualización: enero 2008. Disponible en: <http://www.promonegocios.net/administracion/definicion-eficacia.html>.

TORRES, P. L.; LÓPEZ, E. A. S... *Metodologías Ágiles en el Desarrollo de Software*. Editado por. Publicado el: noviembre 12, 2003 de 2003, última actualización: noviembre 12, 2003. [Consultado el: febrero 9, 2012]. Disponible en: <http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf>.

UNIVERSITARIA, S.A, O. DE C... *UNIVERSITAS XXI - ACADÉMICO PRESENTACIÓN DEL SISTEMA*. Editado por: publicado el: noviembre 8, 2011 de 2011, última actualización: noviembre 8, 2011. [Consultado el: febrero 9, 2012]. Disponible en: <http://ebookbrowse.com/presentacio-uxxi-ac-ocu-pdf-d205516378>.

URDANETA, I. P... *Bibliotecas Universitarias: La Crisis y La Oportunidad*. Editado por: publicado el: 1992 de 1992, última actualización: 1992. [Consultado el: febrero 2, 2012]. Vol. 3, 17 p. Disponible en: <http://ess.iesalc.unesco.org.ve/index.php/ess/article/view/195/157>.

BIBLIOGRAFÍA CONSULTADA

ANTELO, L. A.; ORDUÑEZ, J. M. G. *Módulo Emergencias del Sistema de Información Hospitalaria ALAS HIS*. Tutor. Tesis de Diploma, Universidad de las Ciencias Informáticas, 2009.

ARZUAGA, I. O. J.; CABO, M. A. S... *Sistema para la Gestión de la Información de Profesores y Estudiantes de la Facultad 6: Desarrollo del Módulo "Producción*. Tutor: Tesis de Diploma, Universidad de las Ciencias Informáticas, 2009.

BATISTA, A. L.; MORA, D. G... *Sistema de Gestión de Información de la Facultad 8. Diseño de la Base de Datos*. Tutor: Tesis de Diploma, Universidad de las Ciencias Informáticas, 2007.

CASTILLO, E. M.; JIMÉNEZ, Y. M. *Sistema de Gestión de Información del Polo de Software Educativo y Multimedia*. Tutor. Tesis de Diploma, Universidad de las Ciencias Informáticas, 2009.

CORBAL, E. C... «UNIVERSITAS XXI - ACADÉMICO», *El nuevo sistema informático para la matriculación universitaria*. Editado por: publicado el: agosto 20 de 2008, última actualización: agosto 20, 2008. [Consultado el: febrero 2, 2012]. Disponible en: <http://blog.aprendelo.com/universitas-xxi-academico-nuevo-sistema-informatico-matriculacion-universitaria/>.

DARA, G. *Información detallada módulos SIGA* Última actualización: 2009. Disponible en: <http://www.dara.es/siga/sigainf.htm>.

GARCÍA, D. M. G.; HERNÁNDEZ, A. I. F.. *Implementación del Sistema Integral de Documentación e Información Judicial para el Centro Nacional de Documentación e Información Judicial*. Tutor: Tesis de Diploma, Universidad de las Ciencias Informáticas, 2011.

JAVIER, E... *Módulos (Programación Estructurada)* Última actualización: 2012. Disponible en: <http://www.buenastareas.com/ensayos/Módulos-Programacion-Estructurada/1894885.html>.

IÑIGO, D. C. *Sistema de Gestión de la Información de la Facultad 8. Módulo para la Gestión de la Información Docente*. Tutor. Tesis de Diploma, Universidad de las Ciencias Informáticas, 2007.

PÉREZ, A. G.; HERNÁNDEZ, A. O. *Sistema de Gestión Académica en el Instituto Politécnico de Informática "Mártires de Girón*. Tutor: Tesis de Diploma, Universidad de las Ciencias Informáticas, 2007.

RICO, M. A. S... *Sistema de administración y control de renta de películas y libros vía web utilizando Spring*. Tutor: Universidad de las Américas Puebla, 2006.

RISCO, M. C.; MORALES, M. M. *Sistema para el control de la información de estudiantes y profesores. Desarrollo del Módulo profesores*. Tutor. Tesis de Diploma, Universidad de las Ciencias Informáticas, 2009.

UNIVERSITARIA, O. DE C. *Oficina de Cooperación Universitaria*
Última actualización: 2008. (Software de Gestión Académica Universitas XXI Académico). Disponible en: http://www.ocu.es/portal/page/portal/inicio/software_gestion_universitaria/sistema_gestion_academica.