



Facultad Regional “Mártires de Artemisa”

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

Título:

**Servidor para el módulo Presupuesto del Sistema Informativo
de la Administración Provincial de Artemisa**

Autor: Virgilio Suárez Bello

Tutor: Ing. Frank Rosales Muñoz

Cotutora: MsC. Rita M. Concepción Bravo

Artemisa, Junio 2012

Declaración de autoría

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

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

Virgilio Suárez Bello

MsC. Rita M. Concepción Bravo

Ing. Frank Rosales Muñoz

*"No hay que confundir nunca el conocimiento con la sabiduría.
El primero nos sirve para ganarnos la vida; la sabiduría nos
ayuda a vivir"*

Sorcha Carey

Agradecimientos

Agradecimientos a:

Mis padres Caridad y Virgilio a mis hermanas Izaida y Dianelis por todo el amor, apoyo y confianza que siempre me han brindado, que me ayudaron en cada momento que los necesité desde que comencé en la universidad y han estado conmigo en todo momento, en cada etapa de mi vida.

Mi novia Yenisley y su familia que me brindaron su amor y apoyo en los momentos que lo necesite.

Todos los que me han acompañado a lo largo de la carrera, de todos aprendí algo bueno.

Mis tutores por su paciencia y ayuda durante este año.

Todas las personas que de una forma u otra me ayudaron durante todos estos años en la universidad.

Muchas Gracias.

Dedico el presente trabajo a:

Mis padres y mis hermanas por brindarme a lo largo de estos cinco años todo su amor, esfuerzo y dedicación. Por ofrecerme su apoyo incondicional y por hacer posible la realización de este gran sueño.

Mi novia y su familia por su incondicional apoyo durante todos estos años.

Resumen

Con el avance de la informática y las telecomunicaciones en el mundo, han surgido cambios trascendentales en la forma de procesar la información, esto constituye un elemento relevante en el funcionamiento de una institución. El procesamiento de la información para las empresas, figura un problema fundamental, lo cual se ve representado como gastos, tanto en materiales como en tiempo y esfuerzo por parte del personal que se encarga de realizar estas operaciones. Encontrar una solución que realice estas funcionalidades de forma que se minimice el tiempo, esfuerzo y materiales empleados en procesar la información es imprescindible.

Como parte de la necesidad que tiene la Administración Provincial de Artemisa de automatizar sus procesos, se plantea el Sistema Informativo de la Administración Provincial (SI-NAP). Perteneciente a este proyecto se encuentra el servidor para el módulo Presupuesto que permite elevar los índices de rendimiento, seguridad y eficiencia en el procesamiento de la información del Departamento Presupuesto de la Dirección de Finanzas y Precios de la Administración Provincial de Artemisa.

El trabajo contiene una interesante investigación sobre las técnicas y herramientas de diseño y programación, así como de la Metodología de Desarrollo de Software utilizada. Además, se da una pequeña explicación del presupuesto. Se brinda una panorámica de los principales problemas existentes en el Departamento Presupuesto por lo que es necesaria esta propuesta de solución.

El documento incluye las etapas de Diseño, Implementación y Pruebas del servidor, sobre la base del Análisis que se hizo por parte de los analistas. A su vez, el servidor que se propone, en un futuro permitirá al Departamento Presupuesto de la Administración Provincial de Artemisa ejercer un mayor control del presupuesto que se maneja en la provincia.

Palabras claves

Departamento Presupuesto, Diseño, Implementación, Presupuesto, Servidor.

Introducción	1
Capítulo 1. Fundamentación teórica-metodológica	9
1.1 Principales definiciones	9
1.1.1 Presupuesto.....	9
1.1.2 Sistema presupuestario cubano.....	10
1.1.3 La gestión de la información	10
1.1.4 Módulos en programación.....	11
1.1.5 Ventajas de la modularización de programas	12
1.1.6 Servidores.....	12
1.2 Panorámica de los sistemas de gestión de presupuesto.....	13
1.2.1 Diseño de rendimiento, confiabilidad y eficiencia en el desarrollo de software ..	13
1.2.2 Soluciones informáticas internacionales para la gestión del presupuesto.	14
1.2.3 Soluciones informáticas nacionales para la gestión del presupuesto	16
1.3 Lenguaje unificado de modelado (UML).....	18
1.4 Herramientas CASE.....	19
1.4.1 Visual Paradigm.....	19
1.5 Metodologías de desarrollo de software.....	20
1.5.1 Xtreme Programming (XP)	20
1.5.2 SCRUM.....	21
1.5.3 SXP	21
1.6 Framework para el desarrollo del software.....	22
1.6.1 Spring Inversión de Control (IoC)	23
1.6.2 Hibernate	24
1.6.3 JWebSocket.....	24
1.7 Indicaciones generales para el desarrollo	26
1.7.1 Lenguaje Java.....	26
1.7.2 XML	27
1.7.3 NetBeans IDE	28
Conclusiones del capítulo.....	29
Capítulo 2: Características, análisis y diseño del sistema	30
2.1 Concepción del Servidor.....	30
2.1.1 Descripción de la propuesta de solución	30
2.1.2 Planificación del proyecto por roles.....	32

2.2 Modelo de dominio	33
2.3 Captura de requisitos	34
2.3.1 Lista de reserva del producto (LRP)	34
2.3.2 Historia de usuarios y prototipos de interfaz de usuario	38
2.4 Lista de riesgos	40
2.5 Diseño con metáforas	41
2.6 Diagrama de componentes	42
2.7 Tareas de Ingeniería	44
2.8 Plan de liberación	47
Conclusiones del capítulo	47
Capítulo 3: Validación de la propuesta de solución	48
3.1 Casos de pruebas	48
3.1.1 Prueba unitaria	48
3.1.2 Resultado de las pruebas	53
3.2 Resultados obtenidos	59
3.3 Funcionalidades obtenidas	59
3.4 Aporte social y económico	59
Conclusiones del capítulo	60
Conclusiones	61
Recomendaciones	62
Referencias bibliográficas	63
Bibliografía	65
Anexos	68
Glosario de términos	70

Introducción

La planificación presupuestaria es una práctica milenaria que el hombre ha utilizado a través de los años para poder distribuir mejor los recursos, tanto para el presente como para temporadas futuras. Los egipcios fueron uno de los primeros en desarrollar estas prácticas para poder pronosticar los resultados de sus cosechas para prevenir los años de escasez. Por su parte los romanos estimaban las posibilidades de pagos de sus colonias y así exigirles el tributo correspondiente. Pero no fue hasta finales del siglo XVIII cuando empezó a utilizarse el presupuesto para ayudar a la administración pública.

Los presupuestos se realizan para establecer objetivos, estrategias y elaborar los planes a seguir. En especial se encuentra ligado con la planificación financiera ya que forma parte del ciclo administrativo de planear, actuar y controlar como parte de un sistema de administración. Estos están orientados hacia el futuro y no hacia el pasado, aun cuando pueda ser comparado con períodos anteriores.

Cuando la planificación del presupuesto se realiza de forma manual se pueden cometer errores durante su confección, por esto con el inicio de la era de la informática y las telecomunicaciones se comenzó a pensar en cómo poder automatizar estos procesos. Durante esta nueva revolución la informática adquirió un nivel protagónico.

Entre los usos más frecuentes de la informática, se encuentra, abarcar la sistematización del tratamiento de la información y sus diversas formas de automatizarlos por medio de ordenadores. En los inicios de la informática solo se podían facilitar los trabajos repetitivos y monótonos de los sectores administrativos, con el paso de los años se abordaron otros sectores de la sociedad como el jurídico y el económico. La informatización de todos estos procesos trajo consigo una disminución de los costos y un incremento de la productividad.

La automatización de la información implica el almacenamiento, la organización y la transmisión de la misma, para ello, en la informática intervienen varias tecnologías, herramientas y conceptos que ayudan a desarrollar estos procesos. En términos generales, se puede decir que está compuesta por dos pilares básicos: la computación y la comunicación. En lo que hoy se conoce como informática existe la relación entre muchas de las técnicas y las máquinas que el hombre ha desarrollado a lo largo de la historia, para apoyar y potenciar

sus capacidades de memoria, pensamiento y de comunicación.

En la actualidad es difícil poder concebir un área que no se use de alguna forma el apoyo de la informática. Con ella se pueden cubrir grandes cantidades de funciones que van desde lo más simple hasta los cálculos científicos más complejos. Entre sus principales funciones se encuentran la creación de nuevas especificaciones de trabajo, el diseño, desarrollo e implementación de sistemas informáticos aplicados a varios sectores de la sociedad, la sistematización de los procesos, la optimización de los métodos y los sistemas informáticos existentes.

Por lo general los sistemas informáticos brindan un tratamiento a la información para su manejo que se basan en capturar la información brindada al sistema ya sea por un usuario o por otro sistema. Esta información pasa por un proceso en el cual el sistema decide que tratamiento darle a dicha información y más tarde se transmite un resultado.

Desde el comienzo de la era de la informática y las telecomunicaciones se comenzó a digitalizar todos los procesos que se realizaban de forma manual. Con el tiempo aumentaron las necesidades de gestionar la información, obligando a pequeñas y medianas empresas a apoyarse en herramientas para la gestión del presupuesto. Estos software de presupuesto proporcionan información mensual y anual sobre las variables financieras.

Todo este proceso de informatización se realiza a través de los software de aplicación, los que han sido escritos con el fin de realizar casi cualquier tarea imaginable. Este puede ser utilizado en cualquier instalación informática, independiente de su empleo. Existen miles de estos programas para ser aplicados en diferentes tareas, desde procesamiento de palabras hasta cómo seleccionar una universidad. Existen muchos programas, los cuales se dividen en cuatro categorías: las aplicaciones de negocios, de utilería, las personales y de entretenimiento.

En Cuba se han realizado investigaciones sobre el desarrollo de software para la gestión de la información del presupuesto. Pero estas investigaciones se desarrollan con herramientas privativas o con requisitos muy a la medida para una empresa determinada. Esto trae como consecuencia que no se pueda desplegar el sistema en otras instituciones ya que no cumplen con todas las expectativas y requisitos necesarios para realizar sus labores.

Entre los grandes impedimentos de poder aplicar un software determinado en cualquier institución es que se haya realizado con herramientas privativas, ya que estas se encuentran amparadas por licencias que reservan algunos o todos los derechos de uso, copia, modificación y/o distribución. Los software bajo esta licencia no pueden ser divulgados por ningún medio, ya que los usuarios que lo adquirieron no son sus dueños y solo lo pueden utilizar bajo las reglas que emite la licencia.

Por su parte las aplicaciones realizadas con software libre brindan la oportunidad a los usuarios de poder ejecutar, copiar, distribuir, estudiar, cambiar y/o mejorar el software. De modo más preciso, se refiere a las cuatro libertades que brinda el software libre a los usuarios: “la libertad de usar el programa con cualquier propósito, de estudiar como funciona adaptándolo a sus necesidades, tener acceso al código fuente, poder distribuir copias poder mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.” (Stallman, 1984)

En el año 2011 el Parlamento cubano aprobó separar la dirección de los Consejos de Administración y de las Asambleas provinciales y municipales del Poder Popular, en los nuevos territorios de Artemisa y Mayabeque. En el municipio cabecera de Artemisa se encuentra ubicado el Órgano del Gobierno Provincial, el cual está dividido en 32 direcciones, entre ellas, la Dirección de Finanzas y Precios. La misma está dividida en cuatro departamentos y una sección independiente, de ellos, el Departamento Presupuesto se encarga de gestionar en el territorio las políticas establecidas en materia de presupuesto en la provincia Artemisa.

Los especialistas del Departamento Presupuesto realizan el proceso presupuestario en un período anual que se desarrolla en cuatro etapas, primero planificación a través de anteproyecto de presupuesto que es cuando toda la información se recoge de diferentes fuentes y se introduce en excel. Luego se debe realizar una fundamentación teórica de por qué el monto de cada programa para justificarlo. La segunda etapa es la ejecución, la cual debe diseñarse sobre la base de la Ejecución de la Actividad Presupuestada (EAP) que emite el Ministerio de Finanzas y Precios (MFP).

Otro proceso que se realiza es el control a través de la verificación del gasto público, que es donde se registra en qué se ha empleado los recursos, si está acorde a las resoluciones

de uso y destino de los recursos y si la ejecución es correcta. Y por último está la liquidación de presupuesto que se realiza al final de la ejecución del período fiscal, es decir, al cierre del año.

Este proceso de gestión de información sobre los ingresos, los gastos, las multas, los pagos de subsidios de cada uno de los municipios que se recogen en el Departamento Presupuesto son costosos en cuanto a tiempo y esfuerzo. Debido a esto el especialista de presupuesto debe gestionar toda la información y llenar muchos modelos de forma manual pudiendo ocurrir errores durante este proceso. En ocasiones se deben llenar modelos de presupuesto oyendo el dictado de alguien que se encuentra al otro lado de la línea telefónica con los peligros que esto representa para la seguridad, confiabilidad e integridad de dicha información.

Los datos de ingresos de tarifas socioculturales son enviados por correo electrónico existiendo riesgo de seguridad de la información clasificada del presupuesto de la provincia y después el especialista de presupuesto debe llenar los modelos con esta información. En muchas ocasiones los datos de pagos de subsidios para llenar los modelos de presupuesto son enviados por mensajeros los cuales pueden perder o darle otros usos a esta información.

En el Departamento Presupuesto no existe una adecuada forma de procesar la información de los balances que se realizan. Se dificulta el análisis de los reportes ya que todos los datos se presentan en formato excel, lo que trae como consecuencia que la información pueda ser manipulada de forma incorrecta. Cualquier persona pueda tener acceso a la misma, por lo que la información no es íntegra y no cuenta con la consistencia adecuada, impidiendo al especialista de presupuesto poder hacer un aprovechamiento del tiempo y los datos con los que cuenta.

Cuando se necesita una información sobre desagregación actual de forma urgente se tiene que buscar manualmente y dentro de grandes volúmenes de información o contactar a través del teléfono a las personas, que en algún momento enviaron los datos para de esta forma poder corregir algún error que exista o para tomar decisiones que dependan de esta información.

Toda esta forma inadecuada de gestionar la información del Departamento Presupuesto ocasiona pérdida y duplicado de la información, problemas de organización, poca integridad de los reportes trayendo bajos índices de rendimiento, eficiencia y confiabilidad. Todo esto aumenta los retrasos en la entrega de información al presidente del gobierno, causando grandes problemas para la toma de decisiones en la provincia.

Por la situación problemática descrita anteriormente se declara el siguiente **problema científico**: ¿Cómo contribuir a elevar los índices de confiabilidad, eficiencia y rendimiento en el procesamiento de la información en el Departamento Presupuesto de la Dirección de Finanzas y Precios de la Administración Provincial de Artemisa?

Siendo el **objeto de estudio** el proceso de gestión de la información en los sistemas gubernamentales y el **campo de acción** está delimitado al procesamiento de la información en los sistemas de gestión de información en el área presupuesto.

El **objetivo general** de la investigación es desarrollar un servidor para el módulo Presupuesto del Sistema Informativo de la Administración Provincial (SINAP) de Artemisa, que contribuya a elevar los índices de confiabilidad, eficiencia y rendimiento en el procesamiento de la información.

El trabajo de diploma queda sustentado en la siguiente **idea a defender**: el desarrollo de un servidor para el módulo Presupuesto del SINAP de Artemisa, contribuirá a elevar los índices de confiabilidad, eficiencia y rendimiento de la gestión de la información en el Departamento Presupuesto de la Dirección de Finanzas y Precios de la Administración Provincial de Artemisa.

Para poder dar respuesta a los objetivos planteados se definen las siguientes **tareas de la investigación**:

- Establecimiento de los fundamentos teórico-metodológicos para el procesamiento de la información.
- Caracterización del procesamiento de la información en el Departamento Presupuesto de la Administración Provincial de Artemisa en lo relativo a la confiabilidad, eficiencia y rendimiento de la información.

- Establecimiento de los cimientos que deben sostener el servidor para el módulo Presupuesto del Sistema Informativo de la Administración Provincial de Artemisa.
- Desarrollo del servidor para el módulo Presupuesto del Sistema Informativo de la Administración Provincial de Artemisa.
- Validación del servidor a través de las pruebas unitarias.

El **aporte práctico** de la presente investigación consiste en el desarrollo de un servidor para el módulo Presupuesto del Sistema Informativo de la Administración Provincial de Artemisa, que permita elevar los índices de confiabilidad, eficiencia y rendimiento en el procesamiento de la información.

Población tomada para la investigación: La compone 1 especialista del Departamento Presupuesto.

Muestra tomada para la investigación: Se toma como muestra 1 especialista, lo cual representa el 100% de la población.

Declaración de Variables

Variable Independiente: Servidor.

Variables Dependientes: Confiabilidad, eficiencia y rendimiento.

La presente investigación se lleva a cabo por la necesidad que tiene el Departamento Presupuesto de la Administración Provincial de Artemisa de facilitar el análisis y el procesamiento de la información con el objetivo de elevar los índices de rendimiento, confiabilidad y eficiencia. A nivel mundial existen sistemas de gestión de información que se encargan de hacer más factible la gestión de los gobiernos, estos, a pesar de tener similitudes con la solución a implementar, contienen marcadas diferencias, pues su estructuración es completamente distinta.

Para la realización de este proyecto se emplearon los siguientes **métodos científicos** de la investigación:

Métodos Teóricos

Analítico-Sintético: El análisis nos permite estudiar la influencia de cada factor en particular sobre el procesamiento de la información en el Departamento Presupuesto, la síntesis posibilita descubrir las relaciones que guardan entre sí, poniendo de manifiesto la contribución de cada uno de los factores sobre los demás.

Histórico-Lógico: El objetivo es estudiar cómo se ha comportado los sistemas de gestión de la información en el campo del presupuesto en distintos períodos, las condiciones que influyeron en los cambios y conocer la lógica de su desarrollo, qué elementos de esencia incidieron en los cambios operados en cada etapa.

Modelación: La modelación es el proceso mediante el cual se crea una representación o modelo para investigar la realidad. El tipo de modelo utilizado es el modelo analógico que consiste en un esquema, diagrama o representación donde se refleja la estructura de relaciones y determinadas propiedades fundamentales de la información que se procesa en el Departamento Presupuesto.

Enfoque de Sistema: El presente método proporciona la orientación general para el estudio de los diferentes componentes que integran el Sistema Informativo de la Administración Provincial de Artemisa, como una realidad integral formada por componentes que cumplen determinadas funciones y mantienen formas estables de interacción.

Métodos empíricos

Entrevista: Se realizaron entrevistas a los especialistas del Departamento Presupuesto de la Dirección de Finanzas y Precios de la Administración Provincial de Artemisa con el objetivo de comprender su funcionamiento. **(Ver anexo 1)**

El presente trabajo de diploma se divide en tres capítulos:

Capítulo 1: Fundamentación teórica-metodológica.

El objetivo de este capítulo es tratar conceptos generales sobre el presupuesto y la gestión de la información. Se precisa el estado del arte con el estudio de varias soluciones informáticas para la gestión del presupuesto profundizando en el diseño de la confiabilidad, eficiencia y rendimiento. Se define la metodología, herramientas y tecnologías a utilizar para el desarrollo de la aplicación.

Capítulo 2: Características, análisis y diseño del sistema.

Este capítulo tiene como principal objetivo realizar una propuesta de solución al problema planteado y se especifican los principales artefactos generados durante las primeras fases; se define el modelo de dominio, los requisitos funcionales y los no funcionales, así como todo lo necesario para el diseño de la solución.

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

El objetivo de este capítulo es brindar valides y robustez a la propuesta de solución. Para ello se elaboran y aplican los casos de pruebas unitarias. Se realiza una descripción de la solución y los resultados obtenidos.

Finalmente, se presentan las **Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía, Anexos y Glosario de términos.**

Capítulo 1. Fundamentación teórica-metodológica

En el presente capítulo se realiza un estudio donde se explican las principales características del proceso en cuestión y las soluciones informáticas de gestión del presupuesto existente en el ámbito nacional e internacional. Por otra parte, se abordan las principales características del lenguaje seleccionado para su implementación y las ventajas del uso del framework JWebSocket en el desarrollo de aplicaciones web en tiempo real. Se presentan conceptos básicos de cada una de las tecnologías, herramientas y metodologías relacionadas con el análisis, diseño e implementación para el desarrollo de la propuesta de solución.

1.1 Principales definiciones

1.1.1 Presupuesto

Se pueden dar varias definiciones de lo que es un presupuesto, variando las mismas, en la medida que sus autores enfoquen los conceptos desde diferentes puntos de vista.

“El presupuesto puede ser definido como la presentación ordenada de los resultados previstos en un plan escrito, expresado en términos cuantitativos (monetarios y/o no monetarios, o ambas cosas que determina por anticipado el origen y asignación de los recursos de la organización; es un instrumento fundamental que sirve de apoyo a la realización de los objetivos de trabajo de cualquier organización o institución en un período de tiempo determinado. En él se expresan cuantitativamente los recursos que se asignan para la ejecución y cumplimiento de objetivos, de metas”, así lo refiere: (Ministerio de Educación Superior, 2001).

“Un presupuesto puede definirse como la presentación ordenada de los resultados previstos de un plan, un proyecto o una estrategia. A propósito, se plantea que esta definición hace una distinción entre la contabilidad tradicional y los presupuestos, en el sentido de que estos últimos están orientados hacia el futuro y no hacia el pasado, aun cuando en su función de control, el presupuesto para un período anterior pueda compararse con los resultados reales pasados”(Córdova, 2004).

Capítulo 1: Fundamentación teórica - metodológica

El Diccionario de la Real Academia define la palabra “presupuesto” como participio pasivo irregular del verbo presuponer, dándole, entre otras, la siguiente acepción: “Cómputo anticipado del coste y de los gastos e ingresos de una corporación”.

Después de haber consultado a varios autores se puede decir que es un plan de acción que muestra los gastos planeados y los ingresos esperados dirigido a cumplir una meta prevista, expresada en valores y términos financieros que, debe cumplirse en determinado tiempo generalmente un año y bajo ciertas condiciones previstas.

1.1.2 Sistema presupuestario cubano

“En Cuba es, por definición de su Constitución, una República unitaria, su sistema de Gobierno está estructurado a partir de 1976 y en correspondencia con la división política - administrativa, por un Gobierno central, los Gobiernos provinciales y los municipales. En consecuencia, cada nivel de gobierno, dispone de un presupuesto, que constituye la base financiera para cumplir con su papel de máxima autoridad estatal en los límites del territorio correspondiente. Ello da lugar a un Sistema de Presupuesto que está integrado por 200 presupuestos.”(Ministerio de Finanzas y Precios, 2007)

El presupuesto estatal reúne el Presupuesto Central del Estado y los presupuestos provinciales, incluyendo en estos últimos los correspondientes a los diferentes municipios que los componen. Todos los presupuestos municipales se muestran individualizados en la composición del respectivo presupuesto provincial. En el Presupuesto del Estado se incluye también el Presupuesto de la Seguridad Social.

1.1.3 La gestión de la información

“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. Para poder utilizar la información para tomar decisiones se debe gestionar la información (recabar, registrar y analizar). Por lo tanto, la gestión de la información implica: determinar la información que se precisa, recoger y analizar la información, registrarla y recuperarla cuando sea necesaria, para poder utilizarla y divulgarla” (Bartle, 2011).

Capítulo 1: Fundamentación teórica-metodológica

Una buena gestión de información ha de garantizar que todos los actores trabajen con información y estadísticas iguales o complementarias, pertinentes, precisas y actualizadas. Una adecuada recopilación y gestión de información durante las emergencias puede tener un impacto positivo para la recuperación temprana y las subsiguientes actividades de desarrollo y prevención de desastres.

1.1.4 Módulos en programación

“En programación, un módulo es un fragmento de un programa que se desarrolla de forma independiente del resto del programa. Esta independencia hace posible un mecanismo de compilación por separado que limita la complejidad del programa que se está desarrollando. Al compilarse el módulo por separado, la persona que lo desarrolla sólo debe preocuparse de él, prescindiendo en parte de cómo se utiliza este módulo dentro del programa. Quien escriba el resto del programa no debe preocuparse de los detalles del módulo sino sólo de cómo utilizarlo” (Cerrada, 2010).

Los módulos son, probablemente, la característica más innovadora del lenguaje. Hay cuatro tipos de módulos:

- módulos de programa
- módulos internos
- módulos de definición
- módulos de implementación

Los **módulos de programa** son unidades de programa completas que pueden importar recursos (constantes, variables, procedimientos...) de módulos de biblioteca.

Los **módulos de definición e implementación** son las dos partes que constituyen los módulos de biblioteca.

Los **módulos internos o locales** pueden ser definidos dentro del cuerpo de, bien módulos de programa, bien módulos de implementación, y se usan para ayudar a controlar el ámbito y la visibilidad de los distintos objetos.

1.1.5 Ventajas de la modularización de programas

Los módulos son una herramienta indispensable para crear programas reales cuyo tamaño sea grande. Un programador debe diseñar un programa grande como un conjunto de módulos, relacionados entre sí mediante interfaces definidas apropiadamente. Escribir y depurar el programa es más fácil porque el programador puede trabajar con un módulo cada vez, usando los servicios facilitados por otros módulos pero ignorando los detalles de cómo estos módulos trabajan (principio de abstracción). Este tipo de diseño modular es particularmente necesario cuando el programa se está desarrollando entre un conjunto de programadores, que es lo más habitual en cualquier programa de cierta envergadura.

Los programas modulares son más fáciles de modificar. Puesto que los detalles de implementación de un módulo se ocultan a los demás, se pueden cambiar detalles de un módulo sin afectar al resto. La modularización hace los programas más portables. El programador puede ocultar los detalles dependientes de la máquina en un único módulo, de forma que cuando se transporte a otro ordenador, sólo debe preocuparse de reescribir dicho módulo.

Cada módulo se compila por separado. El programa se divide en trozos, que el compilador puede procesar separadamente. De esta forma, un cambio en un módulo sólo requiere volver a compilar dicho módulo, no el programa completo. El desarrollo de bibliotecas con código reutilizable conlleva no sólo un ahorro de trabajo, sino además un aumento de la fiabilidad del programa, pues dichas bibliotecas están más probadas que si la parte de la biblioteca que se usa se codifica de nuevo.

1.1.6 Servidores

“Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.” (Wiley, 1993)

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo.

1.2 Panorámica de los sistemas de gestión de presupuesto

Para un mejor entendimiento de la investigación se ofrece una panorámica de los sistemas de gestión de presupuestos actuales más utilizados, tanto nacional como internacional. Cada una de ellos con sus características y funciones específicas, los cuales generalmente se encuentran enfocados al sistema gubernamental.

1.2.1 Diseño de rendimiento, confiabilidad y eficiencia en el desarrollo de software

El rendimiento es una variable en el proceso de desarrollo de software a la que no se le suele prestar demasiada atención pero que hace daño cuando el producto se encuentra en una fase avanzada de su construcción (porque requerirá esfuerzo corregirla) y todavía más en producción porque a ese coste hay que sumarle la pérdida de eficiencia y productividad de los usuarios en el uso de la herramienta y la disminución de la confianza de los mismos en el producto, lo que afectará negativamente a los gestores funcionales y técnicos del proyecto. Una buena arquitectura, codificación, diseño de los algoritmos, favorecen implícitamente al rendimiento de un software, sin embargo, la calidad final del producto es un aspecto decisivo en el rendimiento de cualquier producto.

La confiabilidad se refiere a la precisión que una aplicación proporciona sin errores, los servicios que se establecieron en las especificaciones originales. Un diseño que favorece la confiabilidad es el tiempo de funcionamiento de una aplicación antes la aparición de algún error, eliminar los puntos con errores desde el diseño de la aplicación, el control en la detección de errores y de la recuperación para evitar que se produzcan más errores.

La eficiencia de una aplicación se mide por la capacidad de gestionar grandes cantidades de información utilizando la menor cantidad de recursos posibles, la posibilidad de poder dar una respuesta en tiempo real para mantener actualizada la aplicación, sin necesidad de la intervención de un usuario. Uno de los puntos más importantes a medir es la capacidad de reducir la congestión en la red, esto se evidencia en el protocolo utilizado para

establecer la conexiones, el protocolo websocket mantiene conexiones persistentes, lo cual no quiere decir que la interacción entre cliente y servidor varíe un ápice desde el punto de vista lógico: cada par solicitud-respuesta es independiente. Si se han de establecer conexiones con una base de datos, todas las conexiones las realiza el mismo usuario, por lo que se puede emplear un connection pool (agrupamiento de conexiones), el cual permite compartir recursos de forma eficiente entre varios usuarios de la aplicación.

1.2.2 Soluciones informáticas internacionales para la gestión del presupuesto.

ASSETS NS

Sistema de Gestión Integral Estándar y Parametrizado, es comercializado por la firma panameña D'MARCO S.A. Está hecho con las más avanzadas tecnologías y respaldada por una poderosa plataforma de datos. Por todo esto cuenta con un alto prestigio en la industria del software. Está diseñado para múltiples compañías, con una estructura organizativa a varios niveles. Como sistema está diseñado para mantener el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos.

ASSETS NS es una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes de salidas.

Al ser ASSETS NS una aplicación cliente-servidor que realiza una manipulación mínima de datos en el sitio del cliente, todos los procesos se ejecutan en el servidor viajando al usuario las respuestas a cada uno de los procesos, lo que garantiza por una parte, que la carga en la red no sea significativa, permitiendo tener mayor número de usuarios conectados y, por otra, abarata los costos, dado que el hardware en el puesto del cliente no tiene grandes exigencias en cuanto a recursos.

El sistema cuenta con un módulo de Presupuesto que le permite definir el Presupuesto Contable de su entidad tanto para las cuentas nacionales como para las cuentas reales. Facilita el control de la ejecución de dichos presupuestos para un periodo y acumulados hasta la fecha, con análisis comparativos de porciento de ejecución en los distintos

Capítulo 1: Fundamentación teórica-metodológica

presupuestos.

Las opciones que brinda este módulo son las siguientes:

- Definir Presupuesto Contable para Cuentas Nominales y Reales.
- Definir Presupuesto de Producción.
- Análisis de la Ejecución del Presupuesto Contable del período y acumulado.

Sistema Integrado de Gestión y Control de las Finanzas Públicas (SIGECOF)

Este software fue diseñado por el Ministerio de Finanzas de la República Bolivariana de Venezuela desde 1995 y puesta legalmente en producción en el año 2001. En sus inicios fue concebida como una base de datos que sería nutrida por la información presupuestaria, financiera y contable derivada de la ejecución del Presupuesto Nacional.

Por causa de no contar con la tecnología necesaria se hizo imposible la conexión de los diferentes órganos de la República al SIGECOF. Para poder enfrentar las carencias legales, técnicas y tecnológicas en el año 2004, se puso en marcha un ambicioso proyecto de rediseño que arrancó con el perfeccionamiento y rediseño de la herramienta SIGECOF, para adaptarla a los cambios y en definitiva a los que motivaron su creación.

Entre las principales características del sistema se encuentran:

- Proporcionar un tratamiento y registro único de los datos en el sitio más cercano donde ocurre cada transacción.
- Cubrir la totalidad de las operaciones financieras en el ámbito presupuestario y no presupuestario.
- Posibilitar la obtención de la información sobre transacciones realizadas en el momento en que se producen.
- Apoyar el proceso de toma de decisiones y la simplificación de procedimientos.
- Dar información clara y pública sobre la gestión económica financiera del gobierno.

Capítulo 1: Fundamentación teórica-metodológica

- Proteger física y lógicamente dicha información contra el acceso no autorizado y el fraude.

Suite SIMBIOTIC SYSTEM (Módulo DAPHNE Pro+)

La empresa SIMBIOTIC SYSTEM desarrollo diversos módulos para la gestión gubernamental, dentro de sus módulos se encuentran la gestión presupuestaria. El módulo DAPHNE Pro+ es el encargado de gestionar el sistema presupuestario en una entidad. El módulo permite tener en solo cuestión de minutos un presupuesto listo con presentación profesional, todo esto parametrizable y configurable. Base de datos de precios unitarios de materiales.

Este módulo permite la generación de presupuestos desglosando las mediciones y/o el precio de las partidas, incluso detallar el precio de mano de obra y materiales por separado, indicar el precio de coste, etc. Generar automáticamente el contrato anexo al presupuesto, generar la carta asociada al presupuesto e incluso la portada del fax, generarlo en PDF.

1.2.3 Soluciones informáticas nacionales para la gestión del presupuesto

VERSAT Sarasola

Es un sistema de gestión contable-financiero creado por una filial en Villa Clara de TEICO, la empresa del MINAZ encargada de la Informática y las Comunicaciones. Esta empresa aglutinó y mejoró un grupo de programas aislados. En 1999 comenzó el desarrollo del VERSAT Sarasola. Dos años más tarde se comenzaron a utilizar en varias entidades del país.

Para el software se desarrollaron varios módulos para facilitar el control de los activos fijos e inventarios, la fianza y planificación entre otros. El resultado es un sistema integral que se actualiza constantemente en dependencia de las nuevas necesidades que se vayan presentando.

Capítulo 1: Fundamentación teórica-metodológica

Una de las principales características del VERSAT es su confiabilidad, el programa cuenta con diferentes subsistemas y un acceso controlado a cada una de las áreas, ya que en sus configuraciones se definen los niveles de acceso de cada uno de los usuarios.

Entre sus principales características se encuentran:

- Herramienta para la planificación económica, el control y el análisis de gestión.
- Diseñado para su empleo en cualquier tipo de entidad empresarial o gubernamental.
- Permite llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable en este universo.
- Se estructura en un grupo de subsistemas en los cuales se procesan y contabilizan los documentos primarios, donde se anotan los movimientos, los recursos materiales, laborales y financieros que se utilizan en una entidad.
- Rapidez y fiabilidad, a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema.

Después de haberse realizado un estudio de algunas soluciones informáticas para la gestión del presupuesto, se ha llegado a la conclusión de que los sistemas anteriores no cumplen con los requerimientos para su implantación en el Departamento Presupuesto de la dirección de Finanzas y Precios de la Administración Provincial de Artemisa. En el caso de ASSETS NS y SIGECOF no cumplen con las leyes del Proceso Presupuestario vigentes en Cuba, por su parte Suite SIMBIOTIC SYSTEM (Módulo DAPHNE Pro+) es una suite privativa y los precios oscilan entre los \$250 y \$350 dólares. Por su parte la aplicación VERSAT Sarasola es una solución nacional, pero no cumple con los requerimientos del Departamento Presupuesto de la Administración Provincial de Artemisa. Por lo anteriormente planteado se justifica la creación de una solución informática para esta entidad.

1.3 Lenguaje unificado de modelado (UML)

Los lenguajes unificados de modelado (UML) permiten estandarizar la forma de poder representar gráficamente un modelo. Es un lenguaje que se utiliza para poder comunicar las ideas entre los desarrolladores y sirve de apoyo en los procesos de análisis de un problema. Cuenta con una notación estándar y semántica esencial para el modelado de un sistema orientado a objetos.

Entre las principales características que brinda el UML se encuentran:

- Visualizar de forma gráfica para poder facilitar el entendimiento del mismo.
- Especificar cuales son las características de un sistema antes de su construcción.
- Construir a partir de los modelos específicos los sistemas diseñados.
- Documentar los elementos gráficos para que sirvan como documentación del sistema desarrollado.

La representación gráfica de un conjunto de elementos con sus relaciones se denomina diagrama. UML ofrece una gran variedad de diagramas para poder hacer una visualización del sistema desde varias perspectivas:

- Diagrama de caso de uso: representa los procesos del negocio.
- Diagrama de clases: representar la estructura estática de las clases del sistema.
- Diagrama de objeto: simbolizar la estructura estática de las clases del sistema.
- Diagramas de comportamiento: está compuesto por los diagramas de estado que se utiliza para modelar el comportamiento de los objetos del sistema y diagrama de actividad que se utiliza para modelar el comportamiento de los casos de uso, objetos u operaciones.
- Diagramas de interacción: está compuesto por el diagrama de secuencia que se utiliza para especificar el paso de los mensajes entre los objetos y diagrama de

colaboración que se utiliza para modelar la interacción entre los objetos.

- Diagrama de implementación: está compuesto por el diagrama de componente que se utiliza para modelar los componentes y el diagrama de despliegue que se utiliza para modelar el despliegue del sistema.

1.4 Herramientas CASE

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad y acelerar el proceso de desarrollo de software. Cuenta con diccionarios donde puedes almacenar los elementos definidos anteriormente. Permite la carga y descarga de datos de otros sistemas y genera esquemas de bases de datos. Brinda las funcionalidades de comprobación de errores, el análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta. Cuentan con interfaz de usuario, editores de texto, de diseño gráfico para poder definir diagramas y matrices. Las herramientas CASE sustituyen el papel y el lápiz por la PC para hacer de la actividad desarrolladora de software una actividad automatizada.

1.4.1 Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de un software. Ayuda a una más rápida construcción de aplicaciones de alta calidad a un menor costo. Permite poder dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y general la documentación. Brinda la posibilidad de un mapeo de objetos relacionales para los lenguajes de programación Java .NET y PHP. Cuenta con gran cantidad de documentación para aprender a utilizarla y trabajar con ella.

Alguna de sus principales características son:

- Se encuentra disponible en múltiples plataformas (Windows, Linux).
- Capacidad de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

- Genera código para Java y exportación como HTML.
- Soporta versión 2.1 de UML.
- Diagramas de flujo de datos.
- Soporta ORM – Generación de objetos Java desde la base de datos.
- Importación y exportación de ficheros XML.

Visual Paradigm se emplea para el modelado de diagramas debido a que es una herramienta multiplataforma, se puede realizar una distribución automática de diagramas. También posee abundantes tutoriales, demostraciones interactivas y proyectos UML. Es fácil de instalar y actualizar y compatible entre ediciones. Permite exportar los diagramas a imágenes y páginas web HTML.

1.5 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de software. La metodología va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

1.5.1 Xtreme Programming (XP)

XP es una metodología ágil centrada en promover el trabajo en equipo, potencia las relaciones interpersonales, siendo la clave para el éxito del software, se preocupa por el aprendizaje de los desarrolladores. Se basa fundamentalmente en la realimentación entre el cliente y el equipo de desarrollo, mantiene una comunicación fluida entre todas las personas y promueve la simplicidad en las soluciones implementadas. Esta concebida para proyectos con requisitos imprecisos y muy cambiantes, y donde exista un alto riesgo técnico.

Capítulo 1: Fundamentación teórica-metodológica

Entre las características esenciales de XP se encuentran:

- Las pruebas unitarias continuas: son un conjunto de pruebas que se realizan a los principales procesos con el objetivo de obtener posibles errores que pudieran ocurrir en un futuro.
- Recodificación: con la intención de mejorar la legibilidad y mantenibilidad reescribe algunas partes del código pero sin modificar su comportamiento.
- Programación en parejas: las tareas de desarrollo se realizan entre dos personas en un mismo puesto.

La principal suposición que se realiza con XP es disminuir el cambio a lo largo del proyecto para que el diseño evolutivo funcione. Esto se puede conseguir gracias a las tecnologías disponibles para ayudar en el desarrollo del software y las buenas prácticas de esta metodología.

1.5.2 SCRUM

SCRUM más que una metodología de trabajo es un marco de trabajo para la gestión y desarrollo de software, se basa en un proceso iterativo e incremental. Un grupo de integrantes del equipo deciden cómo hacer sus tareas y cuánto van a tardar. Generalmente se utiliza en entornos basados en desarrollado ágil de software. Ayuda a que el trabajo se realice en una misma dirección y con los objetivos claros y además se puede seguir de forma clara los avances obtenidos durante el día de trabajo. Sin embargo, no especifica lo que se debe hacer para realizar el código por lo que se debe complementar con otras metodologías a desarrollar.

1.5.3 SXP

SXP es una metodología de desarrollo de software compuesta por las metodologías XP y SCRUM, la cual es un método adaptativo de gestión de proyectos que se basa en procedimientos ágiles que permitan actualizar los procesos de software para un mejor aprovechamiento de las actividades productivas.

Capítulo 1: Fundamentación teórica - metodológica

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 pueden ver día a día cómo progresa el trabajo.

SXP se compone por cuatro fases principales:

- Planificación: se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo: se realiza la implementación del sistema hasta que esté listo para ser entregado al cliente.
- Entrega: se entrega el software y la documentación.
- Mantenimiento: se realiza el soporte para el cliente.

Se seleccionó esta metodología de desarrollado de software para el presente trabajo diploma por todos los beneficios mencionado anteriormente y porque tiene como premisa la no duplicación de los esfuerzos permitiendo un aprovechamiento del tiempo y solo documentar la información necesaria para futura reutilización, ya que se mantiene una estrecha comunicación entre el cliente y el equipo de trabajo.

1.6 Framework para el desarrollo del software

Framework es un concepto sumamente genérico, se refiere a un ambiente de trabajo, y ejecución, por ejemplo .Net es considerado un framework para desarrollar aplicaciones (sobre Windows). En general los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

Ventajas de utilizar un framework para el desarrollo de un software

- El programador no necesita plantearse una estructura global de la aplicación,

sino que el framework le proporciona un esqueleto que hay que rellenar.

- Facilita la colaboración. Permite que más de un desarrollador pueda trabajar en un mismo proyecto ya que al utilizar un framework se estandariza todo el trabajo.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

1.6.1 Spring Inversión de Control (IoC)

Spring Framework es un framework de código abierto que se utiliza para el desarrollo de aplicaciones en Java. Se encuentra estructurado de tal forma que el desarrollador no está obligado a usar un modelo de programación en particulares y por su diseño ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. Este framework contiene muchas mejoras y extensiones para desarrollar aplicaciones basadas en web.

A diferencia de los frameworks de un solo nivel, tales como Struts o Hibernate, Spring tiene como objetivo ayudar a las aplicaciones de toda la estructura de una manera coherente y productiva, reuniendo más frameworks de un solo nivel para crear una arquitectura coherente. Spring es un framework ligero basado en la inyección de dependencias (DI) de patrones. El uso principal de Spring es como un framework que permite el uso de Inversión de Control (IOC). Es también un potente entorno de aplicaciones Java Enterprise, que proporciona un gran número de funciones y servicios. Para aplicaciones empresariales, Spring es una herramienta excelente porque está bien documentado, es escalable y de código abierto con mucho apoyo de la comunidad.

Con Spring, se puede inyectar la totalidad de sus clases directamente en el contenedor mediante un archivo XML. Esto significa que el contenedor de Spring instancia todas las clases y se inyecta en otros como se define en XML.

Spring ofrece servicios y recursos para las conexiones de bases de datos, transacciones de bases de datos, la seguridad, de mensajería, almacenamiento en caché, y mucho más. Para gestionar la seguridad de la aplicación con todos los grupos, los roles, y los permisos, sólo se tiene que inyectar un poco de clases en el framework de seguridad de Spring.

Soporta todas las tecnologías de acceso a los principales datos, como JDBC, Hibernate, JPA, e iBatis.

1.6.2 Hibernate

Hibernate es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

1.6.3 JWebSocket

JWebSocket es un framework orientado al desarrollo de aplicaciones basadas en websocket, que permite altos niveles de velocidad y seguridad. WebSocket es una nueva tecnología que, en palabras del W3C, permite a las aplicaciones web mantener una comunicación bidireccional con procesos en el lado del servidor. La primera aparición de un borrador de trabajo sobre esta tecnología data de Abril de 2009 y, aunque más de año y medio después sigue en el mismo estado, ha sido una tecnología que ha levantado

Capítulo 1: Fundamentación teórica-metodológica

grandes expectativas, hasta el punto de ser calificada como "el TCP de la Web". Esta tecnología venía a sustituir en buena medida a XMLHttpRequest como forma de comunicación con el servidor, prometiendo simplificar el actual modelo.

Entre las características que hacen interesante a WebSocket se encuentra la promesa de ofrecer una API simple para comunicar con el servidor así como la posibilidad de que estas comunicaciones sean bidireccionales, es decir, que puedan iniciarse en el lado del servidor; todo ello con una mejora en la latencia que actualmente soportan esta clase de comunicaciones

JWebSocket es una solución Open-Source en Java y JavaScript para el protocolo websocket de HTML5. Ofrece una amplia gama de funcionalidades que van desde el intercambio de bajo nivel hasta la sincronización de interfaces gráficas del usuario y las llamadas a procedimientos remotos. Esta tecnología permite programar una solución independiente con un alto nivel de seguridad o integrarla a aplicaciones ya existentes, todos estos procesos con un mínimo de configuraciones y una fácil instalación, reutilizando una gran cantidad de tecnologías, librerías y aplicaciones existentes para la tecnología Java.

Ventajas de su uso

- Comunicación bidireccional.
- Tecnología Full-duplex.
- Conexiones TCP permanentes entre cliente y servidor.
- 400 veces menos sobrecarga en la red.
- Solo 1/3 de la latencia comparado con HTTP.
- Estandarizado por W3C e IETF, fácil integración con proxies y firewalls.

Esta tecnología es compatible con varios navegadores: Internet Explorer (mediante un plugin de flash), Chrome 4+, Opera 11+, Firefox 4+, Safari 5+.

Se seleccionó JWebSocket Framework para el desarrollo de la presente investigación por todas las características y ventajas enunciadas anteriormente. Además, permite utilizar la tecnología websocket que brinda la posibilidad de una comunicación TCP basada en sockets mejorando grandemente la rapidez en la web. Esta tecnología permite una menor sobrecarga en la red en el orden de 400 veces. Permite su integración con Spring IoC para realizar sus configuraciones y con Hibernate para el trabajo con la base de datos. El framework JWebSocket establece una comunicación en tiempo real entre el cliente y el servidor permitiendo una aplicación más rápida y menos sobrecarga en la red.

1.7 Indicaciones generales para el desarrollo

1.7.1 Lenguaje Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems. Este lenguaje toma mucha de las sintaxis de C y C++ pero cuenta con un modelo de objetos más simple y elimina las herramientas de bajo nivel. La memoria es gestionada por el propio lenguaje y no por el programador quitando muchos errores que suelen ocurrir por esta razón. Este lenguaje cuenta con la peculiaridad que es independiente de plataforma, lo que significa que los programas escritos en lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware y software siempre que se cuente con su maquina virtual (JVM).

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como bytecode (específicamente Java bytecode) instrucciones máquina simplificada específica de la plataforma Java. El bytecode es ejecutado entonces en la JVM, un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo de forma unificada.

Ventajas de su uso:

- Lenguaje simple: posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir una aplicación desde el principio.

Capítulo 1: Fundamentación teórica-metodológica

- Orientado a objetos: Java fue diseñado como un lenguaje orientado a objetos. Los objetos se agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos.
- Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- Interpretado y compilado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- Portable: La indiferencia a la arquitectura representa solo una parte de su portabilidad. Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

1.7.2 XML

Extensible Markup Language (XML) es un lenguaje de marcado que define un conjunto de reglas para los documentos de codificación en un formato que sea legible para las personas. Aunque el diseño de XML se centra en los documentos es ampliamente utilizado para la representación de las estructuras de datos, por ejemplo en los servicios web.

Principales objetivos de su uso;

- Idéntico a la hora de servir, recibir y procesar la información que el HTML, para aprovechar toda la tecnología implantada para este último.
- Extensible, para que lo puedan utilizar en todos los campos del conocimiento.
- Fácil de leer y editar.

Capítulo 1: Fundamentación teórica - metodológica

- Fácil de implantar, programar y aplicar a los distintos sistemas.

Permite proporcionar diferentes vistas sobre los datos (HTML, PDF, voz), dependiendo de quién sea el cliente. Facilita la integración desde fuentes de datos heterogéneas, por ejemplo, páginas web, distintas bases de datos, entre otros. Los documentos tienen una estructura que los hace legibles e inteligibles no solo para los ordenadores, si no también para los humanos. Las aplicaciones de XML son fácilmente extensibles mediante definiciones de nuevos tipos de documento.

1.7.3 NetBeans IDE

NetBeans es un entorno de desarrollo hecho principalmente para el lenguaje Java, aunque existen módulos para extender su uso a otros lenguajes de programación. Es un sistema libre, gratuito y sin restricciones de uso. Cuenta con herramientas para crear aplicaciones profesionales para escritorio, web y equipos móviles. Soporta grandes cantidades de librerías y cuenta con versiones para los principales sistemas operativos: MacOS, Linux y Windows.

Principales características:

- Soluciones Rápidas (Quick Fixes) y Verificación de Sintaxis.
- Refactorización.
- Menor consumo de memoria.
- Especificación de dependencias de librerías usando direcciones de librerías relativas (por defecto Java, Web y todos los proyectos tipo Java EE).
- Compartir proyectos más fácilmente con otros miembros de su equipo, cuando por ejemplo use un sistema de control de versión.
- Habilite compilaciones no finalizadas en sus proyectos.
- Control de versión ClearCase.
- Soporte Hibernate y Spring Framework.

Conclusiones del capítulo

Durante el estudio de la bibliografía y el resultado de la investigación realizada en este capítulo se han expuesto los principales puntos de interés relacionados con la gestión del presupuesto. Los análisis de algunas de las soluciones existentes que gestionan la información presupuestaria han demostrado la necesidad de crear un servidor para el módulo Presupuesto que permita elevar los índices de rendimiento, confiabilidad y eficiencia en el Departamento Presupuesto de la Administración Provincial de Artemisa. El estudio de diferentes herramientas, lenguajes, metodologías, framework ha permitido conocer cuales son las que mejor se ajustan para el desarrollo de una solución al problema planteado.

Capítulo 2: Características, análisis y diseño del sistema

En el presente capítulo se comienza a desarrollar una propuesta de solución guiada por la metodología SXP, estableciendo las características del sistema para poder llegar a un mayor entendimiento de este. Se especifican los principales artefactos, definiéndose los requisitos funcionales y los no funcionales. Se definen las historias de usuario y las tareas de ingeniería.

2.1 Concepción del Servidor

Desde el momento en que se crea un proyecto lo primero que realiza el equipo de desarrollo es una entrevista con sus clientes. Todo esto para poder tener una visión de lo que el cliente desea y así poder definir los roles que intervendrán en el desarrollo del producto. En esos momentos se especifican las herramientas, tecnologías y metodología que se utilizaran para el desarrollo del producto.

2.1.1 Descripción de la propuesta de solución

Como parte de la solución informática del Sistema Informativo de la Administración Provincial de Artemisa se desarrolla un módulo Presupuesto. Esta investigación se enfoca en el desarrollo de un servidor para este módulo cuyo objetivo es la automatización del procesamiento de la información en el Departamento Presupuesto de la Administración Provincial de Artemisa.

El servidor permite a cualquier aplicación cliente que cuente con las características necesarias para establecer la comunicación con él, una serie de funcionalidades para poder gestionar la información del Departamento Presupuesto de la Administración Provincial de Artemisa.

Como parte de estas funcionalidades se encuentran insertar, modificar, eliminar y buscar la información de cada uno de los modelos con los que trabaja el especialista del Departamento Presupuesto. Toda esta información que se procesa en el servidor pasa por un sistema de filtros de seguridad para comprobar que no existan errores en los datos enviados por la aplicación cliente.

Capítulo 2: Características, análisis y diseño del sistema

A partir de esta información insertada, el servidor brinda la posibilidad de poder realizar reportes de cada uno de los modelos y realizar las operaciones matemáticas de algunos de los campos de dicho reporte para así ser enviados a la aplicación cliente.

Arquitectura de software

La solución propuesta anteriormente se desarrolla sobre una arquitectura n-capas la cual se basa en separar los componentes tanto como sea posible y cada capa es, simplemente, la agrupación de todos los componentes que tienen una funcionalidad común.



Fig. 2.1: Arquitectura 4 Capas

Ventajas que ofrece el uso de esta arquitectura:

- Desarrollos paralelos (en cada capa)
- Aplicaciones más robustas debido al encapsulamiento
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación completa)
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad (La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento

Capítulo 2: Características, análisis y diseño del sistema

es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad).

2.1.2 Planificación del proyecto por roles

Rol	Responsabilidad	Nombre
Gerente	Es el encargado de tomar las decisiones finales, participa en la selección de objetivos y requerimientos. Evalúa si los objetivos son alcanzables con las restricciones del tiempo y recursos presentes.	Dania Fernández
Jefe de Proyecto	Es el encargado de asignar los recursos, gestionar las prioridades, coordinar las iteraciones con los clientes y usuarios, mantener el equipo de proyecto enfocado en los objetivos. Supervisa el establecimiento de la arquitectura del sistema, la gestión de riesgos y la planificación y control del proyecto.	Frank Rosalez Muñoz
Programador	Elaborar el código de las nuevas funcionalidades a implementar. Escribe e implementa las pruebas unitarias para probar el sistema.	Yuleisy Lugo Alfonso y Virgilio Suárez Bello
Diseñador de BD	Responsable del diseño e implementación de las bases de datos	Nagyara Fernandez Martinez
Analista	Capturar, especificar y validar requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaborar el Modelo de Análisis y Diseño. Colaborar en la elaboración de las pruebas funcionales y el modelo de datos.	Yuleisy Lugo Alfonso y Virgilio Suárez Bello

Tabla 2.1: Planificación del Proyecto por Roles

2.2 Modelo de dominio

Dentro de las actividades más importantes definidas en la metodología SXP se encuentra el Modelo de Historia de Usuario del Negocio en el cual se detallan las características del negocio. En caso que el negocio no este bien definido entonces se genera el Modelo de Dominio. En el mismo se representan las clases conceptuales u objetos del mundo real de forma que ayude a entender el problema en cuestión. Este modelo ayuda a comprender conceptos con los que los usuarios trabajan y deben ser parte de la solución final.

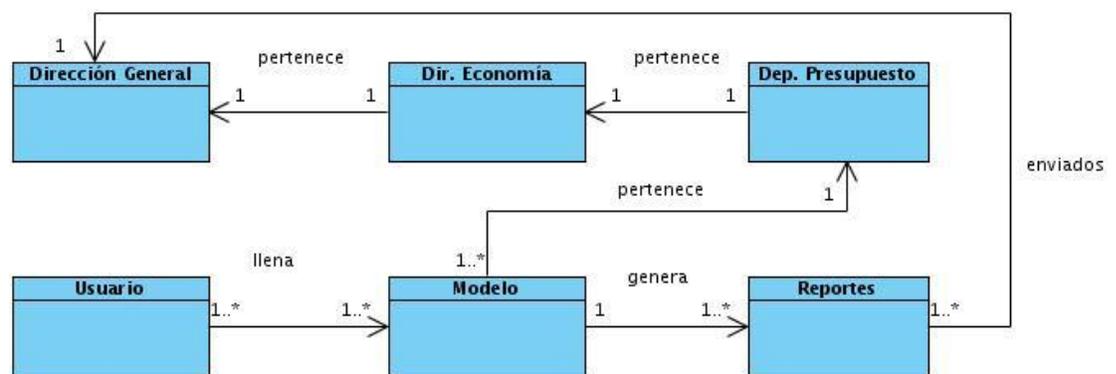


Fig. 2.2: Modelo de Dominio de la Propuesta de Solución

Conceptos del modelo de dominio

- **Dirección General:** Es la representación de la Dirección Central de la Administración Provincial de Artemisa.
- **Dir. Economía:** Es la representación de la Dirección de Economía de la Administración Provincial de Artemisa.
- **Dep. Presupuesto:** Es la representación del Departamento Presupuesto de la Administración Provincial de Artemisa.
- **Usuario:** Especialista del Departamento Presupuesto que se encarga de gestionar toda la información referente al departamento.
- **Modelo:** Plantilla que se llena con la información que maneja el Departamento Presupuesto para realizar su trabajo.

Capítulo 2: Características, análisis y diseño del sistema

- Reportes: Información en formato duro o digital con el resultado de los datos pedidos por alguna dirección.

2.3 Captura de requisitos

La captura de requisitos es un muy importante en el desarrollo de un software ya que permite gestionar las necesidades del proyecto en forma estructurada, cada actividad tendrá los pasos a seguir. Permite la capacidad de predecir cronogramas de proyecto, proporcionando un punto de partida para controlar actividades específicas. Evita rechazos finales del usuario debido a que obliga a los usuarios a considerar sus requerimientos cuidadosamente.

2.3.1 Lista de reserva del producto (LRP)

Una de las actividades más importantes definidas en la metodología SXP es la Lista de Reserva del Producto (LRP) esta es elaborada por el analista en conjunto con el cliente. Esta conformada por los requerimientos que el sistema debe cumplir para satisfacer las necesidades del cliente y las prestaciones con las que se debe contar para poder desarrollar y poner en funcionamiento la aplicación. Los requisitos funcionales se encuentran divididos por prioridades que van desde la muy alta hasta la baja con una estimación por cada requisito. La lista puede crecer mediante se obtiene más conocimiento acerca del producto y el cliente.

Prioridad	Ítem*	Descripción	Estimación	Estimado por
Muy Alta				
	1	Insertar la Información Mensual de Pagos del Presupuesto Central.	1 día	Analista
	2	Insertar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	1 día	Analista
	3	Insertar la Información de Ingresos por el Cobro del Sector Salud Pública.	1 día	Analista

Capítulo 2: Características, análisis y diseño del sistema

4	Insertar la Información de Gastos de Capital en Inversiones Materiales.	1 día	Analista
5	Insertar la Información de Gastos de Capital en Inversiones no Materiales.	1 día	Analista
6	Insertar la Información Mensual del Presupuesto Local.	1 día	Analista
7	Insertar la Información de los Balances de Ingresos y Gastos.	1 día	Analista
8	Insertar la Información de los Balances mensuales de Ingresos y Gastos.	1 día	Analista
9	Insertar la Información de la Recaudación de fondos del Programa Semanal Especial.	1 día	Analista
10	Eliminar la Información Mensual de Pagos del Presupuesto Central.	1 día	Analista
11	Eliminar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	1 día	Analista
12	Eliminar la Información de Ingresos por el Cobro del Sector Salud Pública.	1 día	Analista
13	Eliminar la Información de Gastos de Capital en Inversiones Materiales.	1 día	Analista
14	Eliminar la Información de Gastos de Capital en Inversiones no Materiales.	1 día	Analista
15	Eliminar la Información Mensual del Presupuesto Local.	1 día	Analista
16	Eliminar la Información de los Balances de Ingresos y Gastos.	1 día	Analista

Capítulo 2: Características, análisis y diseño del sistema

17	Eliminar la Información de los Balances mensuales de Ingresos y Gastos.	1 día	Analista
18	Eliminar la Información de la Recaudación de fondos del Programa Semanal Especial	1 día	Analista
19	Modificar la Información Mensual de Pagos del Presupuesto Central.	1 día	Analista
20	Modificar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	1 día	Analista
21	Modificar la Información de Ingresos por el Cobro del Sector Salud Pública.	1 día	Analista
22	Modificar la Información de Gastos de Capital en Inversiones Materiales.	1 día	Analista
23	Modificar la Información de Gastos de Capital en Inversiones no Materiales.	1 día	Analista
24	Modificar la Información Mensual del Presupuesto Local.	1 día	Analista
25	Modificar la Información de los Balances de Ingresos y Gastos.	1 día	Analista
26	Modificar la Información de los Balances mensuales de Ingresos y Gastos.	1 día	Analista
27	Modificar la Información de la Recaudación de fondos del Programa Semanal Especial	1 día	Analista
28	Buscar la Información Mensual de Pagos del Presupuesto Central.	1 día	Analista
29	Buscar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	1 día	Analista

Capítulo 2: Características, análisis y diseño del sistema

	30	Buscar la Información de Ingresos por el Cobro del Sector Salud Pública.	1 día	Analista
	31	Buscar la Información de Gastos de Capital en Inversiones Materiales.	1 día	Analista
	32	Buscar la Información de Gastos de Capital en Inversiones no Materiales.	1 día	Analista
	33	Buscar la Información Mensual del Presupuesto Local.	1 día	Analista
	34	Buscar la Información de los Balances de Ingresos y Gastos.	1 día	Analista
	35	Buscar la Información de los Balances mensuales de Ingresos y Gastos.	1 día	Analista
	36	Buscar la Información de la Recaudación de fondos del Programa Semanal Especial	1 día	Analista
Alta				
	37	Generar reporte de la Información Mensual de Pagos del Presupuesto Central.	2 días	Analista
	38	Generar reporte de la información de los Ingresos por el Cobro del Sector Cultura y Arte.	2 días	Analista
	39	Generar reporte de la Información de los Ingresos por el Cobro del Sector Salud Pública.	2 días	Analista
	40	Generar reporte de la Información de los Gastos de Capital en Inversiones Materiales.	2 días	Analista

Capítulo 2: Características, análisis y diseño del sistema

	41	Generar reporte de la Información de los Gastos de Capital en Inversiones no Materiales.	2 días	Analista
	42	Generar reporte de la Información Mensual del Presupuesto Local.	2 días	Analista
	43	Generar reporte de la Información de los Balances de Ingresos y Gastos.	2 días	Analista
	44	Generar reporte de la Información de los Balances Mensuales de Ingresos y Gastos.	2 días	Analista
	45	Generar reporte la Información de la Recaudación de fondos del Programa Semanal Especial	2 días	Analista
RNF (Requisitos No Funcionales)				
	46	Instalar versión 7 de la Máquina Virtual de Java.		
	47	Instalar NetBeans 7.0.		
	48	Tener una PC con 1 GB de RAM y 80 GB de disco duro como mínimo.		
	49	Navegador web compatible con la tecnología websocket		

Tabla 2.2: Lista de Reserva del Producto

2.3.2 Historia de usuarios y prototipos de interfaz de usuario

En la metodología SXP, las historias de usuarios son las que describen las tareas que el sistema debe hacer. Se realiza en dependencia de los requisitos funcionales descritos en la Lista Reserva del Producto (LRP). Este artefacto sirve como guía para las futuras pruebas de aceptación, comprobando de esta forma la correcta implementación de los

Capítulo 2: Características, análisis y diseño del sistema

requisitos funcionales pedidos por el cliente. Se describen con un lenguaje claro y natural para que puedan ser comprendidas por el resto del equipo que las vaya a utilizar.

Durante la planificación inicial del servidor para el módulo Presupuesto se generaron 18 Historias de Usuarios, de las cuales 9 se encuentran en la prioridad en negocio muy alta que responden a los gestionar información con un tiempo estimado de 2/3 de semana para cada una y un riesgo de desarrollo alto. Los restantes 9 se encuentran en la prioridad en negocio alta y responden a los generar reportes, cuentan con un tiempo estimado de 1/3 de semana cada una y un riesgo de desarrollo alto. A continuación se expone un pequeño ejemplo de dos de las Historias de Usuarios.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar la Información Mensual de Pagos del Presupuesto Central.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Virgilio Suárez Bello	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2/3
Riesgo en Desarrollo: Alto	Puntos Reales: 2/3
Descripción: El objetivo fundamental que seguirá esta historia de usuario es procesar los datos insertados, eliminados, modificados y buscados relacionados con la información mensual de los Pagos del Presupuesto Central.	
Observaciones: Ninguna	
Prototipo de interface: Ninguna	

Tabla 2.3: Historia de Usuario Gestionar la Información Mensual de Pagos del Presupuesto Central

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Generar reporte de la Información Mensual de Pagos del Presupuesto Central.
Modificación de Historia de Usuario Número: Ninguna	

Capítulo 2: Características, análisis y diseño del sistema

Usuario: Virgilio Suárez Bello	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1/3
Riesgo en Desarrollo: Medio	Puntos Reales: 1/3
Descripción: El objetivo fundamental que seguirá esta historia de usuario es procesar los datos para poder generar un reporte con la Información de los Pagos del Presupuesto Central.	
Observaciones: Para generar un reporte es necesario contar con la información mensual de los Pagos del Presupuesto Central.	
Prototipo de interface: Ninguna	

Tabla 2.4: Historia de Usuario Generar reporte de la Información Mensual de Pagos del Presupuesto Central

2.4 Lista de riesgos

La lista de riesgo se elabora para tener constancia de los aspectos negativos que pueden retrasar la culminación del producto. Para ello se describen los tipos de riesgos, el impacto que tendrá, la probabilidad de ocurrencia, el efecto que se espera que tenga. Para en el caso de que algunos de estos riesgos se conviertan en realidad se realiza un plan de mitigación, el cual describe que se debe hacer en cada uno de los casos.

Riesgo	Tipo de riesgo	Impacto	Descripción	Probabilidad	Efecto	Mitigación del riesgo
1	Personal	Productividad	El equipo de desarrollo no cuenta con la suficiente experiencia en las tecnologías JWebSocket.	Muy Alta	Serios	Brindar capacitación a los integrantes del equipo de desarrollo.
2	Personal	Atraso en las actividades	Poco conocimiento de la metodología de desarrollo SXP.	Muy Alta	Serios	Proporcionar documentación de la metodología y capacitación.

3	Tecnológico	Atraso en las actividades	Las computadoras destinadas a la implementación del producto no cuentan con las prestaciones necesarias.	Alta	Catastrófico	Priorizar las computadoras que tienen mayor necesidad de mejorar las prestaciones.
4	Tecnológico	Atraso en las actividades	Fallos continuos del fluido eléctrico.	Baja	Tolerable	Realizar salvadas continuas a la información.
5	Personal	Atraso en las actividades	Poca coordinación entre los integrantes del equipo y el personal de dirección.	Alta	Tolerable	Coordinar las tareas y realizar reuniones para ver los progresos que se han realizado.

Tabla 2.5: Lista de Riesgos

2.5 Diseño con metáforas

El diseño con metáforas describe como debería funcionar el sistema y define un conjunto de nombres que actúan como vocabulario para hablar sobre el dominio del problema. Es un diseño de la solución más simple que pueda ser implementado y funcionar en un momento dado. A partir de esto se genera el artefacto Modelo de Diseño que está compuesto por el diagrama de paquetes, el cual representa dicho diseño.

El diagrama de paquetes describe la distribución que tendrá el sistema una vez culminada su implementación. Se muestran las dependencias lógicas entre cada uno de los paquetes. Permite poder modelar cualquier sistema de software sin importar su nivel de complejidad. Se realiza una descripción de cada uno de los paquetes que componen el modelo de paquetes. A continuación se muestra el diagrama de paquetes para el sistema que se propone.

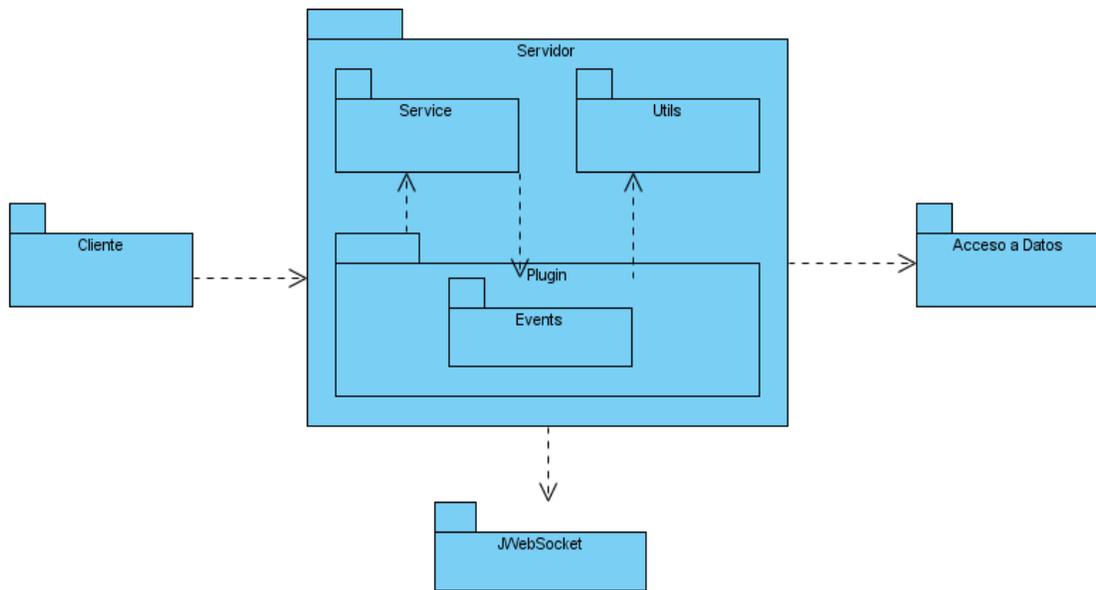


Fig. 2.3: Diagrama de paquetes del servidor para el módulo Presupuesto

Descripción:

El Diagrama de paquetes anterior se usa para reflejar la organización de los paquetes y sus elementos en el servidor para el módulo Presupuesto. El paquete Cliente representa la entrada y salida de los datos (lado cliente) y el paquete Acceso a Datos representa la persistencia de los datos (base de datos).

Los paquetes Plugins, Events, Service y Utils representan 4 de los paquetes en que se encuentra dividido el servidor para el módulo Presupuesto. El paquete JWebSocket representa las librerías del framework JWebSocket.

2.6 Diagrama de componentes

Los diagramas de componentes describen elementos físicos del sistema, así como las relaciones existentes entre ellos. Pueden estar conformados por archivos, paquetes, bibliotecas, entre otros. Se representa todos los tipos de elementos de software que van a ser parte de la aplicación. A continuación se muestra el diagrama de componentes para el sistema que se propone en este trabajo.

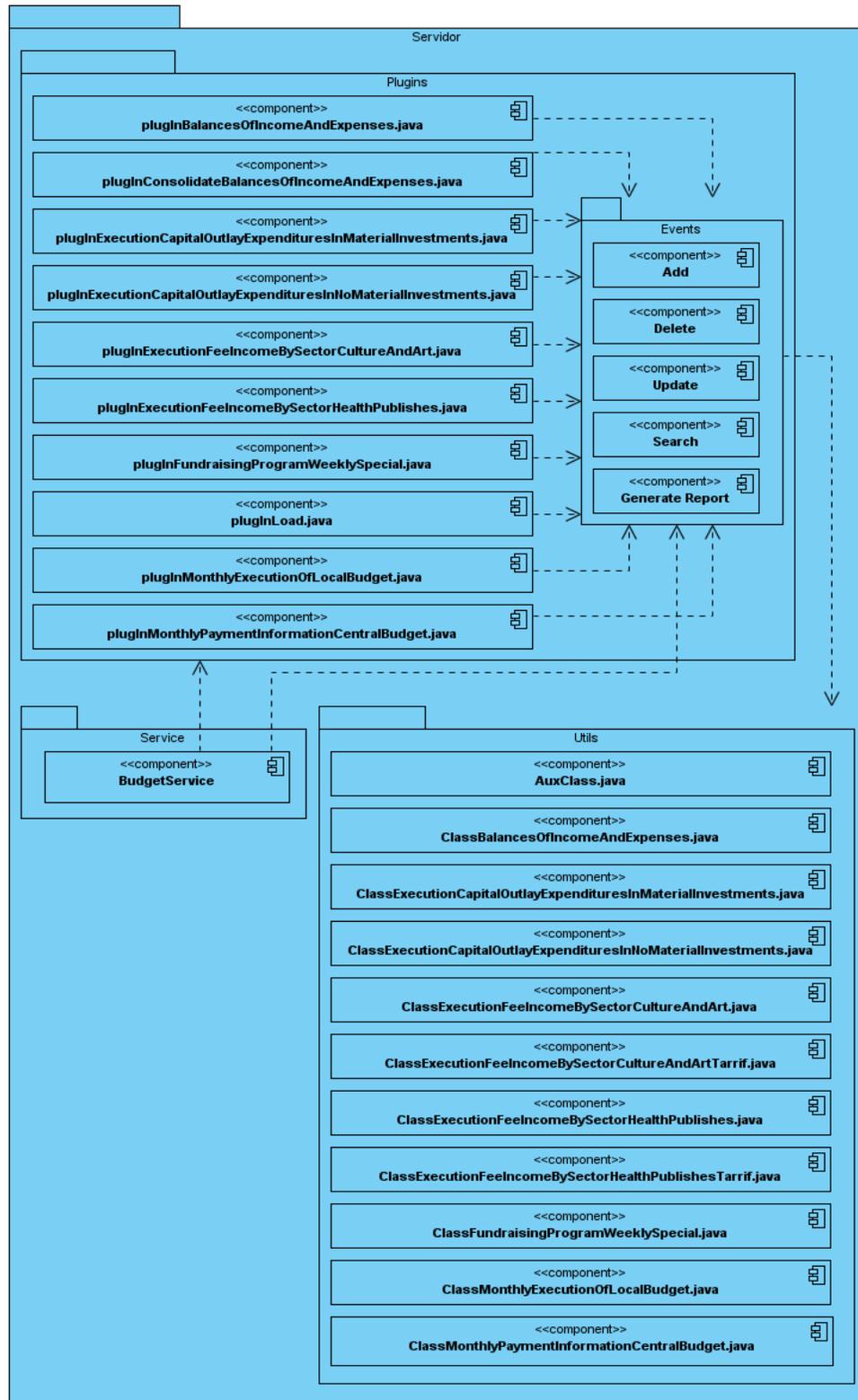


Fig. 2.4: Diagrama de componentes del servidor para el módulo presupuesto

2.7 Tareas de Ingeniería

Las Tareas de Ingeniería facilitan la definición de cada una de las actividades que se describen en las Historias de Usuarios. Permite conocer el tiempo que durará cada una y quien debe desarrollar dicha tarea facilitando poder realizar una buena estimación del tiempo de implementación.

Durante la etapa de desarrollo se generaron 45 Tareas de Ingenierías, en las cuales para cada uno de ellas, el tipo de tarea es desarrollo. Los gestionar información tienen con 4 tareas cada uno y en el caso de los generar reportes una sola tarea por HU. A continuación se brinda las Tareas de Ingeniería de las HU Gestionar la Información Mensual de Pagos del Presupuesto Central y Generar reporte de la Información Mensual de Pagos del Presupuesto Central.

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar los requerimientos para procesar los datos insertados de la Información Mensual de Pagos del Presupuesto Central.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/6
Fecha Inicio: 01/02/2012	Fecha Fin: 01/02/2012
Programador Responsable: Virgilio Suárez Bello	
Descripción: En esta tarea se implementa la función para procesar los datos insertados de la información mensual relacionada con los Pagos del Presupuesto Central.	

Tabla 2.6: Tarea de Ingeniería insertar la Información Mensual de Pagos del Presupuesto Central

Capítulo 2: Características, análisis y diseño del sistema

Tarea de Ingeniería	
Número Tarea: 1.2	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar los requerimientos para procesar los datos eliminados de la Información Mensual de Pagos del Presupuesto Central.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/6
Fecha Inicio: 02/02/2012	Fecha Fin: 02/02/2012
Programador Responsable: Virgilio Suárez Bello	
Descripción: En esta tarea se implementa la función para procesar los datos eliminados de la información mensual relacionada con los Pagos del Presupuesto Central.	

Tabla 2.7: Tarea de Ingeniería eliminar la Información Mensual de Pagos del Presupuesto Central

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar los requerimientos para procesar los datos modificados de la Información Mensual de Pagos del Presupuesto Central.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/6
Fecha Inicio: 03/02/2012	Fecha Fin: 03/02/2012
Programador Responsable: Virgilio Suárez Bello	
Descripción: En esta tarea se implementara la función para procesar los datos modificados de la información mensual relacionada con los Pagos del Presupuesto Central.	

Tabla 2.8: Tarea de Ingeniería modificar la Información Mensual de Pagos del Presupuesto Central

Capítulo 2: Características, análisis y diseño del sistema

Tarea de Ingeniería	
Número Tarea: 1.4	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar los requerimientos para procesar los datos buscados de la Información Mensual de Pagos del Presupuesto Central.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/6
Fecha Inicio: 04/02/2012	Fecha Fin: 04/02/2012
Programador Responsable: Virgilio Suárez Bello	
Descripción: En esta tarea se implementara la función para procesar los datos buscados de la información mensual relacionada con los Pagos del Presupuesto Central.	

Tabla 2.9: Tarea de Ingeniería buscar la Información Mensual de Pagos del Presupuesto Central

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: HU_10
Nombre Tarea: Implementar los requerimientos del procesamiento de los datos para poder generar un reporte con la Información de los Pagos del Presupuesto Central.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 14/03/2012	Fecha Fin: 15/03/2012
Programador Responsable: Virgilio Suárez Bello	
Descripción: En esta tarea se implementara la función del procesamiento de los datos para poder generar un reporte con la Información de los Pagos del Presupuesto Central.	

Tabla 2.10: Tarea de Ingeniería generar el reporte de la Información Mensual de Pagos del Presupuesto Central

2.8 Plan de liberación

En la metodología SXP se elabora un plan de liberación o release en la cual se especifican que historias de usuarios debe salir en cada una de las iteraciones. Esto permite la correcta ubicación de las distintas historias de usuarios en dependencia de las funcionalidades que el cliente desee tener en cada una de las liberaciones del producto. En este artefacto también se especifica la duración que van a tener cada una de las iteraciones y se cuenta con el tiempo total de implementación del producto.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 2	En la siguiente iteración se llevarán a cabo las historias de usuarios que gestionarán toda la información relacionada con el presupuesto de la dirección de Finanzas y Precios.	HU_1, HU_2, HU_3, HU_4, HU_5, HU_6, HU_7, HU_8, HU_9.	6 semanas
Iteración 3	En la siguiente iteración se llevarán a cabo las historias de usuarios que generarán todos los reportes relacionados con el presupuesto de la dirección de Finanzas y Precios.	HU_10, HU_11, HU_12, HU_13, HU_14, HU_15, HU_16, HU_17, HU_18.	3 semanas

Tabla 2.11: Plan de Liberación

Conclusiones del capítulo

A lo largo de este capítulo se revelaron los resultados de las disciplinas de Diseño e Implementación en el proceso de desarrollo del servidor para el módulo Presupuesto. Quedaron aprobados los requisitos funcionales necesarios para obtener un producto eficaz. Se desarrolló el modelo de dominio, la lista de reserva del producto, el diagrama de paquete y el diagrama de componentes, se diseñó y elaboro las tareas de ingeniería. Como complemento se brindó un plan de liberación y la lista de riesgos.

Capítulo 3: Validación de la propuesta de solución

En el presente capítulo se exponen los casos de pruebas unitarias que se le realizaron al servidor; estas fueron realizadas en cada una de las iteraciones, las cuales fueron imprescindibles para avanzar hacia la próxima iteración ya que para lograr un producto con calidad es necesario diseñar e implementar un plan de pruebas desde el principio, y así darle seguimiento a los cambios y desarrollar iterativamente. En este capítulo además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

3.1 Casos de pruebas

En la ingeniería del software, los casos de pruebas o Test Case son un conjunto de condiciones o variables bajo las cuales el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba.

3.1.1 Prueba unitaria

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código (o unidad mínima a ser probada, ejemplo; una clase, una función). Esto sirve para asegurar que cada uno de las partes funcione correctamente por separado, lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. La idea es escribir casos de prueba para cada función no trivial o método en la clase de forma que cada caso sea independiente del resto. Esto último es la esencia de una prueba unitaria: se prueba al componente de forma aislada a todos los demás. Con este propósito se realizaron un conjunto de pruebas unitarias definidas en el marco de trabajo y de las cuales a continuación se presenta una descripción.

Capítulo 3: Validación de la propuesta de solución

3.1.1.1 Casos de prueba para las funciones insertar

Caso de Prueba Unitaria	
Código Caso de Prueba: SP-01	Nombre: Insertar información de los modelos.
Nombre de la persona que realiza la prueba: Virgilio Suárez Bello	
Descripción de la Prueba: El objetivo de la siguiente prueba unitaria es insertar la información de cada uno de los modelos en la base de datos.	
Condiciones de Ejecución: Para poder ejecutar el archivo de prueba debe existir el archivo de configuración persistence-config.xml en la dirección descrita en el archivo de prueba AddTest.java.	
Descripción Archivo Prueba: El archivo de prueba contiene la carga de las configuraciones de la conexión de la base de datos, los objetos de la clase servicio a la cual se le inyecta las configuraciones de la conexión a la base de datos a partir del objeto del tipo GenericDao. En este archivo se crean los eventos para insertar la información de los modelos y se les pasan datos de pruebas. Se realiza el llamado a cada uno de los métodos de insertar que se desea probar.	
Entrada / Pasos de ejecución: Ejecutar el archivo con le nombre AddTest.java.	
Resultado Esperado: Si la prueba se realizó con éxito se espera el mensaje (True), en caso que el municipio insertado no se encuentre en la base de datos se espera el mensaje (No existe el municipio " nombre municipio " en la base de datos), en el caso que los indicadores de uno de los modelos no se encuentren en la base de datos se espera el mensaje (No existe el indicador " nombre indicador " en la base de datos), en el caso que no este en la base de datos la información del modelo padre se espera el mensaje (No existe un "nombre modelo padre" para " nombre del municipio").	
Evaluación de la Prueba: Satisfactoria	

Tabla 3.1: Caso de prueba unitaria insertar información de los modelos

Capítulo 3: Validación de la propuesta de solución

3.1.1.2 Casos de prueba para las funciones eliminar

Caso de Prueba Unitaria	
Código Caso de Prueba: SP-02	Nombre: Eliminar información de los modelos.
Nombre de la persona que realiza la prueba: Virgilio Suárez Bello	
Descripción de la Prueba: El objetivo de la siguiente prueba unitaria es eliminar la información de cada uno de los modelos en la base de datos.	
Condiciones de Ejecución: Para poder ejecutar el archivo de prueba debe existir el archivo de configuración persistence-config.xml en la dirección descrita en el archivo de prueba DeleteTest.java.	
Descripción Archivo Prueba: El archivo de prueba contiene la carga de las configuraciones de la conexión de la base de datos, los objetos de la clase servicio a la cual se le inyecta las configuraciones de la conexión a la base de datos a partir del objeto del tipo GenericDao. En este archivo se crean los eventos para buscar la información de los modelos para luego crear los eventos de eliminar con los datos de la búsqueda anterior. Se realiza el llamado a cada uno de los métodos de eliminar que se desea probar.	
Entrada / Pasos de ejecución: Ejecutar el archivo con le nombre DeleteTest.java.	
Resultado Esperado: Si la prueba se realizó con éxito se espera el mensaje (True), en caso de que no se inserte los identificadores por los que se eliminan se espera el mensaje (No existen datos con estas características).	
Evaluación de la Prueba: Satisfactoria	

Tabla 3.2: Caso de prueba unitaria eliminar información de los modelos

Capítulo 3: Validación de la propuesta de solución

3.1.1.3 Casos de prueba para las funciones modificar

Caso de Prueba Unitaria	
Código Caso de Prueba: SP-03	Nombre: Modificar información de los modelos.
Nombre de la persona que realiza la prueba: Virgilio Suárez Bello	
Descripción de la Prueba: El objetivo de la siguiente prueba unitaria es modificar la información de cada uno de los modelos en la base de datos.	
Condiciones de Ejecución: Para poder ejecutar el archivo de prueba debe existir el archivo de configuración persistence-config.xml en la dirección descrita en el archivo de prueba UpdateTest.java.	
Descripción Archivo Prueba: El archivo de prueba contiene la carga de las configuraciones de la conexión de la base de datos, los objetos de la clase servicio a la cual se le inyecta las configuraciones de la conexión a la base de datos a partir del objeto del tipo GenericDao. En este archivo se crean los eventos para buscar la información de los modelos para luego crear los eventos de modificar con los datos de la búsqueda anterior. Se realiza el llamado a cada uno de los métodos de modificar que se desea probar.	
Entrada / Pasos de ejecución: Ejecutar el archivo con le nombre UpdateTest.java.	
Resultado Esperado: Si la prueba se realizó con éxito se espera el mensaje (True), en caso que el municipio insertado no se encuentre en la base de datos se espera el mensaje (No existe el municipio " nombre municipio " en la base de datos), en el caso que los indicadores de uno de los modelos no se encuentren en la base de datos se espera el mensaje (No existe el indicador " nombre indicador " en la base de datos), en el caso que no este en la base de datos la información del modelo padre se espera el mensaje (No existe un "nombre modelo padre" para " nombre del municipio").	
Evaluación de la Prueba: Satisfactoria	

Tabla 3.3: Caso de prueba unitaria modificar información de los modelos

Capítulo 3: Validación de la propuesta de solución

3.1.1.4 Casos de prueba para las funciones generar reportes

Caso de Prueba Unitaria	
Código Caso de Prueba: SP-04	Nombre: Generar reporte de la información de los modelos.
Nombre de la persona que realiza la prueba: Virgilio Suárez Bello	
Descripción de la Prueba: El objetivo de la siguiente prueba unitaria es generar reportes de la información de cada uno de los modelos en la base de datos.	
Condiciones de Ejecución: Para poder ejecutar el archivo de prueba debe existir el archivo de configuración persistence-config.xml en la dirección descrita en el archivo de prueba GenerateReportTest.java.	
Descripción Archivo Prueba: El archivo de prueba contiene la carga de las configuraciones de la conexión de la base de datos, los objetos de la clase servicio a la cual se le inyecta las configuraciones de la conexión a la base de datos a partir del objeto del tipo GenericDao. En este archivo se crean los eventos para generar los reportes de la información de los modelos. Se realiza el llamado a cada uno de los métodos de generar reporte que se desea probar.	
Entrada / Pasos de ejecución: Ejecutar el archivo con el nombre GenerateReportTest.java.	
Resultado Esperado: Si la prueba se realizó con éxito se esperan todos los datos del reporte generado, en caso contrario se espera una lista vacía.	
Evaluación de la Prueba: Satisfactoria	

Tabla 3.4: Caso de prueba unitaria generar reportes

Capítulo 3: Validación de la propuesta de solución

3.1.1.4 Casos de prueba para las funciones que cargan los nomencladores

Caso de Prueba Unitaria	
Código Caso de Prueba: SP-05	Nombre: Cargar nomencladores
Nombre de la persona que realiza la prueba: Virgilio Suárez Bello	
Descripción de la Prueba: El objetivo de la siguiente prueba unitaria es cargar los nomencladores de la base de datos.	
Condiciones de Ejecución: Para poder ejecutar el archivo de prueba debe existir el archivo de configuración persistence-config.xml en la dirección descrita en el archivo de prueba LoadTest.java.	
Descripción Archivo Prueba: El archivo de prueba contiene la carga de las configuraciones de la conexión de la base de datos, los objetos de la clase servicio a la cual se le inyecta las configuraciones de la conexión a la base de datos a partir del objeto del tipo GenericDao. En este archivo se crean los eventos para cargar los nomencladores. Se ejecutan las funciones correspondientes.	
Entrada / Pasos de ejecución: Ejecutar el archivo con le nombre LoadTest.java.	
Resultado Esperado: Si la prueba se realizó con éxito se esperan todos los datos de los nomencladores, en caso contrario se espera una lista vacía.	
Evaluación de la Prueba: Satisfactoria	

Tabla 3.5: Caso de prueba unitaria cargar nomencladores

3.1.2 Resultado de las pruebas

Para el desarrollo de las pruebas se tuvo en cuenta 2 iteraciones de desarrollo. En ellas se realizaron 4 iteraciones de pruebas unitarias a cada uno de los requisitos. A continuación se muestra el código de uno de los métodos al que se le realizó la prueba unitaria para verificar su correcto funcionamiento. Se aplicaron las pruebas al servicio de nomencladores, a cada funcionalidad de este se le generan los resultados esperados y se comparan con los resultados arrojados.

Capítulo 3: Validación de la propuesta de solución

```
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {

    /**
     * Create conexion with xml
     */
    ApplicationContext appContext = new ClassPathXmlApplicationContext("persistence-config.xml");
    GenericDao dao = (GenericDao) appContext.getBean("genericDAO");

    /**
     * Create object type class BudgetService
     */
    BudgetService service = new BudgetService();
    service.setDao(dao);

    /**
     * Add new Year Plan
     */
    AddYearPlan addYearPlan = new AddYearPlan();

    addYearPlan.setCompanies("3");
    addYearPlan.setConstructionMaterials("3");
    addYearPlan.setDate("2011-09");
    addYearPlan.setFluidMilk("3");
    addYearPlan.setLandLease("3");
    addYearPlan.setMilkHauling("3");
    addYearPlan.setMovilizativasReserves("3");
    addYearPlan.setMunicipality("Caimito");
    addYearPlan.setPanProduct("3");
    addYearPlan.setPaymentType("3");
    addYearPlan.setReparationsCase("3");
    addYearPlan.setSubsidiesForLosses("3");
    addYearPlan.setTrainedExeso("3");

    System.out.println("----- Add Year Plan -----");
    System.out.println("Response: "+service.AddYearPlan(addYearPlan));
    System.out.println("-----");
}
```

Fig. 3.1: Código de pruebas unitarias funcionalidad adicional nuevo Plan Año

```
----- Add Year Plan -----
Hibernate: select this_.id_municipio as id1_0_0_, this_.cod_municipio as cod2_0_0_, this_.municipio as mu
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: select nextval ('mod_economia.eco_tss_fp_plan_ano_id_plan_ano_seq')
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Hibernate: select this_.id_indicador as id1_12_0_, this_.detalles_ind as detalles2_12_0_ from mod_economi
Hibernate: insert into mod_economia.eco_tss_fp_plan_ano (id_indicador, fecha, id_municipio, presup, id_pl
Response: true
-----
```

Fig. 3.2: Resultado final de la prueba adicional nuevo Plan Año

Capítulo 3: Validación de la propuesta de solución

En la siguiente tabla se muestran los resultados de las pruebas realizadas a cada uno de los requisitos funcionales por iteraciones.

Requisitos	1	1.1	2	2.1
Insertar la Información Mensual de Pagos del Presupuesto Central.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de Ingresos por el Cobro del Sector Salud Pública.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de Gastos de Capital en Inversiones Materiales.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de Gastos de Capital en Inversiones no Materiales.	Pasó	Pasó	Pasó	Pasó
Insertar la Información Mensual del Presupuesto Local.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de los Balances de Ingresos y Gastos.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de los Balances mensuales de Ingresos y Gastos.	Pasó	Pasó	Pasó	Pasó
Insertar la Información de la Recaudación de fondos del Programa Semanal Especial.	Pasó	Pasó	Pasó	Pasó
Eliminar la Información Mensual de Pagos del Presupuesto Central.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de Ingresos por el Cobro del Sector Salud Pública.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de Gastos de Capital en Inversiones Materiales.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de Gastos de Capital en Inversiones no Materiales.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información Mensual del Presupuesto Local.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de los Balances de Ingresos y	No	No Pasó	Pasó	Pasó

Capítulo 3: Validación de la propuesta de solución

Gastos.	Evaluado			
Eliminar la Información de los Balances mensuales de Ingresos y Gastos.	No Evaluado	No Pasó	Pasó	Pasó
Eliminar la Información de la Recaudación de fondos del Programa Semanal Especial.	No Evaluado	No Pasó	Pasó	Pasó
Modificar la Información Mensual de Pagos del Presupuesto Central.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de Ingresos por el Cobro del Sector Salud Pública.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de Gastos de Capital en Inversiones Materiales.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de Gastos de Capital en Inversiones no Materiales.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información Mensual del Presupuesto Local.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de los Balances de Ingresos y Gastos.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de los Balances mensuales de Ingresos y Gastos.	No Evaluado	Pasó	Pasó	Pasó
Modificar la Información de la Recaudación de fondos del Programa Semanal Especial.	No Evaluado	Pasó	Pasó	Pasó
Buscar la Información Mensual de Pagos del Presupuesto Central.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de Ingresos por el Cobro del Sector Cultura y Arte.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de Ingresos por el Cobro del Sector Salud Pública.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de Gastos de Capital en Inversiones Materiales.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de Gastos de Capital en Inversiones no Materiales.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información Mensual del Presupuesto	No Pasó	Pasó	Pasó	Pasó

Capítulo 3: Validación de la propuesta de solución

Local.				
Buscar la Información de los Balances de Ingresos y Gastos.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de los Balances mensuales de Ingresos y Gastos.	No Pasó	Pasó	Pasó	Pasó
Buscar la Información de la Recaudación de fondos del Programa Semanal Especial.	No Pasó	Pasó	Pasó	Pasó
Generar reporte de la Información Mensual de Pagos del Presupuesto Central.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la información de los Ingresos por el Cobro del Sector Cultura y Arte.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información de los Ingresos por el Cobro del Sector Salud Pública.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información de los Gastos de Capital en Inversiones Materiales.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información de los Gastos de Capital en Inversiones no Materiales.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información Mensual del Presupuesto Local.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información de los Balances de Ingresos y Gastos.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte de la Información de los Balances Mensuales de Ingresos y Gastos.	No Evaluado	No Evaluado	No Pasó	Pasó
Generar reporte la Información de la Recaudación de fondos del Programa Semanal Especial.	No Evaluado	No Evaluado	No Pasó	Pasó

Tabla 3.6: Pruebas de los requisitos por interacción

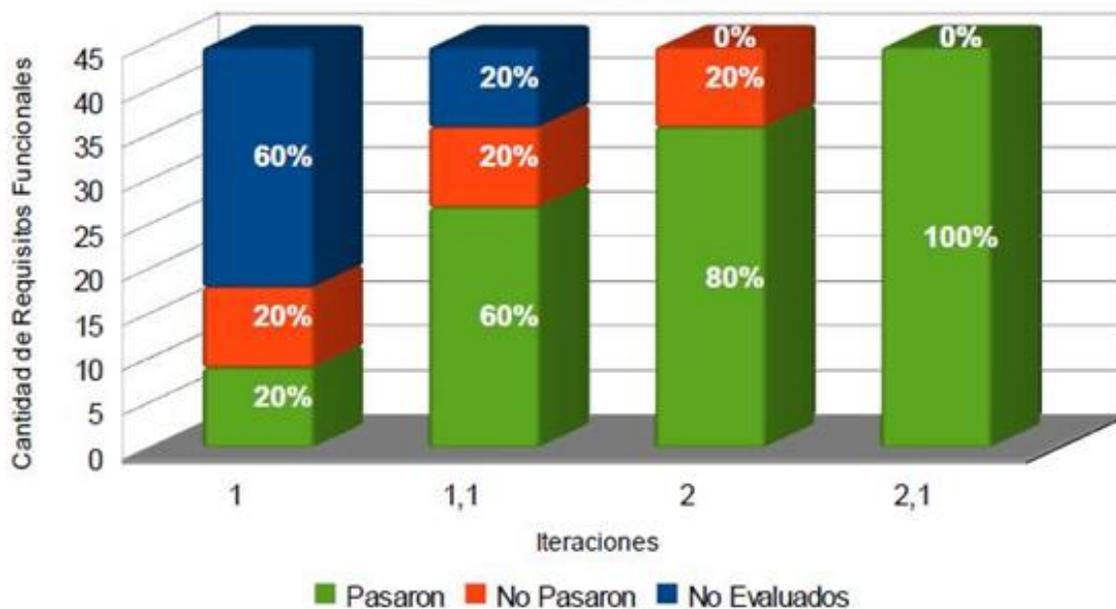


Fig. 3.3: Gráfica de las pruebas realizadas

En las 4 iteraciones de pruebas se encontraron y corrigieron errores llegando a un resultado positivo en todas las funcionalidades. Luego de realizar un análisis de los resultados de las pruebas en cada iteración se concluye que el paquete de servicios desarrollado cumple con todas las funcionalidades y requerimientos definidos, teniendo resultados correctos, altos índices de eficiencia, confiabilidad y rendimiento.

Validación de la solución

El grupo de Calidad de Software del Centro de Desarrollo de la Facultad Regional Mártires de Artemisa realizó la valoración de un grupo de parámetros de forma individual para validar la solución presentada. Dando como resultado satisfactoria en cada uno de los parámetros evaluados, obteniéndose un aval de calidad de software para la solución presentada. Para más información (**Ver anexo 2**).

Capítulo 3: Validación de la propuesta de solución

3.2 Resultados obtenidos

Como resultado de este trabajo el servidor para el módulo Presupuesto queda disponible en su versión 1.0. Se obtuvo un servidor que cumple con todas las especificaciones necesarias para procesar la información del Departamento Presupuesto de la Administración Provincial de Artemisa e integrarse con el Sistema Informativo de la Administración Provincial.

3.3 Funcionalidades obtenidas

Entre las principales funcionales que posee el servidor en su versión 1.0 se pueden mencionar:

- Permite procesar la información de cada uno de los modelos del Departamento Presupuesto de la Administración Provincial de Artemisa.
- Generar reportes por cada uno de los modelos insertados con anterioridad en la base de datos.

3.4 Aporte social y económico

El Departamento Presupuesto, actualmente se encarga de gestionar toda la información referente a la presupuestación en la provincia de Artemisa. Dicho trabajo es llevado en la institución de manera manual, lo que ha traído consecuencias negativas en cuanto a rendimiento, confiabilidad y eficiencia de la información que se manipula en el departamento. El servidor integrado al módulo brinda a los especialistas de la institución una vía más cómoda, eficiente y rápida de realizar su trabajo. Con el uso de este servidor para procesar la información se eliminarán los problemas antes planteados, además disminuirá notablemente el tiempo de procesamiento de la información, de esta forma se verán reducidas las pérdidas de datos contables que son fundamentales en la economía de la provincia y por consiguiente del país.

Capítulo 3: Validación de la propuesta de solución

Conclusiones del capítulo

En este capítulo se elaboraron y aplicaron los casos de pruebas unitarias para dar validez a la propuesta de solución. A partir de lo citado anteriormente se arriba a un prototipo funcional con todas las funcionalidades probadas. El servidor cuenta con todas las cualidades necesarias para su integración con el Sistema Informativo de la Administración Provincial de Artemisa.

Conclusiones

De manera general en el presente trabajo se desarrolló un servidor en su versión 1.0 para el módulo Presupuesto del Sistema Informativo de la Administración Provincial de Artemisa.

Con un carácter más específico se pueden destacar los siguientes aspectos:

- Se caracterizaron las soluciones de gestión del presupuesto tanto nacionales como internacionales existentes en la actualidad, evidenciándose la necesidad de diseñar una solución a la medida para estos fines.
- Fueron seleccionadas las herramientas, metodologías y tecnologías más apropiadas para el desarrollo de la propuesta de solución presentada, evidenciándose la importancia de la utilización de herramientas y tecnologías libres.
- Se desarrolló, asociado a la propuesta presentada, un detallado proceso de ingeniería, obteniéndose todos los componentes necesarios para el desarrollo del software.
- Se validó el funcionamiento de la propuesta a través de pruebas funcionales, mostrándose el aumento de los índices de rendimiento, confiabilidad y eficiencia de la solución presentada.

Recomendaciones

Se recomienda realizar futuras iteraciones de desarrollo del módulo Presupuesto para:

- Exportar a formato PDF los reportes generados.
- Importar los modelos en formato Excel.

Referencias Bibliográficas

1. ASSETS: Sistema de Gestión Integral. [citada 13 Diciembre 2011]. Disponible en: <<http://assets.co.cu/assets.asp>>.
2. Casa Consultora DISAIC. [citada 11 Enero 2012]. Disponible en: <<http://www.disaic.cu/modules.php?name=Products&file=index&pag=prod&id=28>>.
3. Córdova, Johanna. Presupuesto. Año 2004 [citada 7 de Enero 2008] Disponible en: <<http://www.monografias.com/trabajos21/presupuesto/presupuesto.shtml>>.
4. Definición de Cliente / Servidor - Significado y definición de Cliente / Servidor. [citada 8 de Mayo 2012]. Disponible en: <<http://www.mastermagazine.info/termino/4294.php>>.
5. Departamento Lenguajes y Ciencias de la Computación. [citada 2012]. Disponible en: <<http://www.lcc.uma.es/personal/pepeg/mates>>.
6. Gestión de la información. [citada 2012]. Disponible en: <<http://dieumsnh.qfb.umich.mx/gesinfo/>>.
7. Peñalver, Gladis. "MA-GMPR-UR2 Metodología ágil para proyectos de software libre." Universidad de las Ciencias Informáticas. Ciudad de La Habana, Cuba: 2008. [citada 94.] Disponible en: <http://bibliodoc.uci.cu/TD/TD_1309_08.pdf>.
8. Stallman, Richard. Software libre para una sociedad libre. Madrid: Traficantes de Sueños, 2004. Página 45.
9. SIMBIOTIC SYSTEM Software para la empresa desde la empresa. [citada 13 Diciembre 2011]. Disponible en: <<http://www.simbiotic-system.com/promocion.html>>.
10. Sistema Integrado de Gestión y Control de las Finanzas Públicas (SIGECOF) | DIGEMAFE. [citada 13 Diciembre 2011]. Disponible en: <<http://www.promafe.mf.gov.ve/?q=node/49>>.

11. Ministerio de Finanzas y Precios. Las Finanzas Públicas en Cuba, 1999 [citada: 4 de Enero 2008]. Disponible en: <http://www.cubagob.cu/des_eco/finanzas/finanzas.htm>.
12. Ministerio de Educación Superior. Perfeccionamiento de la actividad económica. Modelo de gestión Económico financiera del Ministerio de Educación Superior. Editorial Félix Varela. La Habana, Cuba, 2001. Página 78.
13. Ministerio de Finanzas y Precios. Resolución 04/2007 sobre la Metodología General para la Evaluación de la Ejecución del Presupuesto. Ministerio de Finanzas y Precios, La Habana, Cuba, 2007. Obra citada página. 3.
14. Wiley John. Introduction to Client / Server Systems: A Practical Guide for Systems Professionals. Páginas 105-120.

Bibliografía

- Consejo de Estado (1999). Decreto Ley 192/99 sobre el Sistema Presupuestario. Capítulo I y II. La Habana, Cuba.
- Definición de Cliente / Servidor - Significado y definición de Cliente / Servidor. [citada 8 de Mayo 2012]. Disponible en: <<http://www.mastermagazine.info/termino/4294.php>>.
- El Lenguaje de Modelado Unificado (UML). [citada 24 de Noviembre 2011]. Disponible en: <<http://www.docirs.cl/uml.htm>>.
- El software privativo « José Luis Alvarez Morales. [citado 23 de Noviembre 2011]. Disponible en: <<http://jalvarezm.wordpress.com/2006/09/30/el-software-privativo/>>.
- Extensible Markup Language (XML). [citada 12 de Enero 2012]. Disponible en: <<http://www.w3.org/XML/>>.
- Herramientas CASE. [citada 26 de Noviembre 2011]. Disponible en: <<http://gacycase.galeon.com/>>.
- Hibernate - JBoss Community. [citado 13 de Abril 2012]. Disponible en: <<http://www.hibernate.org/>>.
- jWebSocket » jWebSocket: “A dream is coming true...” [citada 12 de Enero 2012]. Disponible en: <<http://softwarelibre.hab.uci.cu/jwsblog/?cat=1&paged=3>>.
- jWebSocket - The Open Source Java WebSocket Server. [citada 11 de Enero 2012]. Disponible en: <<https://jwebsocket.org/>>.
- Ministerio de Educación Superior. Perfeccionamiento de la actividad económica. Modelo de gestión Económico financiera del Ministerio de Educación Superior. Editorial Félix Varela. La Habana, Cuba, 2001. Página 78.
- Ministerio de Finanzas y Precios. Las Finanzas Públicas en Cuba, 1999 [citada: 4 de Enero 2008]. Disponible en:

- <http://www.cubagob.cu/des_eco/finanzas/finanzas.htm>.
- Ministerio de Finanzas y Precios. Resolución 04/2007 sobre la Metodología General para la Evaluación de la Ejecución del Presupuesto. Ministerio de Finanzas y Precios, La Habana, Cuba, 2007. Obra citada página. 3.
 - Oracle Technology Network for Java Developers. [citada 12 de Enero 2012]. Disponible en: <<http://www.oracle.com/technetwork/java/index.html>>.
 - ¿Qué es el Software Libre? | Hispalinux. [citado 23 de Noviembre 2011]. Disponible en: <<http://www.hispalinux.es/SoftwareLibre>>.
 - Stallman, Richard. Software libre para una sociedad libre. Madrid: Traficantes de Sueños, 2004. Página 45.
 - Sarmiento Euclides, Alfredo (1989). Los presupuestos teoría y aplicaciones. Universidad Distrital. Bogotá, Colombia.
 - Servicios de Información del Transporte (2005). Control presupuestal. El control de la empresa III. No 267, Publicación del Servicio de información de SITRANS, La Habana, Cuba.
 - SIMBIOTIC SYSTEM Software para la empresa desde la empresa. [citada 13 de Diciembre 2011]. Disponible en: <<http://www.simbiotic-system.com/promocion.html>>.
 - Sistema Integrado de Gestión y Control de las Finanzas Públicas (SIGECOF) | DIGEMAFE. [citada 13 de Diciembre 2011]. Disponible en: <<http://www.promafe.mf.gov.ve/?q=node/49>>.
 - Software de presupuesto. Software gestión de presupuestos. [citado 23 de Noviembre 2011]. Disponible en: <<http://www.e-visualreport.com/software-presupuestos.html>>.
 - Spring Framework | SpringSource.org. [citada 11 de Enero 2012]. Disponible en: <<http://www.springsource.org/spring-framework>>.

- SpringSource.org. [citado 13 de Abril 2012]. Disponible en: <<http://www.springsource.org/>>.
- Struts. [citada 11 de Enero 2012]. Disponible en: <<http://javaweb.osmosislatina.com/curso/struts.htm>>.
- UML Diagramas. [citada 24 de Noviembre 2011]. Disponible en: <<http://www.jtmentor.com.ar/post/UML-Diagramas.aspx>>.
- Visual Paradigm - EcuRed. [citada 24 de Noviembre 2011]. Disponible en: <http://www.ecured.cu/index.php/Visual_Paradigm>.
- Visual Paradigm For Uml. [citada 24 de Noviembre 2011]. Disponible en: <<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>>.
- WebSockets: una dura lección. [citado 13 de Abril 2012]. Disponible en: <<http://www.desarrolloweb.com/articulos/websockets.html>>.
- Welcome. [citada 11 de Enero 2012]. Disponible en: <<http://struts.apache.org/>>.
- Welcome to NetBeans. [citada 13 de Diciembre 2011]. Disponible en: <<http://netbeans.org/>>.
- Wiley John. Introduction to Client / Server Systems: A Practical Guide for Systems Professionals. Páginas 105-120.

Anexos

Anexo1

Guía para las Entrevistas

- ¿Cómo funciona el departamento de Presupuesto?
- ¿Cómo se encuentra estructurado?
- ¿Cómo llenan los modelos? ¿Por qué vías lo hacen?
- ¿Quiénes tienen acceso a esos modelos?
- ¿Quiénes los pueden modificar o eliminar?

Entrevista dirigida al Departamento de Presupuesto de la Dirección de Finanzas y Precios de la CAP de Artemisa.

Objetivo: Conocer las deficiencias existentes para gestionar toda la información referente a los presupuestos.

- 1.- A su criterio ¿Cómo considera usted que se maneja la información en el dpto?
- 2.- ¿Cómo funciona el departamento?
- 3.- ¿Cómo se encuentra estructurado?
- 4.- ¿Cómo se lleva a cabo el llenado de los modelos?
- 5.- ¿Quiénes tienen acceso a esos modelos?
- 6.- ¿Qué tecnología emplean para llenar los modelos?
- 7.- ¿Cómo controlan toda esa información?

Anexo 2



Centro de Desarrollo

Aval de Calidad de Software

El grupo de Calidad de Software del Centro de Desarrollo de la Facultad Regional "Mártires de Artemisa" conformado por:

- **Asesora de Calidad:** Ing. Maidel Ojeda Cruz
- **Asesor de Tecnología:** Ing. Domma Moreno Dager
- **Especialista de Calidad:** Ing. Yenisleydi Rodríguez Martínez

emite el presente **Aval de Calidad de Software** en colaboración con los especialistas del Centro de Desarrollo a: Virgilio Suárez Bello, como resultado satisfactorio de su desempeño en la tareas asociadas al proyecto: Sistema Informativo de la Administración Provincial de Artemisa.

Para emitir el presente aval se valoraron un conjunto de elementos evaluados de manera individual teniendo en cuenta los parámetros de calidad de software del proyecto. A continuación se presenta los resultados en cada uno de los aspectos valorados:

Elementos evaluados	Resultado
Estandarización del código fuente del proyecto	Satisfactorio
Limpieza, organización y estructuración del código fuente	Satisfactorio
Funcionalidad e integración del sistema	Satisfactorio
Validación y seguridad de la información gestionada	Satisfactorio
Generación de todos los artefactos de la metodología SXP	Satisfactorio
Cumplimiento de las plantillas establecidas para cada artefacto	Satisfactorio
Ortografía, concordancia y redacción de cada uno de los artefactos	Satisfactorio
Modelación de diagramas con el uso de UML	Satisfactorio
Trazabilidad de los requisitos funcionales y no funcionales	Satisfactorio
Efectividad de los casos de pruebas definidos en el proyecto	Satisfactorio


Ing. Maidel Ojeda Cruz
Asesora de Calidad de SW




Msc. Yanila Vigil Regalado
Directora Centro de Desarrollo

Glosario de términos

AJAX: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

API: es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Crystal Reports: es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos).

Eventos: en la informática existe la programación orientada a eventos, en la que un evento es un mensaje de software que indica que algo ha ocurrido, como un tecleo o un click de un mouse. En el control de procesos, un evento es una ocurrencia que ha ocurrido y que ha sido registrado.

Excel: Microsoft Excel es una aplicación para manejar hojas de cálculo. Este programa es desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables.

Firewall: es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Full-duplex: es utilizado en las telecomunicaciones para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

HTML: siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), es

el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

IETF: en español (Grupo Especial sobre Ingeniería de Internet) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, como transporte, encaminamiento, seguridad.

J2EE: es una plataforma de programación, parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JSP: es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Microsoft SQL Server: es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle.

ORM: es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

Plugins: es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

POO: es programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos.

Proxy: en una red informática, es un programa o dispositivo que realiza una acción en representación de otro, que sirve para interceptar las conexiones de red que un cliente hace a un servidor de destino, por varios motivos posibles como seguridad, rendimiento, anoni-

mato, etc.

Servlets: se refería a pequeños programas que se ejecutan en un servidor.

SQL: es un lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas.

TCP: en español (Protocolo de Control de Transmisión), es uno de los protocolos fundamentales en Internet. Da soporte a muchas de las aplicaciones más populares de Internet (navegadores, intercambio de ficheros, clientes ftp,...) y protocolos de aplicación HTTP, SMTP, SSH y FTP.

Visual Basic: es un lenguaje de programación dirigido por eventos, desarrollado por el alemán Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados.

W3C: es un consorcio internacional que produce recomendaciones para la World Wide Web.