

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“SISTEMA WEB PARA EL CONTROL Y SEGUIMIENTO DE
LA EJECUCIÓN PRESUPUESTARIA”**
CASO: ASAMBLEA LEGISLATIVA DEL ESTADO PLURINACIONAL DE BOLIVIA

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE : Univ. Cesar Roberto Cuenca Díaz
TUTORA : Msc. Lic. Fátima Consuelo Dolz Salvador
REVISOR : Lic. Aldo Ramiro Valdez Alvarado

LA PAZ – BOLIVIA
2011

DEDICATORIA

Este proyecto está dedicado a mis padres, Roberto y Lucy, por todo el cariño, confianza y apoyo que me han brindado a lo largo de estos años, y a todos aquellos, que supieron entenderme y estrecharme su mano cuando la necesite.

Cesar Roberto Cuenca Díaz

AGRADECIMIENTOS

A Dios, por darme el regalo más grande: la vida, y permitirme compartir en todo momento con mis seres queridos.

A mí tutora Msc. Fátima Dolz Salvador, por su apoyo, enseñanzas y colaboración.

A mí revisor Lic. Aldo Valdez Alvarado por su continua orientación, observaciones y sugerencias.

Al Ing. Carlos Salgueiro, al Ing. Novak Zaballa, por depositar su confianza en mi persona y su asesoría, y por haberme brindado su amistad.

Al Lic. Waldo Humerez, por el apoyo y sugerencias, y sobre todo por la cordialidad, amabilidad y amistad brindada.

A mis amigos y amigas, por lo gratos momentos compartidos, y las palabras de aliento en tiempos difíciles.

A mis padres, Roberto Cuenca y Lucy Díaz, cuyo ejemplo ha sido siempre la luz que guía mi camino. A mis hermanos Luz, Roberto y Álvaro por su apoyo, por su amor incondicional y por creer en mí siempre.

A mis hermanos políticos, Carla, Patricia y Omar. A mis sobrinos, Jeancarla, Roberto y Gabriel.

Gracias a todos ustedes.

Cesar Roberto Cuenca Díaz

RESUMEN

El presente proyecto fue desarrollado para la Asamblea Legislativa del Estado Plurinacional de Bolivia (Honorable Cámara de Diputados), en atención al requerimiento de su Unidad de presupuestos dependiente de la Dirección Administrativa Financiera, con el propósito de automatizar los procesos inherentes a la ejecución presupuestaria programada y ejecutada por la mencionada unidad.

El SIGMA (Sistema Integrado de Gestión y Modificación Administrativa) proporciona la planificación, control y ejecución del presupuesto a nivel institucional, como máximo nivel de desagregación, por lo que la unidad de presupuesto debe llevar a cabo un control interno del presupuesto de la Cámara de Diputados.

La unidad de presupuestos no cuenta con un sistema para realizar dicho control y seguimiento, siendo consecuencia de esto, la demora en la estimación presupuestaria de una determinada unidad o partida presupuestaria.

Para el análisis, diseño e implementación del presente trabajo se ha hecho uso de la metodología ágil *XP* junto con el lenguaje de modelado web *WebML*.

Para el desarrollo del sistema se ha elegido el lenguaje del lado del servidor PHP, para el cliente se ha empleado tecnología *AJAX*. Como gestor de base de datos se ha escogido *MySQL*. Se han implementado los siguientes módulos: Módulo de Presupuesto, Módulo de Gastos, Módulo de modificación presupuestaria, Módulo de reportes, módulo de administración de usuarios, partidas y unidades.

Para la estimación de la calidad del producto final se ha hecho uso del estándar ISO 9126-1 que proporciona un modelo con 5 características a ser evaluadas: usabilidad, funcionalidad, fiabilidad, portabilidad y flexibilidad.

INDICE

CAPITULO 1 INTRODUCCIÓN

1.1	INTRODUCCION	1
1.2	ANTECEDENTES	2
1.2.1	INSTITUCIONALES	2
1.2.2	PROYECTOS SIMILARES	4
1.3	PROBLEMÁTICA	4
1.3.1	PROBLEMA CENTRAL	4
1.3.2	PROBLEMAS SECUNDARIOS	5
1.4	OBJETIVOS	6
1.4.1	OBJETIVO CENTRAL.....	6
1.4.2	OBJETIVOS ESPECIFICOS	6
1.5	JUSTIFICACIONES	6
1.5.1	JUSTIFICACION ECONOMICA.....	6
1.5.2	JUSTIFICACION SOCIAL.....	7
1.5.3	JUSTIFICACION TECNOLOGICA	7
1.6	LÍMITES Y RESTRICCIONES	7
1.7	ALCANCE	8
1.8	METODOLOGIA	9
1.9	APORTES	10

CAPITULO 2 MARCO TEORICO

2.1	INGENIERIA DE SOFTWARE	11
2.2	CICLOS DE VIDA DE DESARROLLO DEL SOFTWARE	12
2.2.1	MODELOS TRADICIONALES	13
2.2.2	MANIFIESTO ÁGIL.....	15
2.3	MADUREZ DE SOFTWARE EN METODOLOGÍAS ÁGILES	17
2.4	DIFERENCIAS ENTRE METODOLOGIAS TRADICIONALES Y AGILES...	18

2.5	EXTREME PROGRAMMING (XP)	19
2.5.1	CICLO DE VIDA DE XP	20
2.5.2	REGLAS Y PRÁCTICAS	21
2.5.2.1	HISTORIAS DE USUARIO	22
2.5.2.2	PLAN DE ENTREGAS	22
2.5.2.3	TAREAS	23
2.5.2.4	RECODIFICACIÓN	23
2.5.2.5	PRUEBAS UNITARIAS	23
2.5.2.6	PRUEBAS DE ACEPTACIÓN	23
2.6	INGENIERIA WEB	24
2.6.1	EVOLUCION	24
2.6.2	INGENIERÍA DE REQUISITOS PARA LAS APLICACIONES WEB	25
2.6.3	CLASIFICACION E IDENTIFICACIÓN DE ROLES	26
2.6.4	PRUEBAS	26
2.7	WEBML (WEB MODELING LANGUAGE)	27
2.7.1	DIAGRAMA DE ESTRUCTURA	28
2.7.2	DIAGRAMA DE COMPOSICIÓN	29
2.7.3	DIAGRAMA DE NAVEGACIÓN	31
2.7.4	MODELO DE PRESENTACIÓN	32
2.8	METRICAS DE CALIDAD ISO 9126-1	33
2.8.1	FUNCIONALIDAD	33
2.8.2	FIABILIDAD	33
2.8.3	USABILIDAD	33
2.8.4	EFICIENCIA	33
2.8.5	MANTENIBILIDAD	33
2.8.6	PORTABILIDAD	34
2.9	COCOMO (CONSTRUCTIVE COST MODEL)	34
2.9.1	RESTRICCIONES	34
2.9.2	ESTIMACIÓN	35
2.9.3	MODELO BASE DE ESTIMACIÓN	36
2.10	VALOR ACTUAL NETO	37

2.11	SISTEMA DE PRESUPUESTO	38
2.11.1	CONCEPTO Y OBJETO	38
2.11.2	ESTRUCTURA ORGANIZACIONAL DE LA HONORABLE CAMARA DE DIPUTADOS	39
2.11.3	ÁMBITO DE APLICACIÓN DE LAS NORMAS BÁSICAS.....	40
2.11.4	PRINCIPIOS DEL SISTEMA DE PRESUPUESTO.....	40
2.11.5	EJECUCION PRESUPUESTARIA.....	41
2.11.6	PARTIDAS PRESUPUESTARIAS	41
2.11.7	PROGRAMACIÓN DE LA EJECUCIÓN PRESUPUESTARIA.....	41
2.11.8	MODIFICACIONES AL PRESUPUESTO	41
2.11.9	SEGUIMIENTO DE LA EJECUCIÓN PRESUPUESTARIA	42

CAPITULO 3

MARCO APLICATIVO

3.1	DESARROLLO DEL SISTEMA WEB EMPLEANDO LA METODOLOGÍA XP Y EL MODELADO WEBML.....	43
3.2	FASE DE EXPLORACIÓN	44
3.2.1	HISTORIAS DE USUARIO	44
3.2.2	CLASIFICACIÓN E IDENTIFICACIÓN DE ROLES	48
3.3	FASE DE PLANIFICACIÓN	48
3.3.1	IDENTIFICACION DE TAREAS.....	49
3.3.2	PLAN DE ENTREGAS	55
3.4	FASE DE ITERACIONES.....	56
3.4.1	DIAGRAMA DE ESTRUCTURA	57
3.4.2	DIAGRAMAS DE COMPOSICIÓN.....	58
3.4.3	DIAGRAMAS DE NAVEGACIÓN.....	63
3.5	FASE DE PUESTA EN PRODUCCIÓN	67
3.5.1	HERRAMIENTAS DE DESARROLLO WEB.....	67
3.5.3	MODELO DE PRESENTACIÓN	68
3.6	FASE DE PUESTA EN PRODUCCIÓN	73
3.6.1	PRUEBAS DE ACEPTACIÓN.....	73

CAPITULO 4
CALIDAD DE SOFTWARE

4.1	ISO 9126-1	79
4.1.1	CONFIABILIDAD	79
4.1.2	FUNCIONALIDAD.....	80
4.1.2.1	NÚMERO DE ENTRADAS DE USUARIO	81
4.1.2.2	NÚMERO DE SALIDAS DE USUARIO.....	82
4.1.2.3	NÚMERO DE PETICIONES DE USUARIO	82
4.1.2.4	NÚMERO DE ARCHIVOS	83
4.1.2.5	NÚMERO DE INTERFACES EXTERNAS	83
4.1.3	FIABILIDAD	87
4.1.4	PORTABILIDAD.....	87
4.1.5	FLEXIBILIDAD	88
4.1.6	RESULTADOS.....	89
4.2	COCOMO.....	89
4.3	VAN-TIR.....	91

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1	CONCLUSIONES	92
5.2	RECOMENDACIONES	92
	BIBLIOGRAFIA	94
	ANEXO A – GLOSARIO	98
	ANEXO B – ANALISIS DE INVOLUCRADOS	100
	ANEXO C – ARBOL DE PROBLEMAS.....	101
	ANEXO D – ARBOL DE OBJETIVOS.....	102

CAPITULO 1

INTRODUCCIÓN

1.1 INTRODUCCION

El crecimiento acelerado que ha tenido la Web en estos últimos años está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas de donde proviene, lo que ha provocado que todas las actividades de las personas tiendan a depender de la Web. Internet pasó de ser una forma de diversión a un ente de desarrollo constante

En la última década se han elaborado numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos.

Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Por otro lado, la información, debe ser considerada como uno de los activos más importantes que posee cualquier institución pública o privada. Corresponde entonces que esta sea administrada cuidadosamente de forma eficaz y eficiente¹, ayudando de esta manera a la institución a alcanzar sus metas establecidas.

¹ En el desarrollo del presente documento, definimos eficiente como un adjetivo que hace referencia a conseguir un propósito determinado empleando los medios idóneos.

Las Instituciones públicas tienen la responsabilidad de administrar de manera óptima los recursos del estado, tal es el caso de la Asamblea Legislativa del Estado Plurinacional de Bolivia – Honorable Cámara de Diputados (**HCD**), que mediante su Dirección Administrativa Financiera (**DAF**) debe garantizar el manejo correcto de recursos que el estado le otorga.

Es aquí donde el uso de ordenadores adquiere gran importancia. Se quiere poner a consideración de la institución el análisis, diseño e implementación de un “Sistema Web para Control y Seguimiento de la Ejecución Presupuestaria”.

1.2 ANTECEDENTES

1.2.1 INSTITUCIONALES

En el marco de la Ley vigente 1178, que regula los sistemas de Administración y de Control de los Recursos del Estado, existe una relación con los sistemas nacionales de Planificación e Inversión Pública, cuyo objeto es el de programar, organizar, ejecutar y controlar la captación y el uso eficaz y eficiente de los recursos públicos para el cumplimiento y ajuste oportuno de las políticas, los programas, la prestación de servicios y los proyectos del Sector Público. La ejecución del presupuesto comprende los procesos administrativos de captación de recursos, de realización de desembolsos o pagos, y de ajustes al presupuesto aprobado por la Dirección Administrativa y Financiera (**DAF**).

Para tal fin, la DAF se rige por las Normas Básicas del Sistema de Contabilidad Integrada (**NBSCI**), estas constituyen el instrumento técnico que establece los principios y las normas contables que proporcionan al Sistema de Contabilidad Integrada (**SCI**) una base conceptual única y uniforme a ser observada en la preparación de Estados Financieros del sector público. El Subsistema de Presupuesto se rige en general por NBSCI, y específicamente, por las Normas Básicas del Sistema de Presupuesto (**NBSP**).

Según el Art. 25 de NBSP, la ejecución del presupuesto comprende los procesos administrativos de captación de recursos, de realización de desembolsos o pagos, y

de ajustes al presupuesto aprobado, sujetos a las regulaciones contenidas en las normas legales inherentes a la materia.

Por otro lado, según los Art. 29 y 30 de NBSP corresponde a la HCD elaborar y emitir en forma periódica, reportes de ejecución financiera del presupuesto. Los reportes de ejecución financiera deberán considerar la relación con la ejecución física prevista en el Programa de Operaciones Anual (**POA**), a la vez la HCD debe efectuar la evaluación de la ejecución del presupuesto, vinculando esta evaluación con el cumplimiento gradual y total de los objetivos de gestión previstos.

La DAF tiene como objetivo, integrar los registros presupuestarios, económicos, financieros y patrimoniales, para así poder determinar el costo de las acciones de la institución, por tal motivo, es de vital importancia la generación de información financiera clara y transparente que resulte de fácil comprensión para los usuarios y personas involucradas [Anexo B].

El Subsistema de Presupuesto, al tener como objetivo específico, posibilitar la instrumentación anual de los objetivos, políticas y metas definidas por las autoridades de mayor nivel nacional, se sirve de la contabilidad para informar sobre el logro o no de estos objetivos, dicha instrumentación posee dos momentos:

Momentos de Registro de Recursos y Momentos de Registro de Gasto.

El Subsistema de Presupuesto captura las transacciones con incidencia económica – financiera, en los distintos momentos del registro contable del proceso administrativo presupuestario, con el objetivo de producir información que, muestre la ejecución presupuestaria de recursos y gastos, permita al Órgano Rector del Sistema de Presupuesto evaluar la gestión presupuestaria de la HCD y posibilite el control del cumplimiento legal del presupuesto, de los resultados sobre la gestión de los recursos y programación de operaciones de la Institución.

1.2.2 PROYECTOS SIMILARES

- SISTEMA DE PRESUPUESTO AQUILES (Proyecto de Grado). Desarrollado por el Univ. Jesús Reynaldo Pérez Benavides, en el año 2010, para la Facultad de Ciencias Puras Y Naturales de La Universidad Mayor de San Andrés. Este sistema permite la elaboración de presupuesto mediante el manejo de programas, actividades y tareas en base a una estructura de tipo árbol, con múltiples métodos de inserción, eliminación y modificación.
- MODELO PARA LA INGENIERIA DE REQUERIMIENTOS (Tesis). Desarrollado por la Univ. Linnet Coral Herrera Zurita, en el año 2010. Este trabajo presenta un modelo de análisis en el proceso de elaboración de estados financieros, modelo basado en estándar de especificaciones de requerimientos de la IEEE.
- SISTEMA DE CALCULO DE COSTOS PRESUPUESTARIOS (Proyecto de Grado). Desarrollado por la Univ. Sarah Condori Kapa, en el año 2010, para la unidad de Infraestructura y Mantenimiento de la HCD. Este Sistema permite la cotización detallada para la unidad mencionada, determina el presupuesto y el cálculo en cómputos métricos para cada proyecto

1.3 PROBLEMÁTICA

1.3.1 PROBLEMA CENTRAL

En vista de lo anterior y en el ámbito del proyecto, surge la siguiente pregunta:

“¿Cómo se puede identificar y comprobar el manejo incorrecto de los recursos del Estado, y la evaluación gradual y total de los objetivos de gestión previstos, de forma eficiente y oportuna?”.

1.3.2 PROBLEMAS SECUNDARIOS

Bajo el marco normativo de la HCD, las transacciones se registran una sola vez, a partir de ese registro se obtienen todas las salidas de información presupuestaria financiera, económica y patrimonial que se necesite, con el objeto de lograr eficiencia en la administración se identifican los siguientes problemas actuales:

- La información necesaria para la elaboración del presupuesto no está clasificada, lo que implica el empleo de recursos adicionales a los planificados.
- La planificación anual presupuestaria carece de disponibilidad de información de la gestión anterior, lo que retarda la elaboración de dicho proceso.
- El proceso de Registro de la información en el Sistema de Contabilidad Integrada, realizado a través del “SISTEMA INTEGRADO DE GESTION Y MODERNIZACION ADMINISTRATIVA” (**SIGMA**), no proporciona el registro, proceso y resultado en forma desagregada y por unidad.
- El “*compromiso*”² y “*devengado*”³ de recursos, son procesos dependientes uno del otro, al ser elaborados de manera semiautomática (planillas Excel), existe una considerable demora al ajustar el presupuesto, siendo el trabajo más tedioso.
- El cálculo de la ejecución acumulada del presupuesto, excede al tiempo de solicitud de instancias superiores.

² Acto formal, que se refiere a una obligación jurídica existente o un acto de voluntad de disponer de una suma de dinero.

³ Operaciones que originan variaciones patrimoniales con incidencia en los resultados contables

1.4 OBJETIVOS

1.4.1 OBJETIVO CENTRAL

Implementar un Sistema Web capaz de automatizar el control y seguimiento del proceso administrativo, e integrar los registros presupuestarios de gastos por unidades, de forma desagregada para la Asamblea Legislativa del Estado Plurinacional de Bolivia.

1.4.2 OBJETIVOS ESPECIFICOS

- Permitir el registro de los diferentes datos paramétricos utilizados en la gestión de la ejecución presupuestaria, como ser: Gestiones; Periodos; Partidas de Gasto y Unidades Organizacionales.
- Prever en el desarrollo del proyecto, la generación de reportes de ejecución que sirvan como referente para la generación del Presupuesto Institucional de la siguiente gestión.
- Permitir la alimentación del Presupuesto Institucional de la Gestión correspondiente, en forma desagregada y por Unidad Ejecutora,
- Integrar las adiciones, disminuciones y traspasos en las diferentes partidas del Presupuesto por Unidad Ejecutora, para optimizar el reajuste presupuestario
- Emitir reportes de la ejecución acumulada del presupuesto, así como la ejecución presupuestaria en un determinado periodo.

1.5 JUSTIFICACIONES

1.5.1 JUSTIFICACION ECONOMICA

La Dirección Administrativa Financiera de la Asamblea Legislativa del Estado Plurinacional de Bolivia, realiza el proceso de registro de recursos y gastos por unidades en forma semiautomática (uso de planillas Excel) lo que implica un considerable retardo en las operaciones concernientes al control de la ejecución presupuestaria, retardo que se traduce en recursos perdidos, desde este punto de

vista, la implementación de un sistema de información constituye una solución mayor en beneficios que en costos.

1.5.2 JUSTIFICACION SOCIAL

La Dirección Administrativa Financiera de la institución es la unidad beneficiaria del proyecto, ya que este, reducirá los costos de la producción de la información contable, tanto en tiempo como en recursos tangibles. Los funcionarios de esta unidad están capacitados en el uso de plataformas web, por lo que el sistema puede desarrollarse y ejecutarse en la institución. Así también otras unidades serán beneficiadas indirectamente.

1.5.3 JUSTIFICACION TECNOLOGICA

La Dirección de Informática de la institución cuenta con el soporte técnico necesario para llevar a cabo la implementación del sistema web, cabe mencionar que actualmente están en funcionamiento sistemas web, dichos sistemas se encuentran almacenados en servidores y bajo la administración del personal correspondiente. Por otro lado el usuario final cuenta con los medios tecnológicos para hacer uso del producto final.

1.6 LÍMITES Y RESTRICCIONES

El presente proyecto de desarrollo web esta funcionalmente limitado por los requerimientos de la DAF y por la NBSP:

- El Sistema Web de Presupuesto no permitirá una conexión directa con el SIGMA, pero si facilitará la compatibilidad y verificación en términos contables.

- Las partidas presupuestarias, serán registradas según el *clasificador de partidas*⁴ vigente.
- Se permitirá la interacción del sistema únicamente con tres tipos de usuarios: Administrador, Operativo y Visitante.
- Los reportes podrán generarse entre un rango de fechas, teniendo como longitud máxima de dicho rango una gestión completa.
- El Sistema Web está comprendido por los siguientes módulos: Módulo de presupuesto, Módulo de Gastos y Módulo de Modificación Presupuestaria.

1.7 ALCANCE

El alcance del proyecto está íntimamente ligado a los principios contables que rigen a la ejecución presupuestaria, esta tiene como atribuciones específicas.

- El Sistema Web permitirá el registro de la planificación presupuestaria, la cual consiste en la asignación de recursos monetarios a las distintas unidades organizacionales de la HCD.
- El Sistema Web permitirá el registro de gastos por unidad organizacional y por partida, en forma desagregada.
- El Sistema Web permitirá el registro de modificación presupuestaria, la cual consiste en el ajuste financiero de partidas presupuestadas, conforme al reglamento NBSP
- El Sistema Web permitirá la adición de nuevas partidas presupuestarias.
- El Sistema Web permitirá la administración de usuarios, la cual incluye el registro de acciones de un determinado usuario desde su ingreso al sistema, hasta su salida del mismo.

⁴ El clasificador de partidas es un documento oficial con el que se regulan las partidas de gasto y su organización, para las instituciones públicas.

- El Sistema Web procesará y producirá información presupuestaria, patrimonial y financiera útil, con características de oportunidad, razonabilidad y confiabilidad para la toma de decisiones por los responsables de la gestión financiera pública y para terceros interesados en la misma, para esto el sistema deberá ser capaz de emitir reportes de ejecución presupuestaria y reportes de libro mayor por partida.

En tal sentido, el Sistema Web de Control y Seguimiento de la Ejecución Presupuestaria, no puede ni debe realizar procesos ajenos a los principios anteriormente mencionados.

1.8 METODOLOGIA

Existen numerosas metodologías para el desarrollo de software, en los últimos años la tendencia de desarrollo de software ha incursionado en el campo de las metodologías ágiles. El manifiesto para un desarrollo de software ágil expone:

“Estamos descubriendo formas mejores de desarrollar software, desarrollándolo y ayudando a otros a desarrollarlo. A través de este trabajo hemos empezado a valorar: individuos e interacciones sobre procesos y herramientas, software funcionando por sobre documentación completa, cooperación con el cliente sobre negociación de contratos, respuesta frente a cambios”.

En la actualidad las tres metodologías más usadas son: Cristal Methodologies, Scrum y Xp⁵ [Beck, K. 2003] (Extremme Programming) escogiendo esta última como la metodología aplicada para el desarrollo de este proyecto.

Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con

⁵ agilmanifesto.org, Traducción del inglés hecha por Straub Pablo.

requisitos muy cambiantes cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad.

1.9 APORTES

El aporte más importante del presente proyecto de grado es el desarrollo de un Sistema Web que va a permitir a los servidores públicos de la DAF administrar, ejecutar y controlar de manera eficiente y eficaz, los recursos que el estado otorga a la HCD, además de facilitar, mediante la producción de reportes financieros, la toma de decisiones al Director del Departamento.



CAPITULO 2

MARCO TEORICO

2.1 INGENIERIA DE SOFTWARE

IEEE⁶ define a la Ingeniería del Software como: *“la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software”* [Hans Van Vliet, 2002]. La ingeniería de software es una disciplina que concierne a todos los aspectos de la producción del software.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería del software. Los objetivos de la Ingeniería del Software son:

- Mejorar la calidad de los productos de software
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

La ingeniería de software se relaciona con varias disciplinas como Gerencia, Matemáticas, Gestión de proyectos, Gestión de la calidad, Ingeniería de sistemas, Ergonomía del software. Abarca un conjunto de áreas del conocimiento que son la base fundamental para el desarrollo de un proyecto software; estas áreas son: Requerimientos, Diseño, Construcción, Pruebas, Mantenimiento, Gestión de la

⁶ Corresponde a las siglas de (Institute of Electrical and Electronics Engineers) en español **Instituto de Ingenieros Eléctricos y Electrónicos**, es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías, dedicada a la estandarización.

configuración, Gestión de la ingeniería, Procesos de ingeniería, Herramientas y métodos de la ingeniería, Calidad del software; a continuación se detallan las más importantes: [Hans Van Vliet, 2002].

- a) **Requerimientos de software:** un requerimiento se define como una exigencia que debe ser cumplida para dar solución a un problema del mundo real. Contiene áreas como: especificación de requerimientos, análisis, validación, clasificación, negociación, etc.
- b) **Diseño del software:** es el proceso de definir la arquitectura, componentes, interfaces y otras características relativas al sistema como tal. Fundamentos, claves en el diseño, estructura y calidad son algunas de las áreas secundarias que comprende el diseño de software.
- c) **Construcción del software:** se refiere a la creación detallada de un sistema software a través de la combinación de codificación, verificación, pruebas de unidad, pruebas de integración y depuración.
- d) **Pruebas del software:** consiste en la verificación dinámica del comportamiento de un software ante un conjunto limitado de casos de prueba. Contiene áreas como: niveles de pruebas, técnicas de pruebas, etc.
- e) **Mantenimiento del software:** las actividades de mantenimiento comienzan teóricamente cuando el producto final es liberado, pero en la práctica empieza desde etapas mucho más tempranas, debido a los cambios en las necesidades del usuario a las que la aplicación debe adaptarse.

2.2 CICLOS DE VIDA DE DESARROLLO DEL SOFTWARE

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere). También se denomina a veces paradigma.

Entre las funciones que debe tener un ciclo de vida se pueden destacar:

- Determinar el orden de las fases del proceso de software
- Establecer los criterios de transición para pasar de una fase a la siguiente
- Definir las entradas y salidas de cada fase
- Describir los estados por los que pasa el producto
- Describir las actividades a realizar para transformar el producto
- Definir un esquema que sirve como base para planificar, organizar, coordinar y desarrollar

2.2.1 MODELOS TRADICIONALES

La ingeniería del software establece y se vale de una serie de modelos que establecen y muestran las distintas etapas y estados por los que pasa un producto software, desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento, hasta la retirada del producto. A estos modelos se les denomina “Modelos de ciclo de vida del software”. [Ian Sommerville, 2001]

Un modelo de ciclo de vida del software, describe las fases principales de desarrollo de software.

Existen varias alternativas de modelos de ciclo de vida, y la elección para un determinado proyecto es de suma importancia. A continuación se muestran algunos de los modelos tradicionales y más utilizados. [Ian Sommerville, 2001]

a) Modelo en Cascada

Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo se ve fluyendo hacia abajo (como una cascada) sobre las fases que componen el ciclo de vida.

b) Modelo Iterativo

Es un modelo derivado del ciclo de vida en cascada. Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras.

c) Modelo de Desarrollo Incremental

Este modelo se centra en la entrega de un producto operativo con cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

d) Modelo en Espiral

Este modelo de desarrollo combina las características del modelo de prototipos y el modelo en cascada. El modelo en espiral está pensado para proyectos largos, caros y complicados.

Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un modelo en cascada, pero no necesariamente ha de ser así.

e) Modelo de Prototipos

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la

construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar.

2.2.2 MANIFIESTO ÁGIL

A principios del 200 en las montañas Wasatch de Utah, se reunieron 17 desarrolladores convencidos de que era necesario un cambio en las metodologías “clásicas”⁷ de desarrollo de software.

Tras esta reunión se creó *The Agile Alliance*⁸, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”, estableciendo en él cuatro principios [Beck & Beedle, 2001].

- **Se valora a los individuos y las interacciones** sobre los procesos y las Herramientas.
- **Se valora a las aplicaciones que funcionan** sobre la documentación exhaustiva.
- **Se valora la colaboración del cliente** sobre las negociaciones contractuales.
- **Se valora la respuesta al cambio** sobre el seguimiento de un plan

A continuación se describen las metodologías ágiles más notables.

a) Scrum⁹

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas

⁷ Entre ellos se encontraban los creadores de XP, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development y Pragmatic Programming. Juntos proclamaron lo que se ha dado a conocer como el “Manifiesto for Agile Software Development”

⁸ Traducido al español: La Alianza Ágil, www.agilealliance.com

⁹ www.controlchaos.com

sprints, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto.

b) Crystal Methodologies¹⁰

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos.

El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas.

c) Dynamic Systems Development Method (DSDM)¹¹

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Presenta como características un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación.

d) Adaptive Software Development (ASD)¹²

Sus principales características son: iterativo, orientado a los componentes de software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente.

¹⁰ www.crystalmethodologies.org

¹¹ www.dsdm.org

¹² www.adaptivesd.com

e) Lean Development (LD)¹³

Utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente.

Su principal característica es introducir un mecanismo para implementar dicho cambios.

f) Extreme Programming (XP)

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes y jefes de proyecto).

El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres. Este mecanismo indica que, por ejemplo, si el cliente establece el alcance y la calidad, y el jefe de proyecto el precio, el grupo de desarrollo tendrá libertad para determinar el tiempo que durará el proyecto.

2.3 MADUREZ DE SOFTWARE EN METODOLOGÍAS ÁGILES

La Tabla 2.1 compara las distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad, etc.

También incorpora como referencia no ágil el Capability Maturity Model¹⁰ (CMM).

¹³ www.poppendieck.com

	CMM	ASD	Crystal	DSDM	FDD	LD	SCRUM	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
- Resultados	2	5	5	4	4	4	5	5
- Simplicidad	1	4	4	3	5	3	5	5
- Adaptabilidad	2	5	5	3	3	4	4	3
- Excelencia técnica	4	3	3	4	4	4	5	5
- Prácticas de colaboración	2	5	5	4	4	4	5	5
Media CM	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

Tabla 2.1 Ranking de “agilidad” (Los valores más altos representan una mayor agilidad)

Fuente: [INTECO, 2009]

Como se observa en la Tabla 2.1, todas las metodologías ágiles tienen una significativa diferencia del índice de agilidad respecto a CMM y entre ellas destacan ASD, Scrum y XP como las más ágiles.

2.4 DIFERENCIAS ENTRE METODOLOGIAS TRADICIONALES Y AGILES

Existen distintas propuestas metodológicas para el desarrollo de software. Por una parte tenemos aquellas propuestas más tradicionales que se centran en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos a producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos.

La filosofía de las metodologías ágiles consiste en centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

La Tabla 2.2 recoge esquemáticamente las principales diferencias entre metodologías tradicionales y metodologías ágiles.

Metodologías Ágiles	Metodologías Tradicionales
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos Principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 2.2 Diferencias entre metodologías ágiles y no ágiles

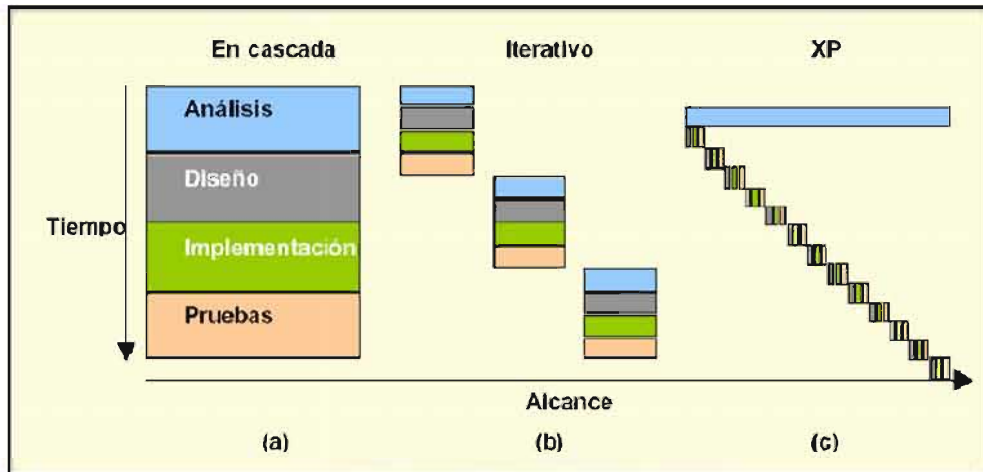
Fuente: [Beck, K. 2003]

A continuación se presenta un análisis más detallado de XP.

2.5 EXTREME PROGRAMMING (XP)

El ciclo de vida de un proyecto XP incluye, al igual que las otras metodologías, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto. [César F. & Acebal 2002]

Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan a XP (y que serán detalladas más adelante).



2.5.1.2 FASE DE PLANIFICACIÓN

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas o denominado también *Release Plan*¹⁴.

2.5.1.3 FASE DE ITERACIONES

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

2.5.1.4 FASE DE PUESTA EN PRODUCCIÓN

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa, lo que se denomina "fine tuning"¹⁵.

2.5.2 REGLAS Y PRÁCTICAS

La metodología XP tiene un conjunto importante de reglas y prácticas para sus distintas fases: [Beck, K. 2003]

¹⁴ Traducido al español: Plan de Entregas, típicamente esta fase consiste en una o varias reuniones grupales de planificación.

¹⁵ Fine Tuning, hace referencia a la fase donde no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.

2.5.2.1 HISTORIAS DE USUARIO

Las “Historias de usuarios” (*“User stories”*) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

2.5.2.2 PLAN DE ENTREGAS (*“Release Plan”*)

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, directores, etc.). XP denomina a esta reunión “Juego de planeamiento” (*“Planning game”*), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planeamiento).

Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores.

Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

2.5.2.3 TAREAS

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación.

2.5.2.4 RECODIFICACIÓN (Refactoring)

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

2.5.2.5 PRUEBAS UNITARIAS

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados.

Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

2.5.2.6 PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El usuario final debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada, estas son consideradas como “pruebas de caja negra”.

Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

2.6 INGENIERIA WEB

2.6.1 EVOLUCION

En los primeros años de la implementación del internet pocas personas tenían acceso al contenido de la red. Con el paso del tiempo se observó el gran potencial que podría tener este grandioso invento y comenzó un gran intercambio de información con variedad de temáticas, y con ello apareció la necesidad de compartir dicha información de manera más sencilla, entonces se creó el navegador “Mosaic” el cual facilitaba la visualización a través de hipertexto..

Durante años la Web ha sido un sitio informativo, un sitio de lectura en donde pocas personas escribían y los demás se dedicaban a leer. Según muchos autores esto es lo que podría llamarse la Web 1.0.

Visualizando la evolución para las personas que estamos del otro lado, ésta es significativamente grande y en ocasiones confusa; pasamos de páginas estáticas hechas con HTML, Java Script e imágenes al uso de CMS y estándares como XHTML, CSS, P3P por nombrar unas cuantas tecnologías, métodos o formas de ver las cosas. Todo enfocado al “Usuario Final” y a su interacción y producción. La Web crece a medida que los usuarios colaboran usando para ello la Web como una plataforma.

Esta evolución tecnológica es lo que muchos llaman ahora la Web 2.0; no es una versión, sino un concepto y una forma de ver las cosas. [EscalonaMJ, 2002]

2.6.2 INGENIERÍA DE REQUISITOS PARA LAS APLICACIONES WEB

Este apartado empieza dando una definición conjunta de requerimiento basándose tanto en el punto de vista del usuario como del desarrollador:

“Un requerimiento es una especificación de lo que debería ser implementado. Son descripciones de cómo el sistema debe comportarse, es una propiedad o un atributo y son sobre todo restricciones en el proceso de desarrollo del sistema“.

[Torres Bosh, 2008]

Las especificaciones de los requerimientos no incluyen detalles de diseño e implementación, información sobre la planificación del proyecto, o planes de pruebas. Los requerimientos de aplicaciones al igual que los de software tradicional, comprenden 3 niveles distintos: requerimientos del negocio, del usuario y funcionales. [Brambilla &Manolescu, 2002].

- i. **Requerimientos del Negocio:** Representan los objetivos de la empresa o usuario que solicita el sistema.
- ii. **Requerimientos del Usuario:** Describen lo que el usuario de la aplicación web será capaz de realizar una vez que se haya implementado el mismo.
- iii. **Requerimientos Funcionales:** Especifican las funcionalidades del software que los desarrolladores deben construir para que el usuario logre cumplir sus objetivos planteados.

La Ingeniería de Requerimientos (IR) comprende actividades que son críticas para el éxito de la Ingeniería Web. Los requerimientos incorrectos, incompletos o ambiguos pueden derivar en severas dificultades en el desarrollo, incluso pueden causar la cancelación de proyectos.

La IR trata con los principios, métodos y herramientas para extraer, describir, validar y gestionar los requerimientos. [Brambilla &Manolescu, 2002]

2.6.3 CLASIFICACION E IDENTIFICACIÓN DE ROLES

La entrevista es una técnica muy utilizada puesto que permite conseguir un mejor entendimiento del problema y por lo tanto de las posibles soluciones. En la figura 2.3 podemos apreciar un modelo de entrevista.

Se identifica y clasifica los tipos de actores de la aplicación, ello implica identificar a las clases y sub-clases de actores como se puede observar en la figura 2.4

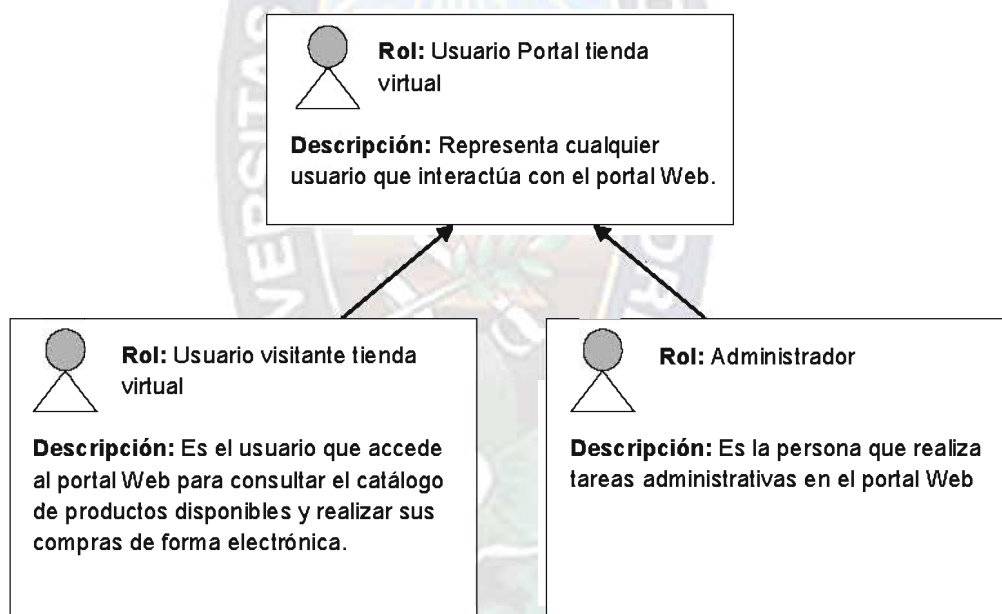


Figura 2.4: Clasificación e Identificación de roles

Fuente: [Soto & Palma, 2004]

2.6.4 PRUEBAS

Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo pueda comprender los resultados obtenidos, para ello se hará uso de especificación de casos de prueba.

CASO DE PRUEBA	Ver notas a través del código de alumno	
ROLES	Alumno	
TAREAS	T1	
PRECONDICIONES	El alumno debe estar registrado en la base de datos del colegio	
DESCRIPCIÓN		
PETICIONES	RESPUESTAS	
1. El usuario ingresa su código “Si no recuerda su código de alumno, puede buscarlo en (Caso de uso: Buscar código de alumno)” 2. El usuario ingresa su contraseña.	3. El sistema verifica el código y contraseña. 4. Si la contraseña y el código no son válidos, el sistema informa que los datos son inválidos y vuelve a pedir el código y contraseña.	
POSTCONDICIONES	El portal debe estar listo para generar una vista preliminar del formulario de inscripción a eventos llenado con los datos del solicitante.	

Figura 2.5: Plantilla para la especificación de Casos de Prueba

Fuente: [Soto & Palma, 2004]

2.7 WEBML (Web Modeling Language¹⁶)

WebML es un lenguaje de modelado gráfico utilizado para apoyar las actividades del diseño de sitios Web. Define varios tipos de diagramas: de estructura, composición y navegación.

WebML Se basa en la distribución de nodos en los niveles del hipertexto sobre las páginas del nivel de presentación. [Ceri et al., 2000].

¹⁶ Traducido al español: Lenguaje de Modelado Web

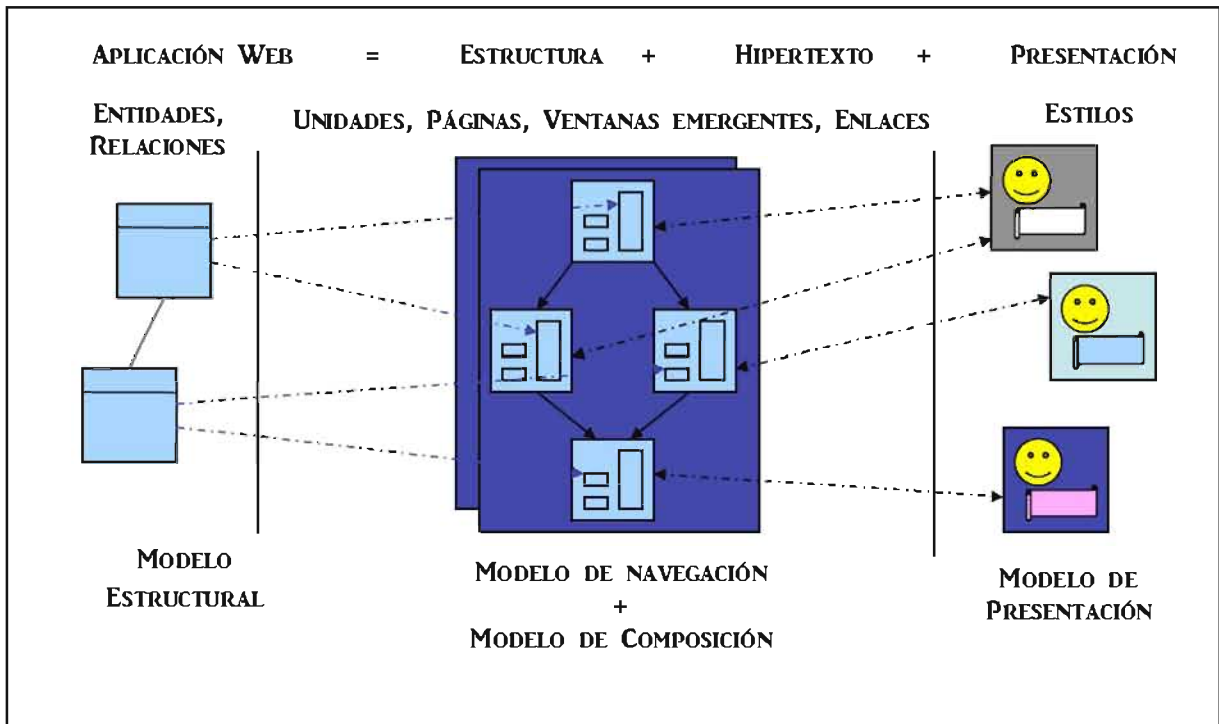


Figura 2.6: Enfoque WebML

Fuente: [Pastor, 2008]

2.7.1 DIAGRAMA DE ESTRUCTURA

El modelo estructural que describe la organización conceptual de datos; este modelo es compatible con:

- Modelo Entidad-Relación
- Diagramas de clases en UML.

En la figura 2.7 se muestra un ejemplo de un diagrama de estructura, el cual consiste de cuatro entidades:

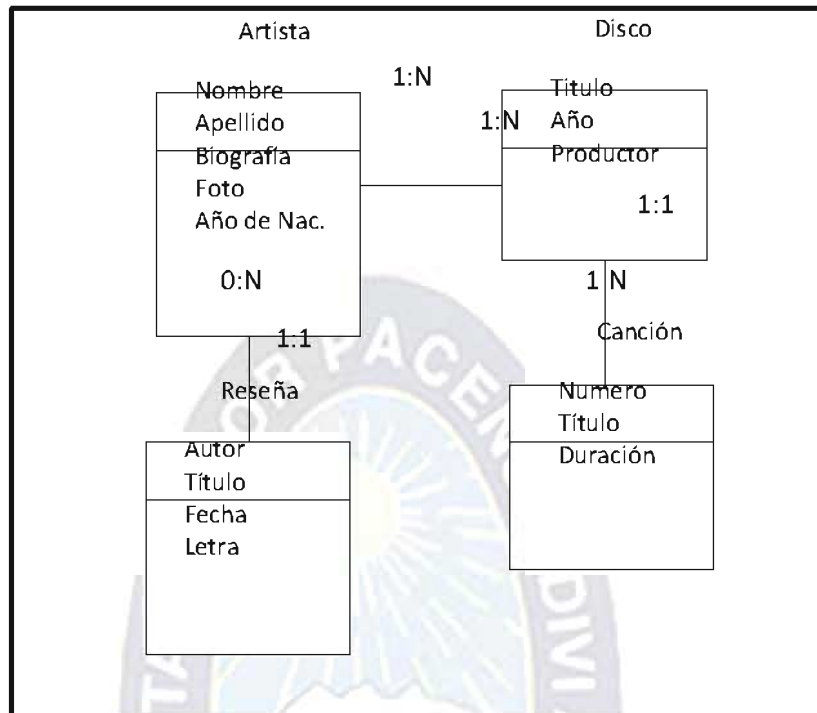


Figura 2.7: Diagrama de estructura

Fuente: [Ceri et al., 2000]

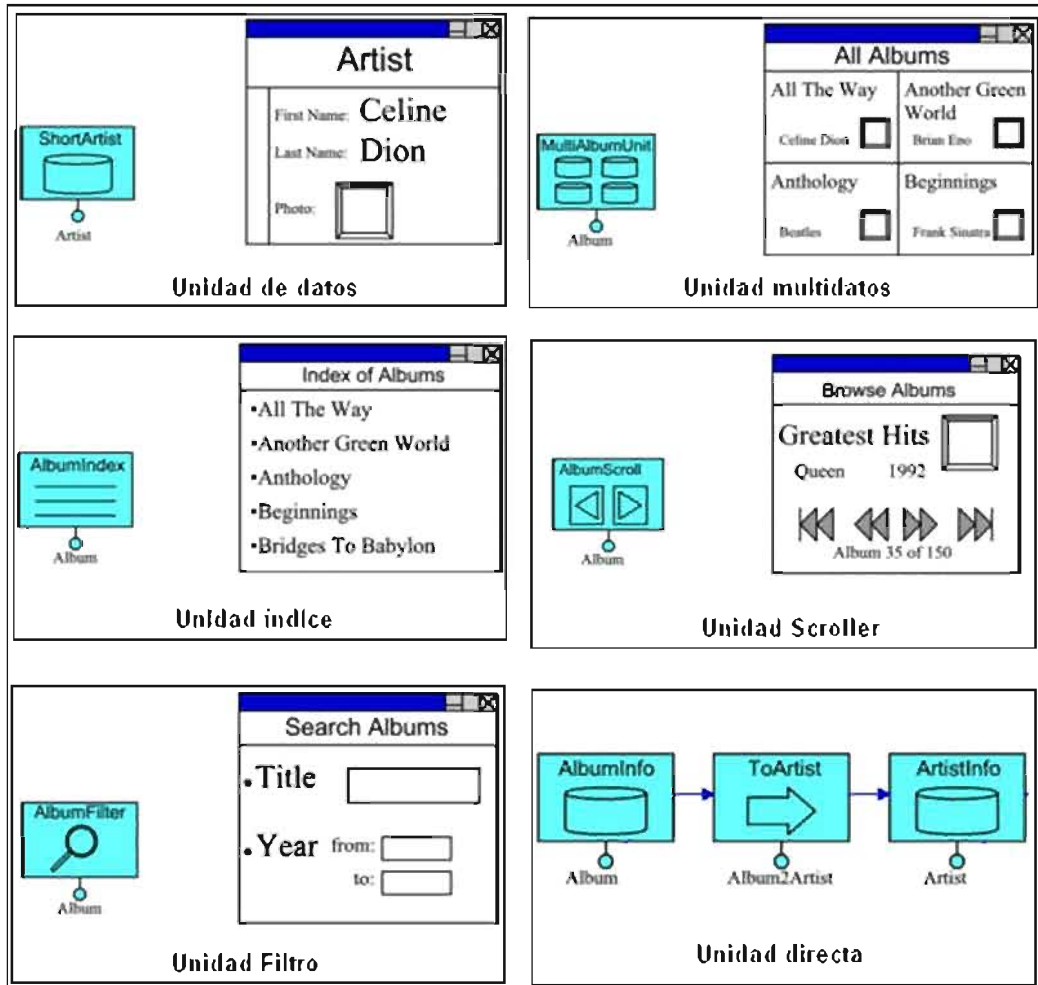
2.7.2 DIAGRAMA DE COMPOSICIÓN

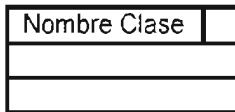
En este modelo se identifican qué páginas componen el hipertexto y con qué contenido; y el modelo de navegación que expresa cómo se enlazan las páginas para formar el hipertexto. Para estos y los demás modelos propuestos, WebML define una notación propia.

El propósito del diagrama de composición es definir los nodos que forman parte del hipertexto contenido en el sitio Web, es decir, se especifican las páginas y las unidades (elementos atómicos de información que deben aparecer en el sitio Web) que componen el sitio Web.

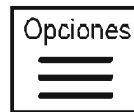
WebML soporta seis tipos de unidades que pueden ser usadas para componer hipertexto: [Ceri et al., 2000]

- a) **Unidades de Datos:** Muestran información sobre un solo objeto, son definidas para seleccionar una mezcla de información. Para definir una unidad de datos se requiere la indicación del concepto al cual se refiere la unidad y la selección de los atributos de la unidad.
- b) **Unidad Multidatos:** Muestra información sobre un conjunto de objetos, presenta múltiples instancias de una entidad o componente. Una unidad multidatos tiene dos partes: el contenedor que incluye las instancias que se desean mostrar y la unidad de datos usada para la presentación de cada instancia.
- c) **Unidad Índice:** presenta múltiples instancias de una unidad o componente como una lista, esta unidad tiene dos partes principales: el contenedor que incluye las instancias que se desean mostrar (las instancias deben ser una entidad, una relación o un componente) y los atributos usados como clave del índice.
- d) **Unidad Scroller:** provee comandos para desplazarse a través de los objetos en un contenedor. Esta unidad es normalmente usada junto con una unidad de datos, la cual representa el elemento actual visualizado del contenedor.
- e) **Unidad Filtro:** provee campos de entrada para buscar los objetos en un contenedor, esta unidad es normalmente usada junto con una unidad índice o multidatos, la cual muestra los objetos que coinciden con las condiciones de búsqueda.
- f) **Unidad Directa:** Expresa un tipo particular de índice, el cual contiene un solo objeto asociado a otro objeto por una relación uno a uno.

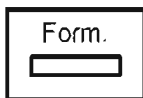




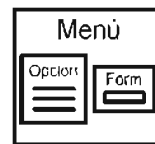
Clase navegacional o nodo



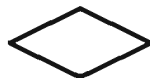
Opciones
Muestra un conjunto de enlaces y permite el acceso directo a cualquiera de ellos.



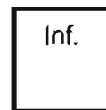
Formulario / ingreso de datos
Permite el ingreso de datos que sirven para seleccionar las colecciones de opciones.



Menú
Un menú puede contener opciones o formularios.



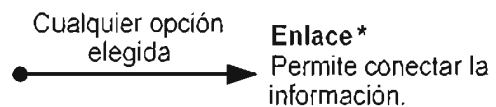
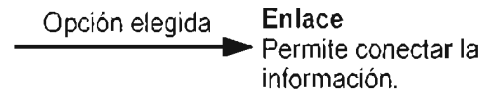
Condición
Permite controlar si se ha cumplido alguna condición y encaminar la navegación.



Documento
Este icono representa a una página donde se muestra solo información, como ser lista de materiales.



Lista de elementos
Posee la misma función que el elemento "opciones", simplemente se hace la distinción de que se presenta una lista de elementos del mismo tipo.



Este modelo usualmente asocia construcciones de un lenguaje de presentación (por ejemplo HTML) con cada unidad de presentación (por ejemplo: páginas, marcos), dando como resultado vistas funcionales de la aplicación Web. [Ceri et al., 2000]

2.8 METRICAS DE CALIDAD ISO 9126-1

El modelo de calidad ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características.

2.8.1 FUNCIONALIDAD

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas: Idoneidad, exactitud, interoperabilidad, seguridad y cumplimiento de normas.

2.8.2 FIABILIDAD

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido: Madurez, capacidad de recuperación y tolerancia a fallos.

2.8.3 USABILIDAD

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios: Aprendizaje, comprensión, operatividad.

2.8.4 EFICIENCIA

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas: Comportamiento en el tiempo y comportamiento de recursos

2.8.5 MANTENIBILIDAD

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software: Estabilidad, facilidad de análisis, facilidad de cambio y facilidad de pruebas.

2.8.6 PORTABILIDAD

Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra: Capacidad de instalación, capacidad de reemplazo, adaptabilidad y Co-Existencia

2.9 COCOMO (Constructive Cost Model)¹⁷

Como se conoce, una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos.

En el año 1981 Barry Boehm publica el modelo COCOMO, acorde a las prácticas de desarrollo de software de aquel momento [Boehm 1981].

COCOMO Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

2.9.1 RESTRICCIONES

- Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.

¹⁷ Traducido al español: "Modelo Constructivo de Costos"

- La medición por líneas de código no es válida para orientación a objetos.
- Utilizar este modelo puede resultar un poco complicado, en comparación con otros métodos (que también sólo estiman).

2.9.2 ESTIMACIÓN

Las ecuaciones a utilizar son:

$$E = a(KI)^b * m(X), \text{ en persona-mes}$$

$$Tdev = c(E)^d, \text{ en meses}$$

$$P = E / Tdev, \text{ en personas}$$

Donde:

- E es el esfuerzo requerido por el proyecto, en persona-mes
- Tdev es el tiempo requerido por el proyecto, en meses
- P es el número de personas requerido por el proyecto
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo.
- KI es la cantidad de líneas de código, en miles.
- m(X) Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en **modos** que representan el tipo de proyecto, y puede ser:

a) **Modo orgánico**: un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía

desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).

b) Modo semilibre o semiencajado: corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

c) Modo rígido o empotrado: el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

2.9.3 MODELO BASE DE ESTIMACIÓN

Luego de estimar el esfuerzo veremos cómo se ajusta el modelo para:

- Estimaciones más tempranas, correspondiente al diseño temprano (Pre-arquitectura).
- Mantenimiento.
- Estimación de número de defectos esperados.

a) Estimación de puntos de función

Debemos considerar que cada pantalla o reporte a generar equivale a 8 puntos de función (de ahora en adelante pf).

A continuación una referencia más detallada:

- Cada pantalla de consulta vale 4 pf.
- Cada pantalla que carga datos vale 5 pf.
- Cada reporte vale 5 pf
- Cada tabla de una base de datos vale 10 pf
- Cada interfaz con otro sistema vale 6 pf.

La simplificación consiste en considerar que toda pantalla reporte vale 5 puntos función y que sólo el 60% de los puntos de función provienen de pantalla el resto vienen de tablas o archivos lógicos.

b) Estimación de fórmulas básicas de esfuerzo calendario y personal requerido

La fórmula básica de esfuerzo (PM, persona-mes):

$$PM = 2.94 * \text{Tamaño}^E * [$$

Donde

- Tamaño se mide en KSLOC
- EMI corresponde a los factores de ajuste de costo.
- E es el exponente no lineal, calculado según:
- $E = 0.91 + 0.01 * \sum SFi$
- SFi son parámetros de costo (cost-drivers).

2.10 VALOR ACTUAL NETO

También conocido valor actualizado neto (en inglés *Net present value*), cuyo acrónimo es VAN (en inglés NPV), es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

El método de valor presente es uno de los criterios económicos más ampliamente utilizados en la evaluación de proyectos de inversión. Consiste en determinar la equivalencia en el tiempo 0 de los flujos de efectivo futuros que genera un proyecto y comparar esta equivalencia con el desembolso inicial. Cuando dicha equivalencia es

mayor que el desembolso inicial, entonces, es recomendable que el proyecto sea aceptado.

La fórmula que nos permite calcular el Valor Actual Neto es:

$$VAN = \sum_{t=1}^n \frac{V_t}{(1+k)^t} - I_0$$

V_t representa los flujos de caja en cada periodo t .

I_0 es el valor del desembolso inicial de la inversión.

n es el número de períodos considerado.

El tipo de interés es k . Si el proyecto no tiene riesgo, se tomará como referencia el tipo de la renta fija, de tal manera que con el VAN se estimará si la inversión es mejor que invertir en algo seguro, sin riesgo específico. En otros casos, se utilizará el coste de oportunidad.

Cuando el VAN toma un valor igual a 0, k pasa a llamarse TIR (tasa interna de retorno). La TIR es la rentabilidad que nos está proporcionando el proyecto.

2.11 SISTEMA DE PRESUPUESTO

2.11.1 CONCEPTO Y OBJETO

El Sistema de Presupuesto es un conjunto ordenado y coherente de principios, disposiciones normativas, procesos e instrumentos técnicos, que tiene por objeto la formulación, aprobación, ejecución, seguimiento y evaluación, de los presupuestos públicos, según los programas y proyectos definidos en el marco del Plan General de Desarrollo Económico y Social de la República.

Asimismo tiene por objeto la generación y obtención de información confiable, útil y oportuna sobre la ejecución presupuestaria, para la evaluación del cumplimiento de la gestión pública, en este caso, la gestión pública de la HCD.



ASESORIA LEGAL	DAF	DIR. INFORMACION	DIR. INFORMATICA	DIR. PROTOCCOLO	DIR. TRANSPARENCIA	DIR. REDACCION	CFMA	RRHH
DIR. GRAL. ASUNJUR.	DAF	DIR. INFORMACION	DIR. INFORMATICA	DIR. PROTOCCOLO	DIR. TRANSPARENCIA	DIR. REDACCION	DIR. PLANIF.	BIENESTAR SOC.
	DEP. B&YS&						CFMA	DIR. RRHH
	DEP. FINANC.						SEC. VENT. UNICA	SEC. PLAN. IAS
	SEC. ALMACENES							SEC. REG. PERSONAL
	SEC. HEMOCLO							U. ADM. PERSONAL
	SEC. INF. Y MANT.							U. DESARROLLO EVA Y CAP.
	SEC. SERTEC							
	U. COMPRAS Y CONT.							
	U. ACTIVOS FIJOS							
	U. CONTABILIDAD							
	U. PRESUPUESTOS							
	U. TESORERIA							

AUD.INTERNA	BANCADAS	BRIGADAS	COMISIONES	DIRECTIVA	PARLAMENTOS	SECRETARIA GENERAL
DEP. AUD.INTERNA	IMAS	BENI	COSNTITUCION LEGISLACION	1RA SECRET.	AMAZONICO	SEC.REDACCION
	CONVERGENCIA	CHUQUISACA	DERECHOS HUMANOS	1RA. VICE	ANDINO	SEC.SEGUIMIENTO Y COORD.
	UN	COCHABAMBA	ECO.PLURAL	2DA. VICE	INDIGENA ORIGINARIO	SECRETARIA GRAL.
		LA PAZ	EDUC. Y SALUD	2DA. SECRET.	LATINOAMERICANO	U.APOYO A LA TECNICA LEGIS.
		ORURO	ETICA	3RA.SECRET.	MUNDIAL	U.DE REVISION
		PANDO	GOBIERNO DEFENSA	4TA. SECRET.		
		POTOSI	JUSTICIA PLURAL	PRESIDENCIA		
		STA. CRUZ	NACIONES Y PUEB.INDIGENAS			
		TARIJA	ORGANIZ.TERRITORIAL			
			PLANIFICACION POLITICA			
			POLITICA INTERNACIONAL			
			POLITICA SOCIAL			
			REGION AMAZONICA			



- b) Sostenibilidad:** El Presupuesto de cada entidad debe sujetarse a las posibilidades reales de su financiamiento.
- c) Universalidad:** El presupuesto de cada entidad y órgano público debe contener todos los recursos y gastos que se estimen disponer y realizar para cada gestión fiscal.
- d) Transparencia:** El presupuesto debe ser expresado en términos claros, y difundido tanto al interior de cada entidad como para conocimiento público.
- e) Flexibilidad:** El presupuesto puede ser objeto de ajustes o modificaciones, debiéndose sujetar los mismos a las disposiciones legales y técnicas establecidas o que se establezcan para este efecto.

2.11.5 EJECUCION PRESUPUESTARIA

La ejecución del presupuesto comprende los procesos administrativos de captación de recursos, de realización de desembolsos o pagos, y de ajustes al presupuesto aprobado, sujetos a las regulaciones contenidas en las normas legales inherentes a la materia¹⁸.

2.11.6 PARTIDAS PRESUPUESTARIAS

Todas las entidades y órganos públicos deberán utilizar obligatoriamente los Clasificadores Presupuestarios emitidos y aprobados por el Ministerio de Hacienda para cada gestión fiscal¹⁹.

2.11.7 PROGRAMACIÓN DE LA EJECUCIÓN PRESUPUESTARIA

LA HCD deberá elaborar el cronograma de ejecución de gastos, a nivel agregado o institucional, y en detalle o por categoría programática, identificando las partidas de gasto, y asignando un valor a cada partida para todas las unidades ejecutoras.

2.11.8 MODIFICACIONES AL PRESUPUESTO

Durante el ejercicio fiscal, podrán efectuarse cambios al presupuesto por las siguientes causales:

¹⁸ Sistema de Presupuesto Normas Básicas, RESOLUCION SUPREMA Nro. 225558, Artículo 25

¹⁹ Sistema de Presupuesto Normas Básicas, RESOLUCION SUPREMA Nro. 22558, Artículo 16

- Modificación en el Programa de Operaciones Anual por la incorporación de nuevos objetivos, por cambios en los objetivos iniciales previstos, o por cambios en las metas de estos objetivos
- Desviaciones respecto a las previsiones presupuestarias iniciales, sin cambios en los objetivos.²⁰

2.11.9 SEGUIMIENTO DE LA EJECUCIÓN PRESUPUESTARIA

Según las Normas Básicas del Sistema de Presupuesto, cada entidad y órgano público deberá elaborar y emitir en forma periódica, reportes de ejecución financiera del presupuesto. Estos reportes deben ser preparados a nivel agregado o institucional y por categoría programática.



²⁰ Sistema de Presupuesto Normas Básicas, RESOLUCION SUPREMA Nro. 22558, Artículo 28

CAPITULO 3

MARCO APLICATIVO

3.1 DESARROLLO DEL SISTEMA WEB EMPLEANDO LA METODOLOGÍA XP Y EL MODELADO WEBML

Los desarrollos de software ágil, que adoptan los principios del “Agile Manifesto”, no son anti-metodológicos. Por el contrario, siguen su metodología, diferente a la de los métodos clásicos de desarrollo.

Se trata de lograr un equilibrio, en el que, por ejemplo, la documentación es concreta y útil, y no burocrática y los planes existen, pero reconociendo sus limitaciones en el actual mundo en permanente cambio. [INTECO, 2009].

Para la documentación del presente proyecto se combinará con el lenguaje de modelado WebML

Como ya se ha visto WebML es un lenguaje de modelado gráfico utilizado para apoyar las actividades del diseño de sitios Web. Provee gráficos, formalismos, especificaciones y diseño de procesos.

Define varios tipos de diagramas: de estructura, composición y navegación.

En este caso, WebML dará soporte a la documentación del presente proyecto, en función a las fases de XP, ya que las historias de usuario no proporcionan especificaciones funcionales de manera estandarizada.

En la siguiente tabla se muestra el uso de WebML, dentro de la metodología de desarrollo ágil XP, siguiendo los principios del “Agile Manifesto”:

METODOLOGÍA XP	PROCESOS XP	PROCESOS WEBML
Fase de Exploración	<ul style="list-style-type: none"> Historias de Usuario 	<ul style="list-style-type: none"> Clasificación e Identificación de Roles
Fase de Planificación	<ul style="list-style-type: none"> Tareas Plan de Entregas (Release plan) 	
Fase de Iteraciones		<ul style="list-style-type: none"> Diagrama de Estructura Diagramas de Composición Diagramas de Navegación
Fase de Puesta en Producción	<ul style="list-style-type: none"> Casos de Prueba 	<ul style="list-style-type: none"> Modelo de Presentación

Tabla 3.1: Desarrollo Web empleando la metodología ágil XP y el lenguaje de modelado WebML

Fuente: Elaboración Propia

3.2 FASE DE EXPLORACIÓN

3.2.1 HISTORIAS DE USUARIO

A continuación se muestran las historias de usuario recolectadas de la Unidad de Presupuestos, las historias fueron recabadas en distintas reuniones llevadas a cabo con el personal de dicha unidad, según el cronograma establecido.

La siguiente historia de usuario, describe el proceso de registro del presupuesto anual.

Historia de Usuario	
Número: 1	Usuario: Operativo
Nombre historia: Registro de Presupuesto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
<p>Descripción: Permitir la alimentación del Presupuesto de la Gestión vigente, en forma desagregada y por Unidad Organizacional, se debe registrar la relación: partida presupuestaria, unidad y monto en bolivianos.</p> <p>El presupuesto registrado deberá ser accedido mediante la búsqueda de partida y unidad, una vez accedido se permitirá la eliminación del mismo.</p>	

Tabla 3.2: Historia de usuario: Registro de presupuesto

Fuente: Elaboración propia

La siguiente historia de usuario, describe el proceso de registro de la ejecución presupuestaria.

Historia de Usuario	
Número: 2	Usuario: Operativo
Nombre historia: Registro de Gastos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
<p>Descripción: Permitir el registro de gastos, mediante la inserción de los datos consignados en el formulario de gasto C-31 (que se alimenta al sistema SIGMA), incluyendo la Unidad, la fecha, número de documento y detalle de las partidas de gasto.</p>	

Tabla 3.3: Historia de usuario: Registro de gastos

Fuente: Elaboración Propia

La siguiente historia de usuario describe requerimientos asociados a la seguridad del sistema, también la necesidad de un registro de acciones y acceso.

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Control y seguimiento de usuarios.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
<p>Descripción: Permitir el acceso al sistema solamente a usuarios autorizados, al mismo tiempo, registrar las acciones del usuario operativo y administrador, con el fin de auditar las operaciones llevadas a cabo por el mismo.</p> <p>Restringir el acceso del usuario visitante, es decir, solo podrá acceder a la información presupuestaria de la unidad a la que corresponde.</p>	

Tabla 3.4: Historia de usuario: Control y seguimiento de usuarios

Fuente: Elaboración propia

La siguiente historia de usuario describe la parametrización de partidas de gasto.

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Registro de Partidas	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
<p>Descripción: Las partidas presupuestarias están conformadas por valores paramétricos, que deben ser reguladas por el administrador o el usuario operativo, los mismos podrán añadir nuevas partidas.</p>	

Tabla 3.5: Historia de usuario: Administración de parámetros

Fuente: Elaboración propia

La siguiente historia de usuario describe la capacidad que deberá tener el sistema de realizar modificaciones al presupuesto, y el tipo de modificaciones existentes.

Historia de Usuario	
Número: 5	Usuario: Operativo
Nombre historia: Modificación presupuestaria.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 3
Descripción: Permitir el registro de modificaciones al presupuesto entre distintas partidas y distintas unidades, como también entre mismas partidas y distintas unidades,	

Tabla 3.6: Historia de Usuario: Modificación Presupuestaria

Fuente: Elaboración Propia

La siguiente historia de usuario describe las salidas del sistema, traducidas en reportes, se describen tipos y una breve descripción de la información a generar.

Historia de Usuario	
Número: 6	Usuario: Operativo
Nombre historia: Reportes.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 2
Descripción: Permitir la emisión de reportes de ejecución presupuestaria, también reportes de libro mayor por partida, dichos reportes mostraran el presupuesto inicial, la modificación presupuestaria, el presupuesto ejecutado, saldo y porcentajes de ejecución.	

Tabla 3.7: Historia de Usuario: Modificación Presupuestaria

Fuente: Elaboración Propia

3.2.2 CLASIFICACIÓN E IDENTIFICACIÓN DE ROLES (ACTORES)

Primero se debe identificar a todos los actores que van a interactuar con la aplicación, posteriormente se los clasifica en clases y subclases de actores [Soto & Palma, 2004]. Esta clasificación permite organizar la información evitando la confusión en casos de que se tengan actores que compartan tareas (ver figura 3.1)

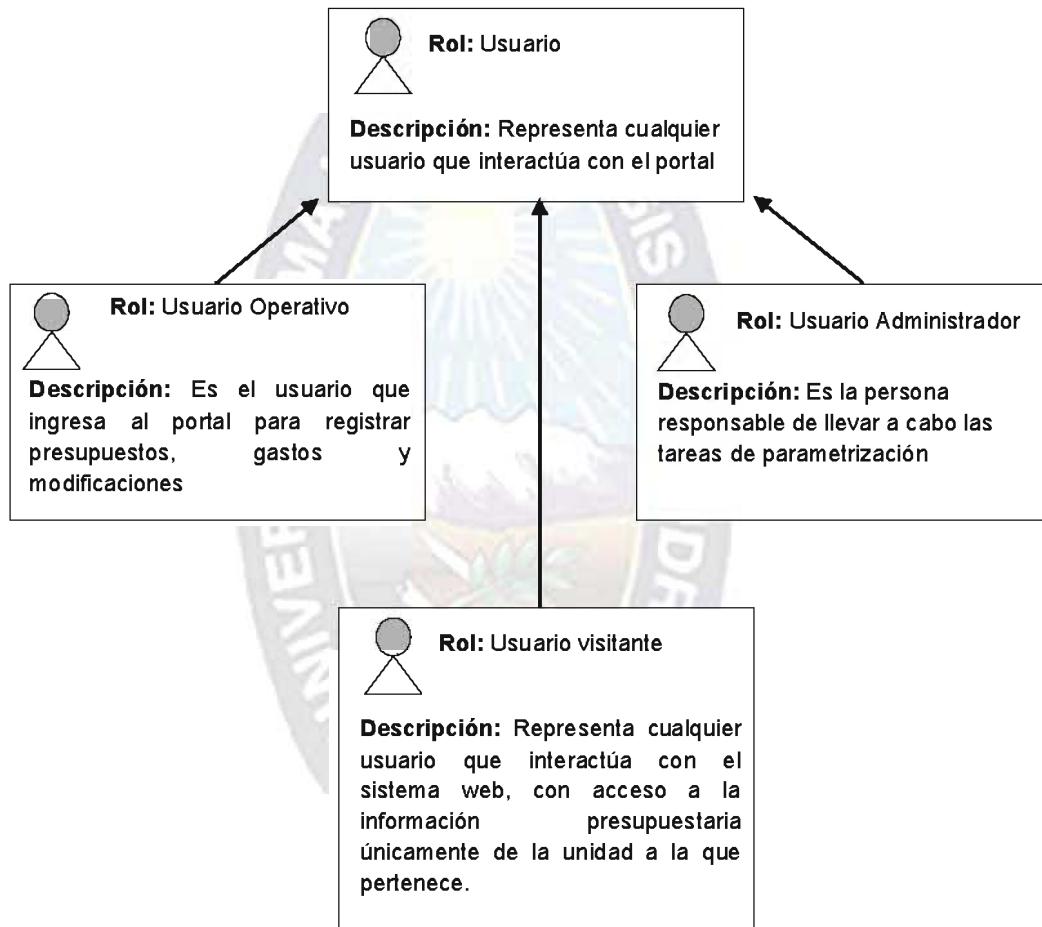


Figura 3.1: Clasificación e Identificación de roles

Fuente: Elaboración Propia

3.3 FASE DE PLANIFICACIÓN

Una vez recopiladas las historias de usuario, se asignan a estas distintas tareas, para luego consensuar con todos involucrados del proyecto, el plan de entregas.

3.3.1 IDENTIFICACION DE TAREAS (Task Cards)

A continuación se muestran las tareas asociadas a cada historia de usuario, y las características del mismo.

3.3.1.1 REGISTRO DE PRESUPUESTO

La siguiente tarea describe la funcionalidad del Registro de Presupuesto.

Tarea de Programación	
Número: 1	Tipo de tarea: Mejora
Nombre de Tarea: Interfaz de registro de presupuesto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Descripción: Desarrollar una interfaz que despliegue dos campos: "Unidad" y "Grupo de Gasto", al ser ambos seleccionados desplegar otra interfaz que contenga un formulario para asignar presupuesto a la unidad seleccionada, con las partidas correspondientes al grupo de gasto.	

Tabla 3.8: Tarea de Programación: Interfaz de registro de presupuesto

Fuente: Elaboración Propia

La siguiente tarea describe la funcionalidad para búsqueda de registros presupuestarios.

Tarea de Programación	
Número: 2	Tipo: Mejora
Nombre de Tarea: Interfaz de búsqueda de presupuesto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Descripción: Desarrollar una interfaz que permita la búsqueda de registros presupuestarios mediante una determinada unidad, desplegar en los resultados: Partida, Presupuesto, Modificación Presupuestaria, Presupuesto Ajustado, y permitir desde esta interfaz la opción eliminar.	

Tabla 3.9: Tarea de Programación: Interfaz de búsqueda de presupuesto

Fuente: Elaboración Propia

3.3.1.2 REGISTRO DE GASTOS

La siguiente tarea describe la funcionalidad y condiciones necesarias para el Registro de Gastos.

Tarea de Programación	
Número: 3	Tipo: Mejora
Nombre de Tarea: Interfaz de registro de gastos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: Desarrollar una interfaz que permita el registro de gastos, desplegando únicamente las unidades que poseen presupuesto, de esta manera se debe registrar, la glosa y monto. Además se registrará el código del formulario C-31, la fecha y el tipo de gasto.	

Tabla 3.10: Tarea de Programación: Interfaz de registro de gastos

Fuente: Elaboración Propia

La siguiente tarea describe la funcionalidad para búsqueda de gastos presupuestarios.

Tarea de Programación	
Número: 4	Tipo: Mejora
Nombre de Tarea: Interfaz de búsqueda de gastos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: Desarrollar una interfaz que permita la búsqueda de gastos, por periodos y por unidad, desplegando en el resultado: fecha de gasto, código de formulario C-31, partida, unidad, glosa y monto. Además se debe permitir la edición de un determinado gasto o conjunto de gastos.	

Tabla 3.11: Tarea de Programación: Interfaz de búsqueda de gastos

Fuente: Elaboración Propia

3.3.1.3 CONTROL Y SEGUIMIENTO DE USUARIOS

La siguiente tarea describe el proceso de autenticación de usuarios, con el fin de restringir el acceso únicamente a usuarios autorizados.

Tarea de Programación	
Número: 5	Tipo: Mejora
Nombre de Tarea: Interfaz de ingreso al sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Descripción: Desarrollar una interfaz que permita la autenticación de usuarios autorizados, a su vez autenticar la dirección física del equipo al que pertenece el usuario.	

Tabla 3.12: Tarea de Programación: Interfaz de ingreso al sistema

Fuente: Elaboración Propia

La siguiente tarea describe la funcionalidad para el control y seguimiento de usuarios, con fines de auditoría.

Tarea de Programación	
Número: 6	Tipo: Mejora
Nombre de Tarea: Log de ingreso y acciones	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: Registrar, de manera transparente al usuario, hora, fecha, nombre de equipo, IP de equipo, del cual se conecta al sistema.	

Tabla 3.13: Tarea de Programación: Log de ingreso

Fuente: Elaboración Propia

La siguiente tarea describe la capacidad que debe tener el sistema para altas, bajas y modificaciones de usuarios.

Tarea de Programación	
Número: 7	Tipo: Mejora
Nombre de Tarea: Interfaz de Administración de usuarios.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Descripción: Permitir la inserción de nuevos usuarios, registrando el nombre, contraseña, nivel de usuario y unidad a la que pertenece. Permitir la búsqueda de usuarios por cualquiera de sus características, para su posterior eliminación o edición.	

Tabla 3.14: Tarea de Programación: Log de acciones

Fuente: Elaboración Propia

3.3.1.4 REGISTRO DE PARTIDAS

La siguiente tarea describe la capacidad que debe tener el sistema para altas, bajas y modificaciones de partidas de gasto.

Tarea de Programación	
Número: 8	Tipo: Mejora
Nombre de Tarea: Interfaz de registro de partidas	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Descripción: Mediante un formulario permitir el registro de nuevas partidas presupuestarias, con los atributos: Código, Descripción y Grupo de Gasto al que pertenece.	

Tabla 3.15: Tarea de Programación: Interfaz de registro de partidas

Fuente: Elaboración Propia

3.3.1.5 MODIFICACIÓN PRESUPUESTARIA

Las siguientes tareas describen la funcionalidad para el proceso de modificación presupuestaria entre distintas unidades organizacionales y distintas partidas.

Tarea de Programación	
Número: 9	Tipo: Mejora
Nombre de Tarea: Interfaz Disminución Modificación SIGMA	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 3
Descripción: Mediante un formulario permitir la selección de una partida de gasto para luego introducir el monto a ser disminuido, el sistema desplegará las unidades que tengan presupuesto para esa partida. El usuario seleccionara distintas unidades organizacionales con sus respectivos montos a ser disminuidos, la suma de estos deberá coincidir con el total ingresado a ser disminuido,	

Tabla 3.16: Tarea de Programación: Interfaz disminución SIGMA

Fuente: Elaboración Propia

Tarea de Programación	
Número: 10	Tipo: Mejora
Nombre de Tarea: Interfaz Incremento Modificación SIGMA	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 3
Descripción: Mediante un formulario permitir la selección de una partida de gasto para luego introducir el monto a ser incrementado. El sistema permitirá la selección múltiple de todas las unidades organizacionales. El usuario seleccionara distintas unidades organizacionales con sus respectivos montos a ser disminuidos, la suma de estos deberá coincidir con el total ingresado a ser disminuido,	

Tabla 3.17: Tarea de Programación: Interfaz incremento SIGMA

Fuente: Elaboración Propia

3.3.1.6 REPORTES

Las siguientes tareas describen la funcionalidad para las salidas del sistema, tipo reporte.

Tarea de Programación	
Número: 11	Tipo: Mejora
Nombre de Tarea: Interfaz reporte ejecución presupuestaria	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 3
Descripción: Desplegar una tabla que contenga las siguientes columnas: partida, presupuesto (en Bs.), modificación (en Bs), presupuesto ajustado (en Bs.), presupuesto ejecutado (en Bs.) y porcentaje de ejecución. El usuario seleccionara podrá filtrar la tabla por fecha, componente, área o unidad organizacional.	

Tabla 3.18: Tarea de Programación: Interfaz reporte ejecución presupuestaria

Fuente: Elaboración Propia

Tarea de Programación	
Número: 12	Tipo: Mejora
Nombre de Tarea: Interfaz reporte libro mayor por partida	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 3
Descripción: Desplegar una tabla que contenga las siguientes columnas: unidad organizacional, fecha, glosa, código formulario C-31, monto ejecutado (en Bs.) Para visualizar este reporte se deberá seleccionar una partida, luego se podrá filtrar la tabla por fecha, componente, área o unidad organizacional.	

Tabla 3.19: Tarea de Programación: Interfaz reporte ejecución presupuestaria

Fuente: Elaboración Propia

3.3.2 PLAN DE ENTREGAS (RELEASE PLAN)

En esta fase se desarrollan las funcionalidades del sistema a implementar, en función a las tareas especificadas en la Fase de Planificación.

ID	ACTIVIDAD	INICIO	FINAL	DURACIÓN	Junio 2011				Julio 2011				Agosto 2011				Septiembre 2011			
					S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14		
1	Módulo de Presupuesto	15/06/2010	29/06/2010	2s	■	■														
2	Módulo de Gastos	1/07/2010	9/07/2010	2s			■	■												
3	Módulo de Modificación al presupuesto	12/07/2010	16/07/2010	3s					■	■	■									
4	Módulo de Reportes	19/07/2010	23/07/2010	3s								■	■	■						
5	Administración de Partidas	26/07/2010	30/07/2010	1s										■						
6	Administración de usuarios	2/08/2010	20/08/2010	1s											■					
7	Control y Seguimiento de Usuarios	9/08/2010	27/08/2010	2s												■	■			

Tabla 3.20: Plan de Entregas

Fuente: Elaboración Propia

3.4 FASE DE ITERACIONES

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido en el plan de entregas.

Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación.

Como se ha mencionado anteriormente, las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, se especificaran diagramas convencionales basados en la notación WebML.

Al final de cada iteración se obtendrá un entregable funcional, para ello se llevarán a cabo las tareas en el orden establecido.

Las iteraciones nos han de servir para medir el avance del proyecto, una iteración terminada sin errores es una medida clara de avance.

Por consiguiente, la implementación exitosa de un módulo, se traducirá como el cumplimiento de un objetivo específico, y la totalidad de los objetivos específicos se traducirá en el éxito total del proyecto.

3.4.1 DIAGRAMA DE ESTRUCTURA

A continuación se definen las entidades o contenedores de datos y sus relaciones, este diagrama expresa el contenido de un sitio Web en términos de entidades y relaciones.

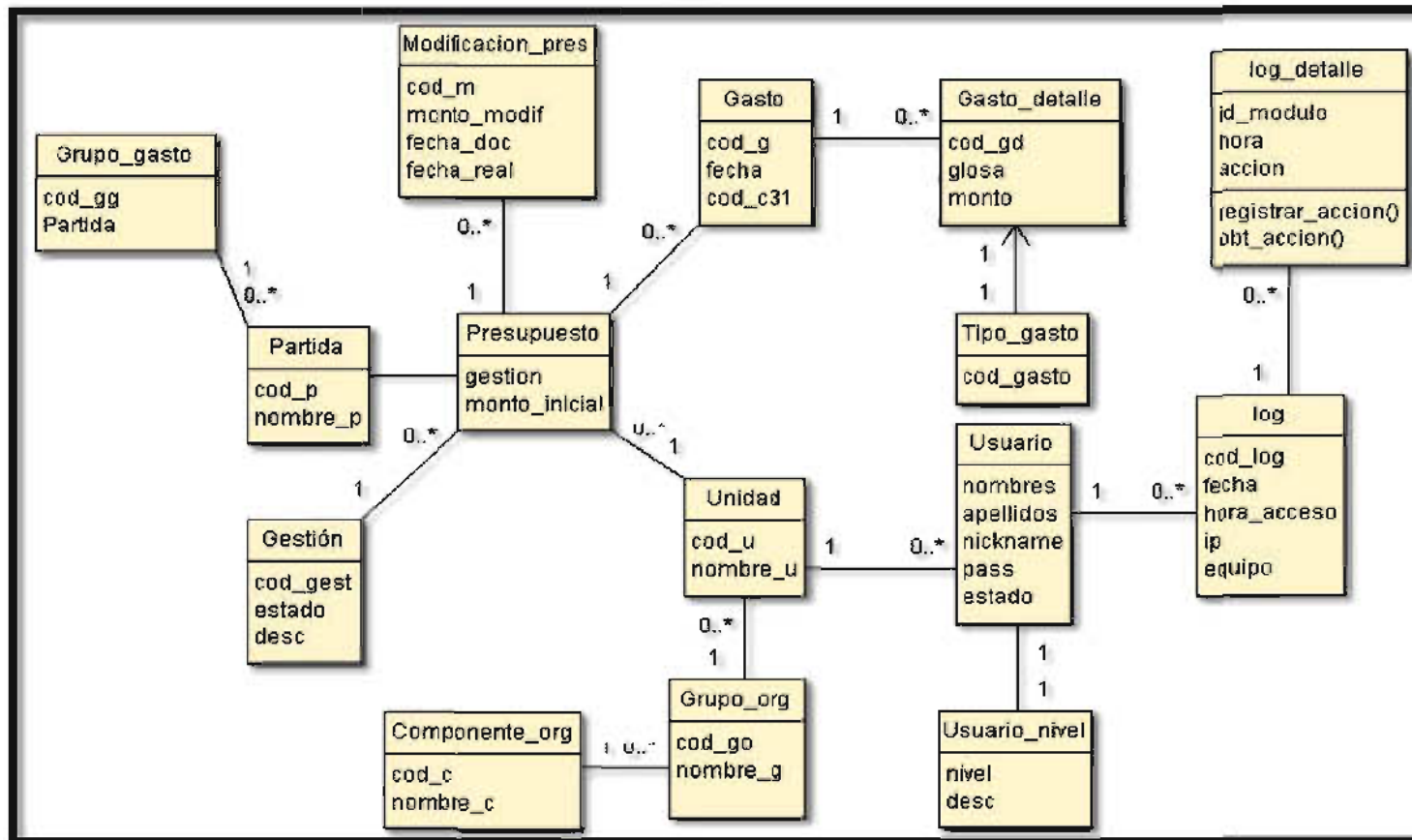


Figura 3.2: Diseño Conceptual – Sistema Web Control y Seguimiento de la Ejecución Presupuestaria

Fuente: Elaboración propia

3.4.2 DIAGRAMAS DE COMPOSICIÓN

A continuación se definen los nodos que forman parte del hipertexto contenido en el sitio Web, es decir, se especifican las páginas y las unidades (elementos atómicos de información) que componen el sitio Web.

3.4.2.1 PRESUPUESTO

La figura 3.3 describe la lógica de la extracción de datos y cómo se utilizan estos para componer la interfaz de usuario, dos interfaces para este caso.

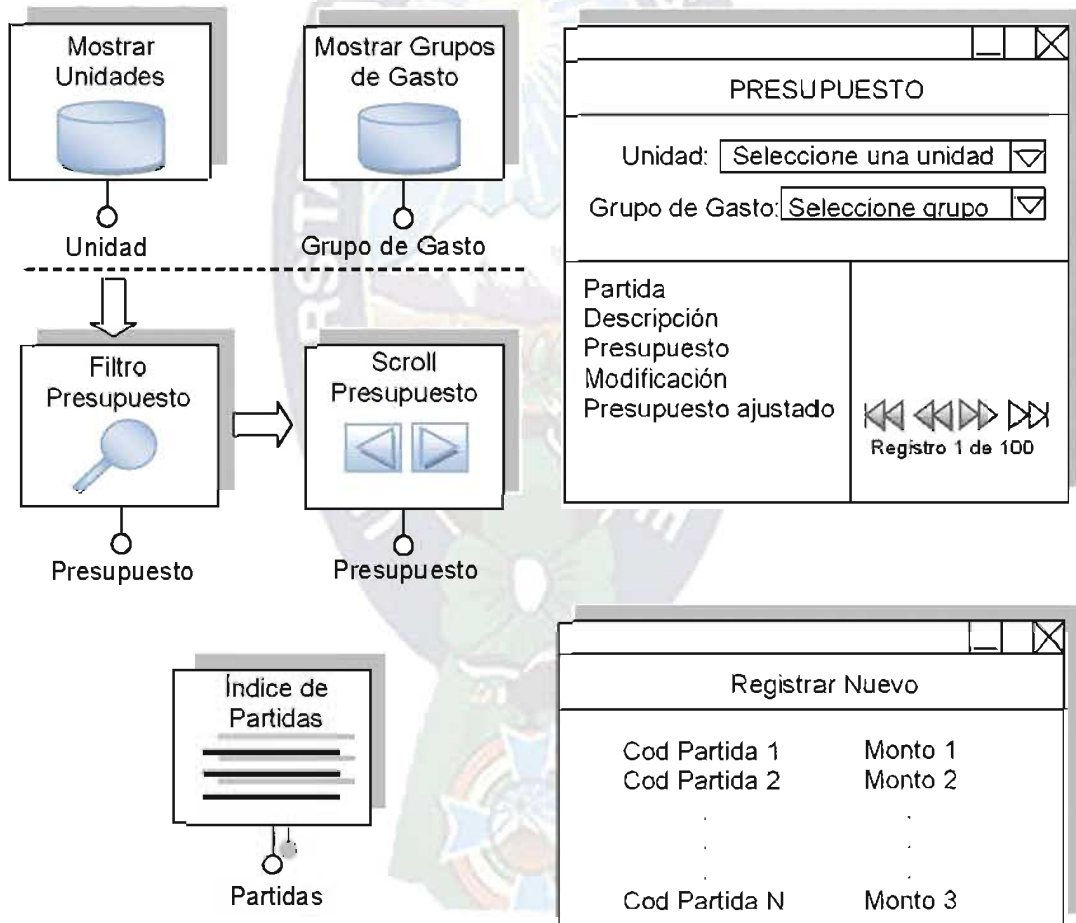


Figura 3.3: Diagrama de composición – Presupuesto

Fuente: Elaboración Propia

3.4.2.2 GASTOS

La figura 3.4 describe las interfaces y su relación con los datos, para el módulo de gastos.

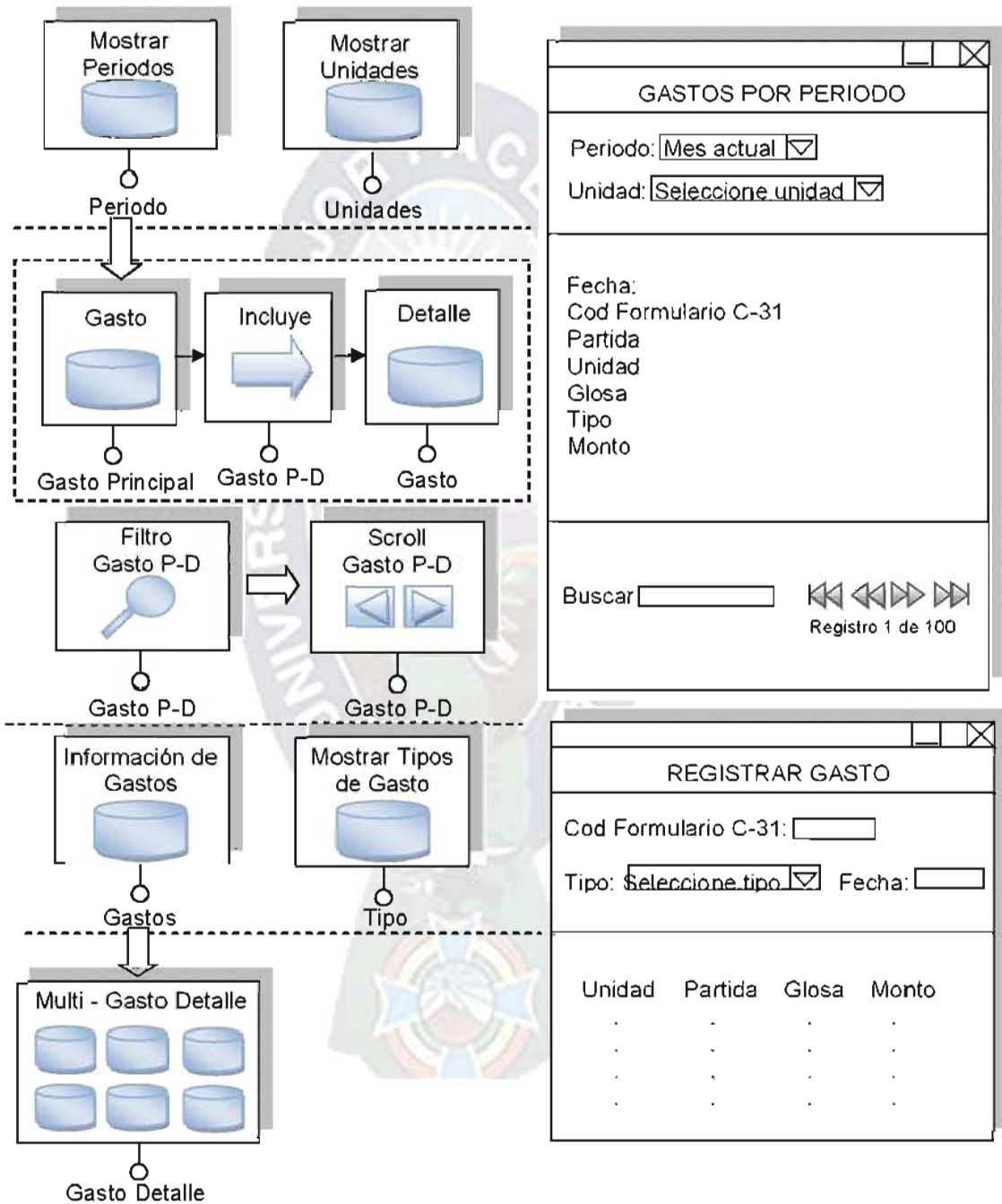


Figura 3.4: Diagrama de composición – Gastos

Fuente: Elaboración Propia

3.4.2.3 MODIFICACIÓN PRESUPUESTARIA

La figura 3.5 describe la relación interna con la interfaz de usuario para el módulo de Modificación al presupuesto en términos de unidades elementales de contenido.

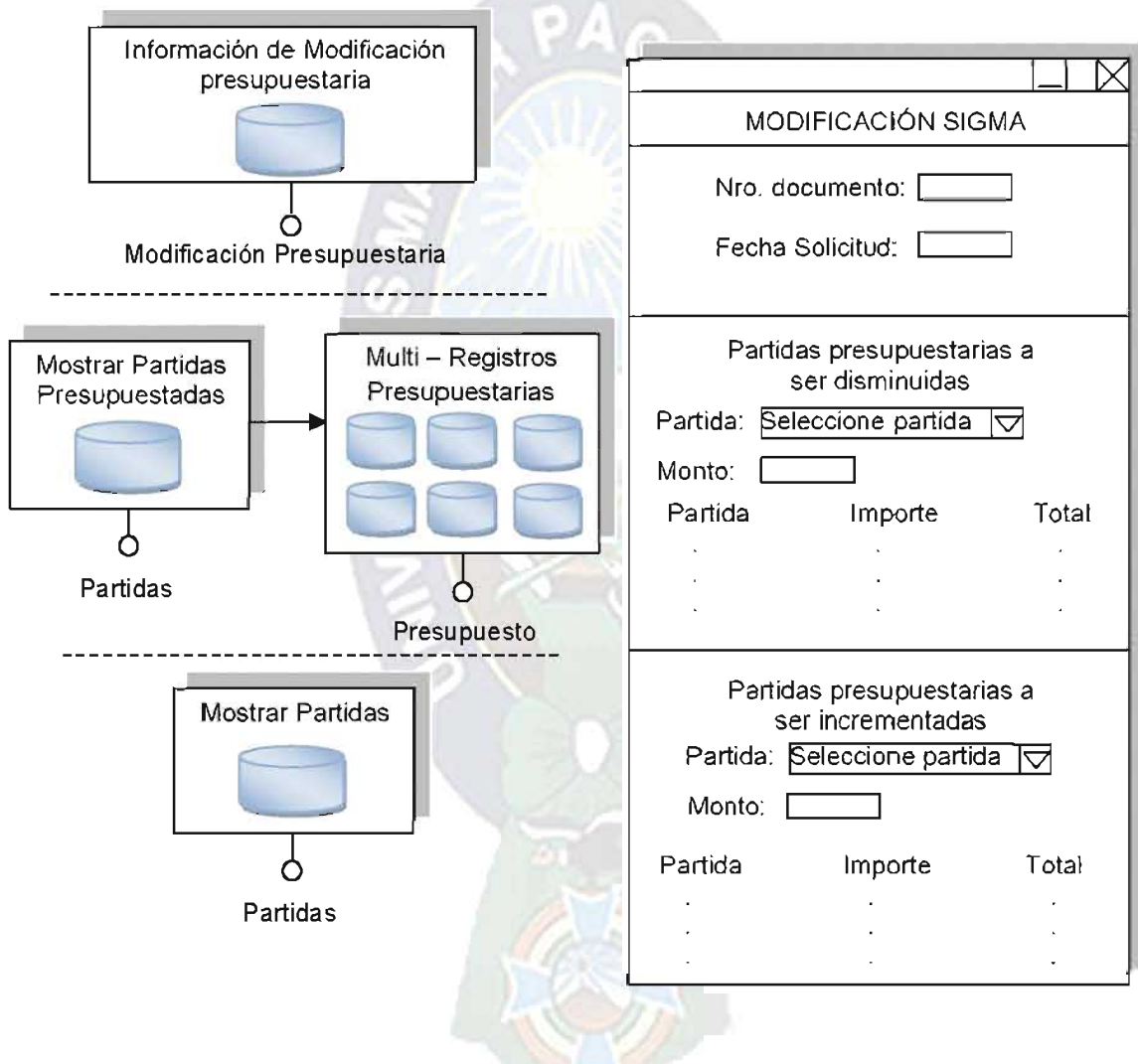


Figura 3.5: Diagrama de composición – Modificación Presupuestaria

Fuente: Elaboración Propia

3.4.2.4 REPORTE EJECUCIÓN PRESUPUESTARIA

La figura 3.6 describe la relación interna de una salida del sistema tipo reporte, es decir, su relación con las unidades de datos, y la información que contiene.

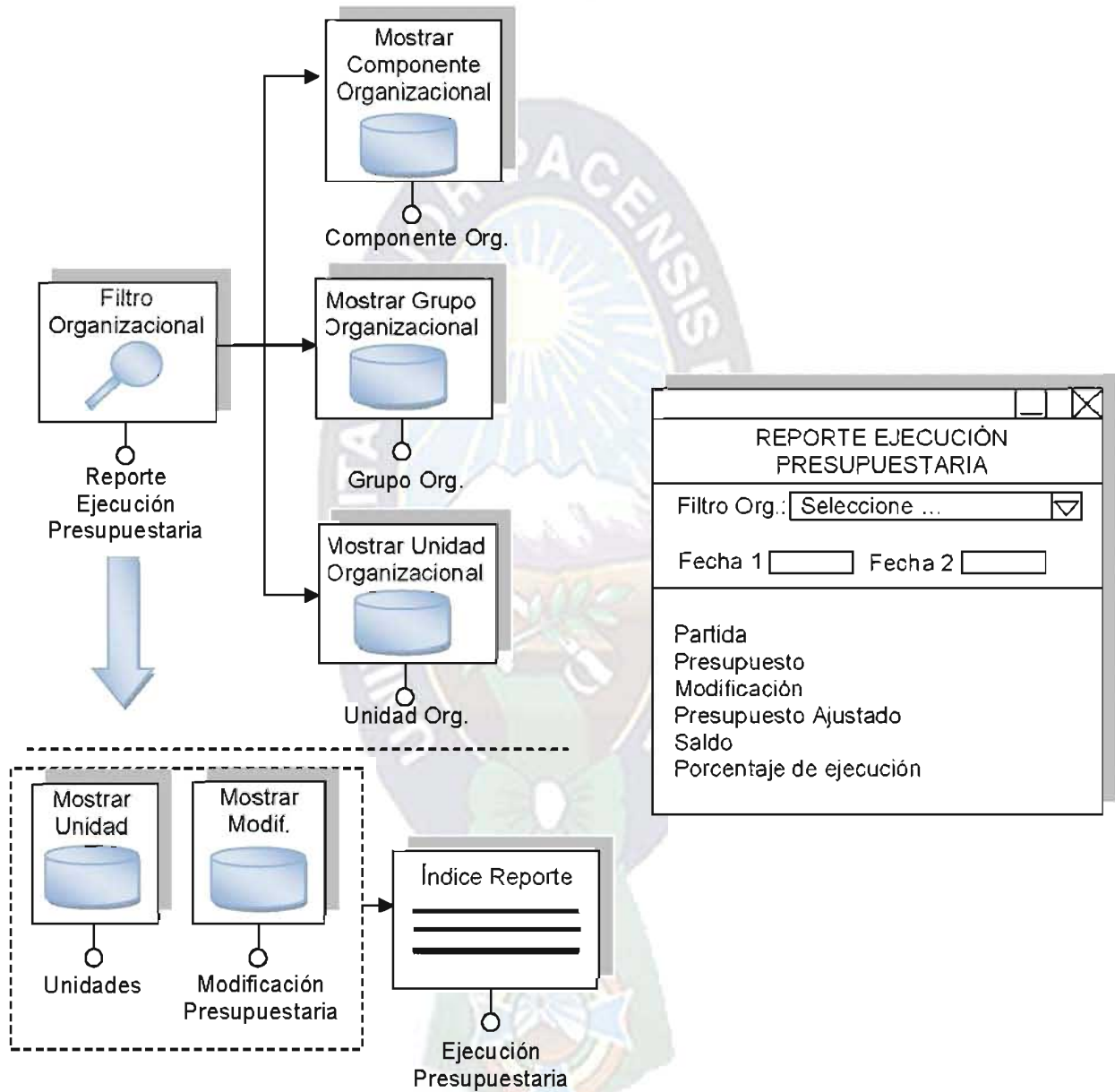


Figura 3.6: Diagrama de composición – Modificación Presupuestaria

Fuente: Elaboración Propia

3.4.2.5 REPORTE LIBRO MAYOR

La figura 3.7 describe la relación interna de una salida del sistema tipo reporte, es decir, su relación con las unidades de datos, y la información que contiene.

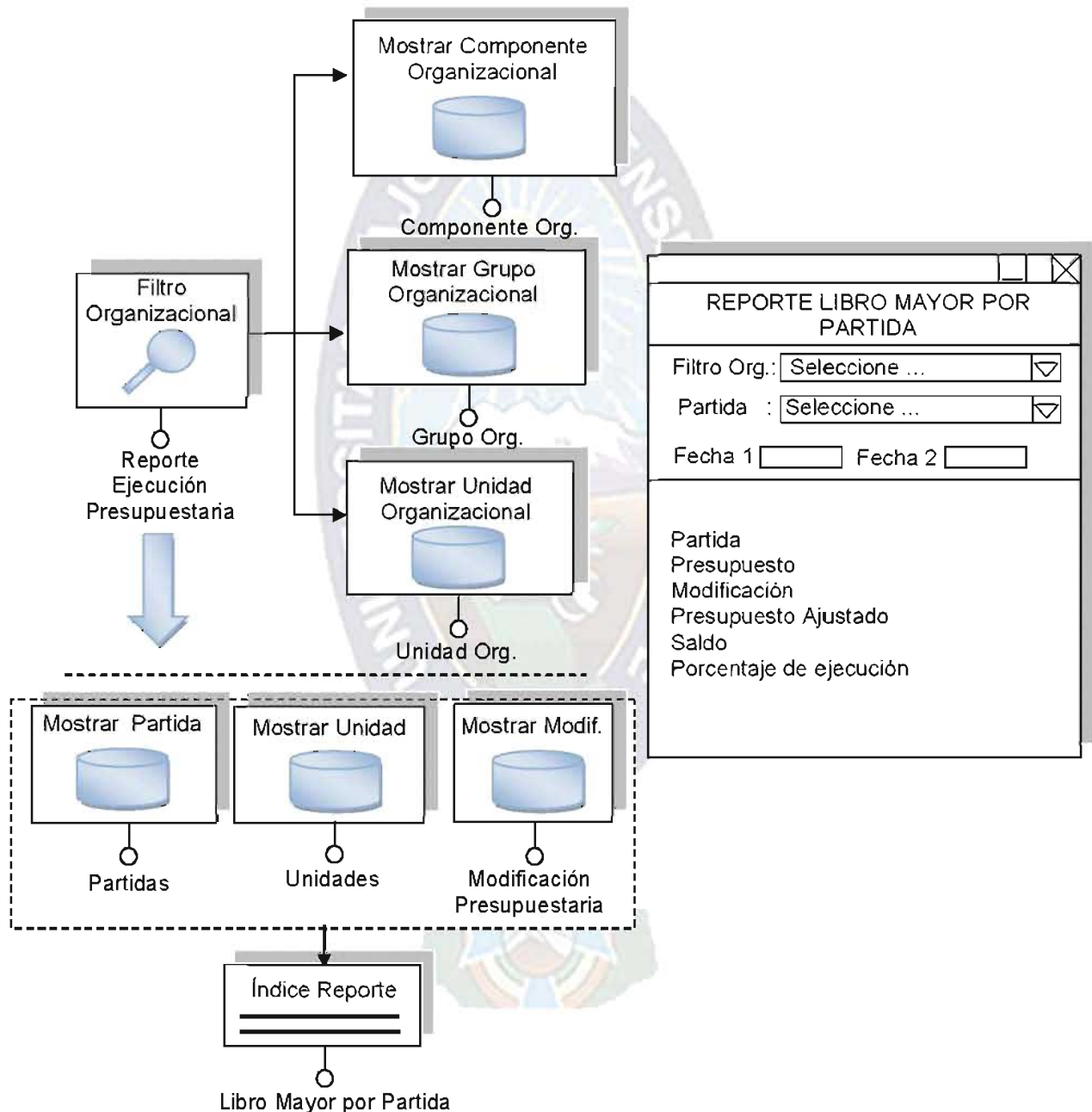


Figura 3.7: Diagrama de composición – Modificación Presupuestaria

Fuente: Elaboración Propia

3.4.3 DIAGRAMAS DE NAVEGACIÓN

A continuación se describe el contexto en el cual se desenvolverá el usuario es decir la información a la que tendrá acceso y por donde podrá navegar.

3.4.3.1 PRESUPUESTO

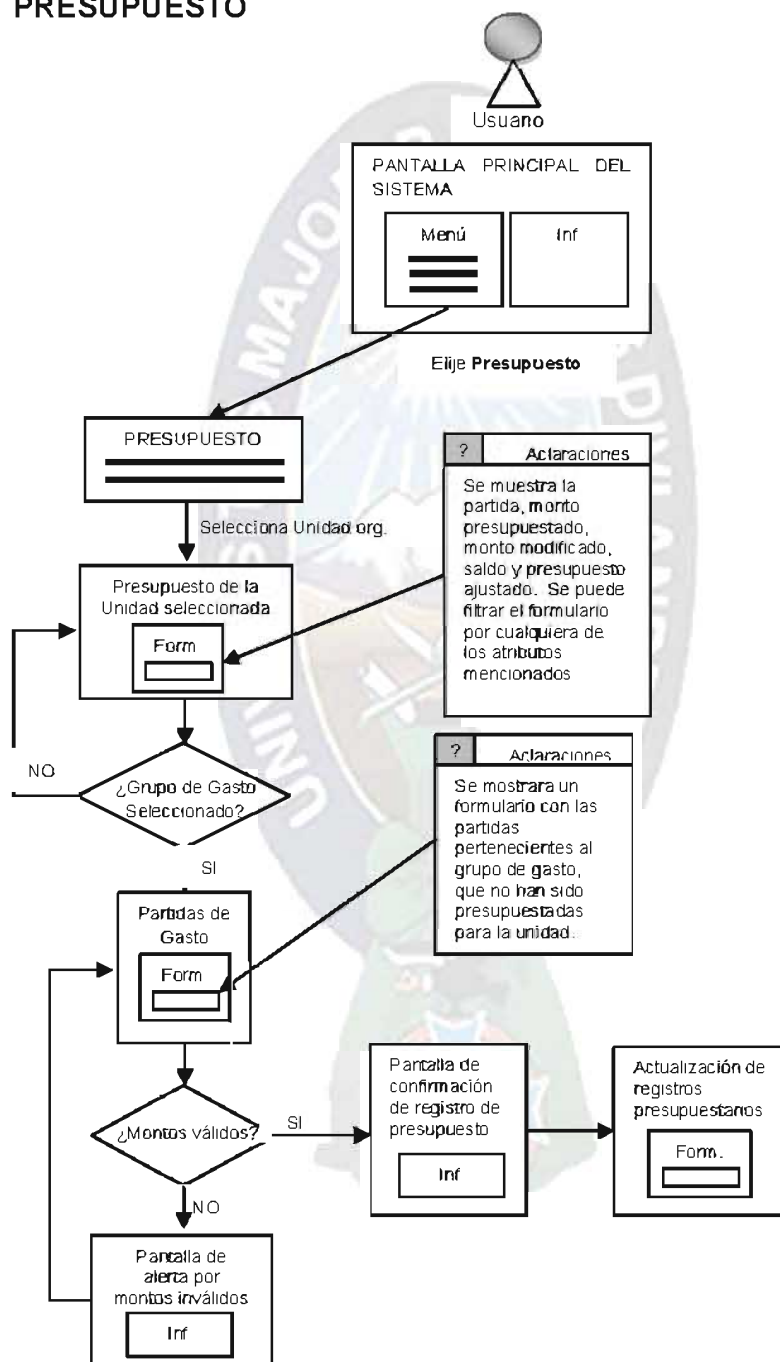


Figura 3.8: Diagrama de navegación – Presupuesto

Fuente: Elaboración Propia

3.4.3.2 GASTOS

La figura 3.9 muestra el comportamiento navegacional del módulo de Gastos, desde el ingreso al sistema hasta la finalización del proceso.

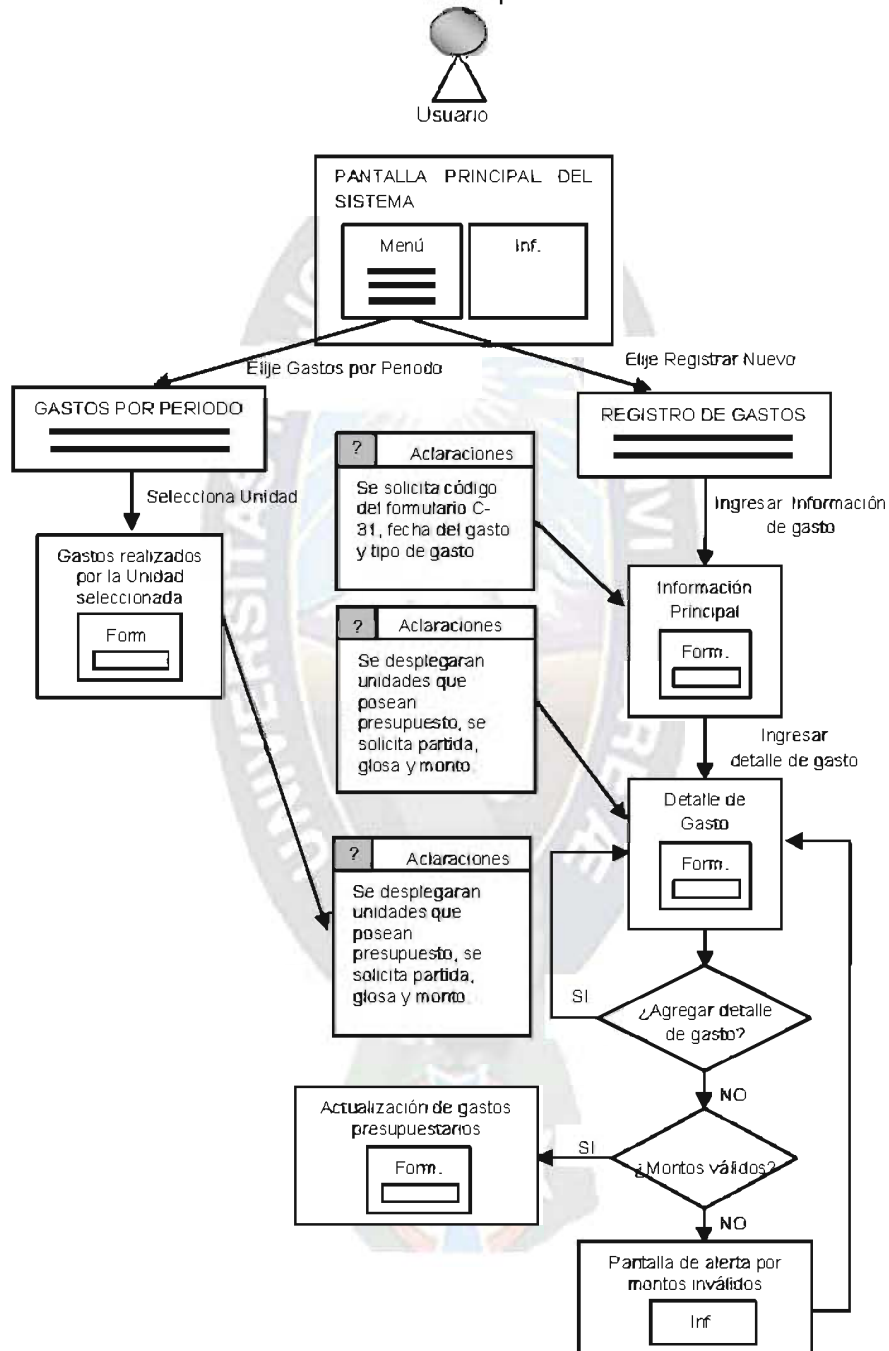


Figura 3.9: Diagrama de navegación – Gastos

Fuente: Elaboración Propia

3.4.3.3 MODIFICACIÓN PRESUPUESTARIA

La figura 3.10 muestra el comportamiento navegacional del módulo de Modificación al presupuesto, desde el ingreso al sistema hasta la finalización del proceso.

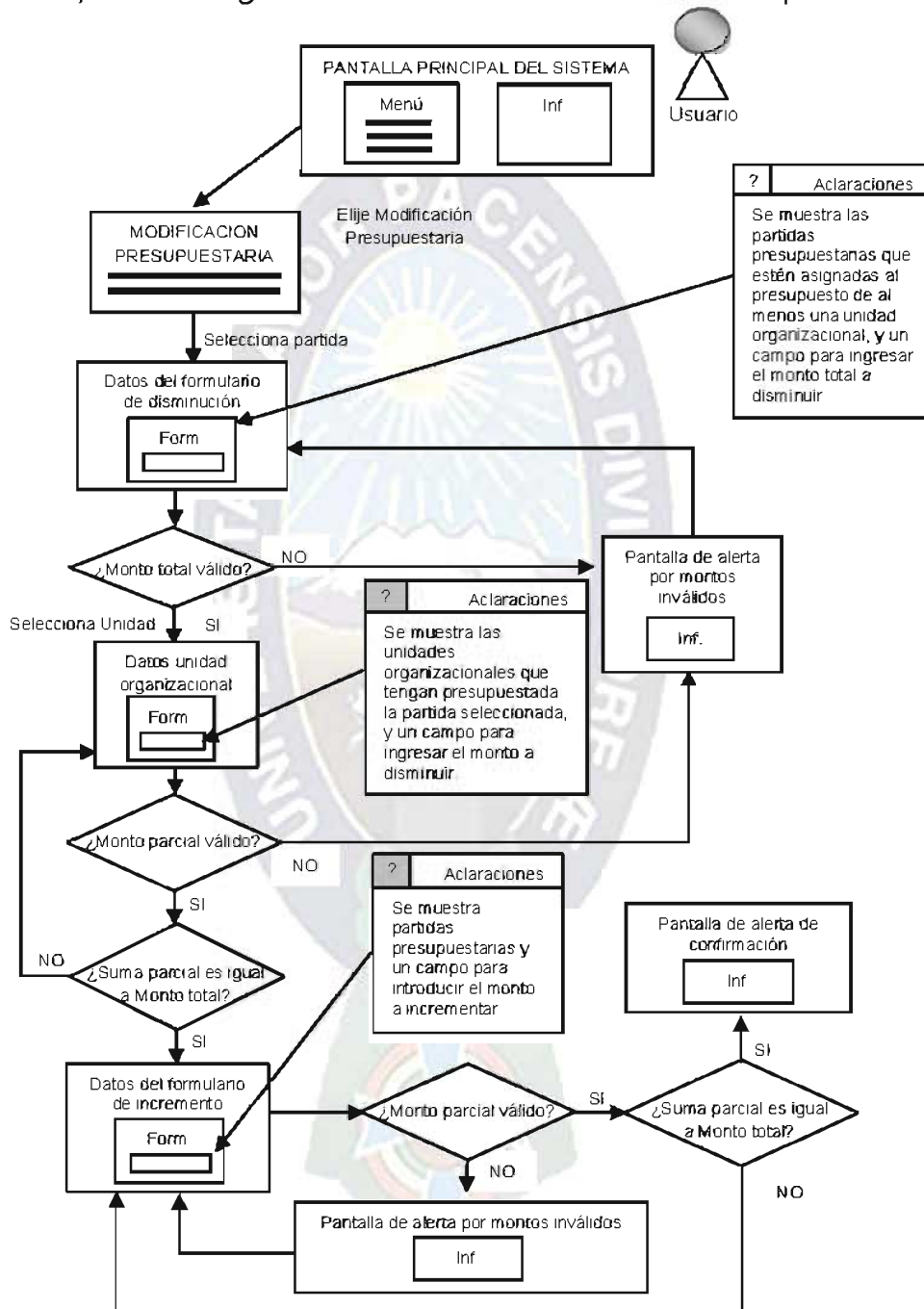


Figura 3.10: Diagrama de navegación – Modificación Presupuestaria

Fuente: Elaboración Propia

3.4.3.4 REPORTES

La figura 3.11 muestra el comportamiento navegacional del módulo de Reportes, desde el ingreso al sistema hasta la el proceso de salida, tipo reporte.

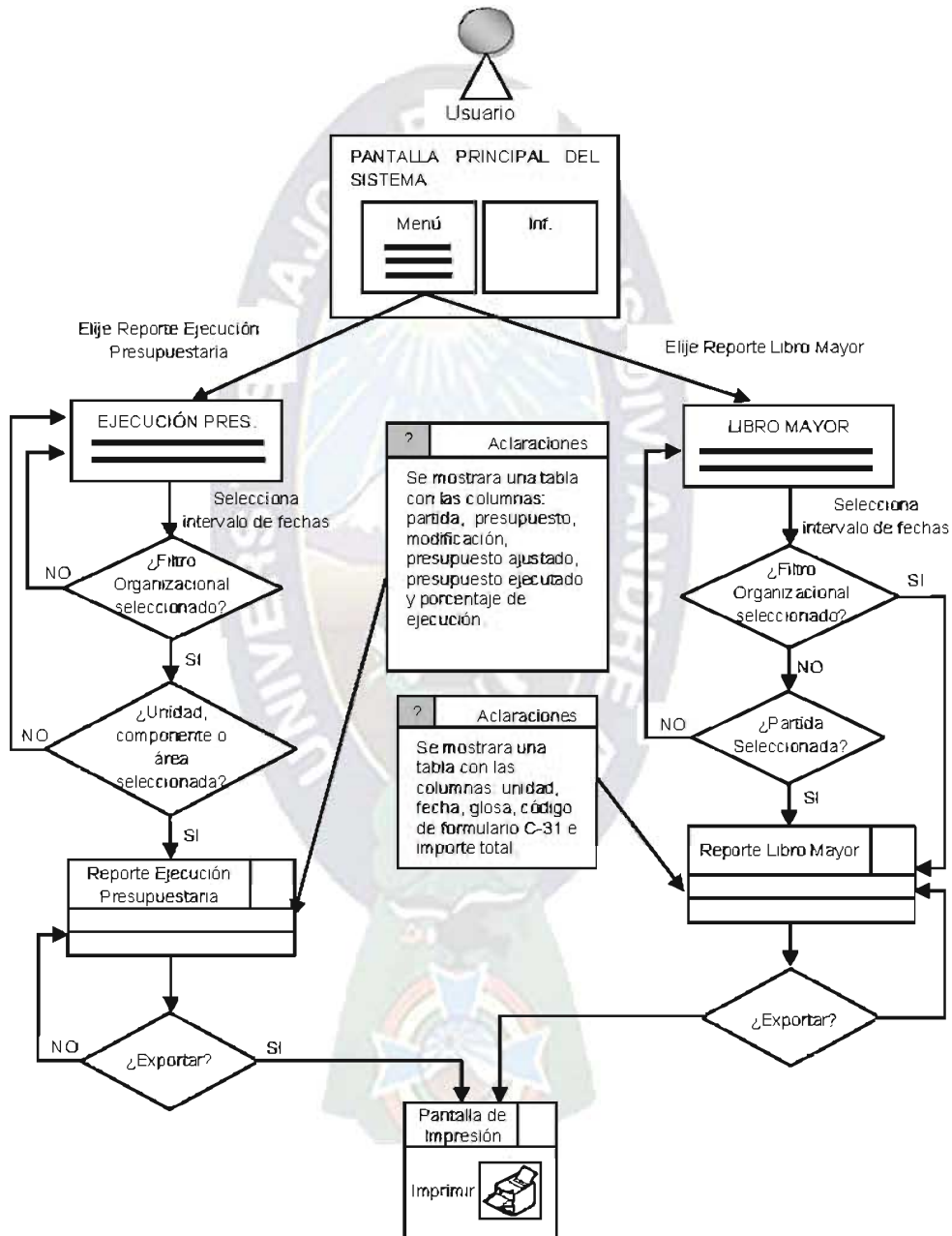


Figura 3.11: Diagrama de navegación – Reportes

Fuente: Elaboración Propia

3.5 FASE DE PUESTA EN PRODUCCIÓN

En esta fase se presentan pantallas principales, la tecnología empleada para su implementación, y las pruebas de aceptación correspondientes a cada historia de usuario.

3.5.1 HERRAMIENTAS DE DESARROLLO WEB

- **The Color Picker.** Como un apoyo en la elaboración de hojas de estilo, permitiendo capturar cualquier color en formato hexadecimal.
- **Komodo Active Edit.** Para el desarrollo del código PHP del lado del servidor.
- **Mozilla Firefox + Plugin Firebug.** Para la depuración de errores de las aplicaciones Javascript y Ajax.
- **ArgoUML** Para el modelado estructural haciendo uso del diagrama de clases.
- **MySQL.** Sistema de gestión de bases de datos relacional, multihilo y multiusuario.

3.5.2 LENGUAJES DE PROGRAMACIÓN

- **PHP 5.** Lenguaje usado para la implementación de las aplicaciones del lado del servidor.
- **HTML.** Código usado para la elaboración de los documentos Web.
- **Java script.** Lenguaje empleado para la elaboración de los scripts del lado del cliente.
- **AJAX.** Conjunto de tecnologías utilizada para interactuar con el servidor a través de peticiones.
- **Hojas de estilo CSS.** Código empleado para el diseño del entorno del Sistema Web.
- **MySQL Workbench.** Herramienta complementaria para el gestor de Base de Datos MySQL, proporciona entorno gráfico y asistencia para el diseño del modelo lógico de la base de datos.



SISEP - Mozilla Firefox

SISEP

http://localhost/seg/CD/mod_pres.php

Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario:daf

REGISTRO DE PRESUPUESTO ANUAL POR UNIDAD

UNIDAD: DIR INFORMATICA
GRUPO: Seleccione Grupo de Gasto

Mostrar: 10 registros

PARTIDA	DESCRIPCION	PRESUPUESTO	MODIFICACION	PRES. AJUST.	ENT
0000	SERVICIOS PERSONALES	50,000.00	0.00	50,000.00	
13110	Regenerado Corto Plazo (Salud)	20,000.00	0.00	20,000.00	X
13120	Pena de Riesgo Profesional Regimen de Largo Plazo	30,000.00	0.00	30,000.00	A
20000	SERVICIOS NO PERSONALES	11,000.00	0.00	11,000.00	
21100	Comunicaciones	5,000.00	0.00	5,000.00	A
21200	Energía Eléctrica	6,000.00	0.00	6,000.00	X
30000	MATERIALES Y SUMINISTROS	292,000.00	0.00	292,000.00	
31110	Gastos Destinados al Pago de Retiros al Personal de las Inst. Públicas	150,000.00	0.00	150,000.00	X
31120	Gastos Por Alimentación y Otros Similares etc. En Reuniones, Sem. Y otros Eventos	72,000.00	0.00	72,000.00	A
31200	Productos Agrícolas, Pecuarias y Forestales	90,000.00	0.00	90,000.00	X
PARTIDA	DESCRIPCION	PRESUPUESTO	MODIFICACION	PRES. AJUST.	ENT

Mostrando 1 a 10 de 11 registros

Firefox

http://localhost/sep/v.03/orm_grupo_gasto.php?id_gg=10000&id_u=1

PARTIDA	DESCRIPCION	MONTO
11220	Bono de Antigüedad	20300.00
11400	Aguinaldos	35300.00
11600	Asignaciones Familiares	
11700	Sueldos	
12100	Personal Eventual	
13110	Regimen de Corto Plazo (Salud)	
13120	Prima de Riesgo Profesional-Regimen de Largo Plazo	48300.00
13131	Aporte Patronal Solidario 3%	
13200	Aporte Patronal para Vivienda	

Guardar Restablecer

SISEP - Mozilla Firefox

http://localhost/sep/v.04/mod_ereg.php

Sistema Web de Ejecución Presupuestaria

Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario:daf

REGISTRO DE EJECUCION PRESUPUESTARIA

Periodo: Diciembre UNIDAD: Todas las Unidades

Mostrar 10 registros

FECHA	C-31	PARTIDA	UNIDAD	GLOSA	TIPO	MONTO
08-12-2011	01	11700-Sueldos	FRESENCIA	Sueldo 1	Original	4,500.00
08-12-2011	02	11400-Aguinaldos	SECHEMOCLO	Aguinaldo 1	Original	9,000.00
FECHA	C-31	PARTIDA	UNIDAD	GLOSA	TIPO	MONTO

Mostrando 1 a 2 de 2 registros

SEP - Mozilla Firefox

SEP

http://localhost/sep/V.04/mod_registro_gasto.php

Sistema Web de Ejecución Presupuestaria

Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario:daf

Registro de Gastos por Unidad

Fecha: 09-12-2011 Tipo: Original

Prev.	Com.	Dev.	Pago	Unidad	Partida	Glosa	Monto
01	-	-	-	PRESIDENCIA	11700 Sueldos	Sueldo 1	4,500.00
02	-	-	-	SEC.HEMICICLO	11400 Aguinaldo	Aguinaldo 1	9,000.00
				Seleccione una Unidad	Seleccione Partida		

Agregar Gasto Guardar Registro

Mozilla Firefox

http://localhost/sep/V.04/mod_modif.php

http://localhost/sep/V.04/mod_modif.php

Sistema Web de Ejecución Presupuestaria

Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario:daf

MODIFICACIÓN S.I.G.M.A.

Nro. Documento: 001 Fecha Solicitud: 09-12-2011 Fecha Modificación: 09-12-2011

PARTIDAS PRESUPUESTARIAS A SER DISMINUIDAS

Partida	DESCRIPCIÓN	IMPORTE	TOTAL
11220	Bono de Antigüedad		-10,000.00
	SEC.HEMICICLO	-10,000.00	
TOTAL			-10000.00

PARTIDAS PRESUPUESTARIAS A SER INCREMENTADAS

Partida	DESCRIPCIÓN	IMPORTE	TOTAL
11700	Sueldos		10,000.00
	PRESIDENCIA	10,000.00	
TOTAL			10000.00

Modificar Presupuesto

Sistema Web de Ejecución Presupuestaria Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario: daf

REPORTE EJECUCION PRESUPUESTARIA

Fecha 1: 01-01-2011 Unidad Grupo Area
 Fecha 2: 09-12-2011 UNIDAD DIR INFORMATICA

Mostrar PDF

PARTIDA	PRESUPUESTO	MODIFICACION	PPES. AJUST.	EJECUTADO	SALDO	% EJECUCION
10000 SERVICIOS PERSONALES	50,000.00	3.00	50,000.00	0.00	50,000.00	0.00%
13110 Regimen de Corto Plazo (Salud)	20,000.00	-	20,000.00	-	20,000.00	0.00%
13120 Prima de Riesgo Profesional-Regimen de Largo Plazo	30,000.00	-	30,000.00	-	30,000.00	0.00%
20000 SERVICIOS NO PERSONALES	14,000.00	3.00	14,000.00	0.00	14,000.00	0.00%
21100 Comunicaciones	5,000.00	-	5,000.00	-	5,000.00	0.00%
21200 Energía Eléctrica	9,000.00	-	9,000.00	-	9,000.00	0.00%
30000 MATERIALES Y SUMINISTROS	392,000.00	3.00	392,000.00	0.00	392,000.00	0.00%
31110 Gastos Destinados al Pago de Retirofijos al Personal de las Inst. Públicas	150,000.00	-	150,000.00	-	150,000.00	0.00%
31120 Gastos Por Alimentación y Otros Similares efec. En Reuniones, Sem. Y otros Eventos	72,000.00	-	72,000.00	-	72,000.00	0.00%
31300 Productos Agrícolas, Pecuarios y Forestales	90,000.00	-	90,000.00	-	90,000.00	0.00%
32100 Papel	80,000.00	-	80,000.00	-	80,000.00	0.00%

ejec_por_fecha_pdf.php (application/pdf Objeto) - Mozilla Firefox

ASAMBLEA LEGISLATIVA DEL ESTADO PLURINACIONAL DE BOLIVIA
Cámara de Diputados

EJECUCION PRESUPUESTARIA AL 09 DE DICIEMBRE 2011
DIR. INFORMATICA

Partida	Presupuesto	Modificación	Total Ajust.	Ejecutado	Saldo	% Ejecución
10000 SERVICIOS PERSONALES	50,000.00	0.00	50,000.00	0.00	50,000.00	0.00%
13110 Regimen de Corto Plazo (Salud)	20,000.00	-	20,000.00	-	20,000.00	0.00%
13120 Prima de Riesgo Profesional-Regimen de Largo Plazo	30,000.00	-	30,000.00	-	30,000.00	0.00%
20000 SERVICIOS NO PERSONALES	14,000.00	0.00	14,000.00	0.00	14,000.00	0.00%
21100 Comunicaciones	5,000.00	-	5,000.00	-	5,000.00	0.00%
21200 Energía Eléctrica	9,000.00	-	9,000.00	-	9,000.00	0.00%
30000 MATERIALES Y SUMINISTROS	392,000.00	0.00	392,000.00	0.00	392,000.00	0.00%
31110 Gastos Destinados al Pago de Retirofijos al Personal de las Inst. Públicas	150,000.00	-	150,000.00	-	150,000.00	0.00%
31120 Gastos Por Alimentación y Otros Similares efec. En Reuniones, Sem. Y otros Eventos	72,000.00	-	72,000.00	-	72,000.00	0.00%
31300 Productos Agrícolas, Pecuarios y Forestales	90,000.00	-	90,000.00	-	90,000.00	0.00%
32100 Papel	80,000.00	-	80,000.00	-	80,000.00	0.00%
TOTAL EJECUTADO:	426,000.00	0.00	426,000.00	0.00	426,000.00	0.00%

SISEP - Mozilla Firefox

http://localhost/rep/V04/repote2.php

Sistema Web de Ejecución Presupuestaria Gestion:2011

Inicio Administración Presupuesto Gastos Modificación Reportes Usuario:df

REPORTE LIBRO MAYOR


Unidad UNIDAD: Todos los unidades PARTIDA: 11700 Sueldos
 Grupo
 Área

Fecha 1: 01-01-2011 Fecha 2: 09-12-2011

UNIDAD	FECHA	GLOSA	IP C-31	IMPORTE TOTAL
PRESENCIA	1-12-2011	Sueldo 1	01	4,500.00
PRESENCIA	1-12-2011	Sueldo 2	005	5,100.00
PRESENCIA	1-12-2011	Sueldo 3	006	3,500.00
TOTAL EJECUTADO:				13,100.00
TOTAL PRESUPUESTADO:				415,600.00
% DE EJECUCION:				3.15%

libro_mayor_pdf.php (application/pdf Objeto) - Mozilla Firefox

http://localhost/sep/V04/reportes/libro_mayor_pdf.php



ASAMBLEA LEGISLATIVA DEL ESTADO PLURINACIONAL DE BOLIVIA
Cámara de Diputados

MAYOR POR PARTIDA AL 09 DE DICIEMBRE-2011
11700-Sueldos

UNIDAD	FECHA	GLOSA	N° C-31	IMPORTE TOTAL
PRESENCIA	9-12-2011	Sueldo 1	01	4,500.00
PRESENCIA	9-12-2011	Sueldo 2	005	5,100.00
PRESENCIA	9-12-2011	Sueldo 3	006	3,500.00
TOTAL EJECUTADO:				13,100.00
TOTAL PRESUPUESTADO:				415,600.00
% DE EJECUCION:				3.15%

3.6 FASE DE PUESTA EN PRODUCCIÓN

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. Para ello se en esta fase se realizan casos de prueba, para determinar si son necesarias tareas de ajuste.

3.6.1 PRUEBAS DE ACEPTACIÓN

Esta plantilla muestra las acciones o peticiones que realiza el usuario y las respuestas de parte del sistema, también identifica las precondiciones²¹ y postcondiciones²².

Las figuras 3.21, 3.22, 3.23, 3.24 y 3.25 muestran un caso de prueba para cada tarea asociada.

Es decir:

Caso de Prueba 1	verifica	Tareas 1 y 2
Caso de Prueba 2	verifica	Tareas 3 y 4
Caso de Prueba 3	verifica	Tareas 9 y 10
Caso de Prueba 4	verifica	Tareas 11 y 12
Caso de Prueba 5	verifica	Tareas 5, 6, 7, 8, 9, y 10

En modelos tradicionales, la participación del usuario final es de vital importancia, para el éxito del proyecto, su participación en la fase de pruebas.

En este caso, la importancia de la participación del usuario final, abarca todas las fases del proyecto.

²¹ Requisitos que deben cumplirse para un normal desarrollo de las acciones del caso de uso.

²² Resultado esperado como consecuencia de la ejecución del caso de uso

CASO DE PRUEBA	Registro de presupuesto
ROLES	Usuario Operativo
TAREAS	T1, T2
PRECONDICIONES	Las partidas presupuestarias deben estar registradas, como también las unidades organizacionales.
DESCRIPCIÓN	
PETICIONES	RESPUESTAS
<p>1. El usuario accede al sistema web.</p> <p>2. El usuario selecciona la opción “Presupuesto” en el menú principal.</p> <p>4. El usuario elige una unidad organizacional</p> <p>6. El usuario selecciona un grupo de gasto.</p> <p>9. El usuario procede a guardar los montos introducidos</p>	<p>3. El sistema muestra el formulario, permitiendo seleccionar una unidad.</p> <p>5. El sistema despliega las partidas presupuestadas para la unidad seleccionada.</p> <p>7. El sistema despliega un formulario de “Registro de presupuesto”, mostrando únicamente las partidas que no están presupuestadas para la unidad</p> <p>8. Al ingresar los montos, el sistema válida los datos introducidos”.</p> <p>10.1 Si los montos introducidos no son válidos, el sistema despliega un cuadro de información”.</p> <p>10.2 Si los montos introducidos son válidos, el sistema procede a guardarlos, cerrando el formulario de registro.</p>
POSTCONDICIONES	<p>El sistema web actualiza los registros presupuestarios para la unidad correspondiente, añadiendo las nuevas partidas registradas.</p> <p>El sistema registra la fecha, hora, IP de usuario y nombre de usuario.</p>

Tabla 3.21: Caso de Prueba “Registro de presupuesto”

Fuente: Elaboración propia

CASO DE PRUEBA	Registro de gastos	
ROLES	Usuario Operativo	
TAREAS	T3, T4	
PRECONDICIONES	Los gastos a ser ejecutados deben corresponder a partidas presupuestadas, es decir, no puede haber ejecución presupuestaria sin partidas presupuestadas.	
DESCRIPCIÓN		
PETICIONES	RESPUESTAS	
<p>1. El usuario accede al sistema web.</p> <p>2. El usuario selecciona la opción "Gastos" en el menú principal.</p> <p>5. El usuario introduce el código del formulario, la fecha y el tipo de gasto.</p> <p>6. El usuario selecciona la unidad organizacional.</p> <p>8. El usuario selecciona la partida de gasto, para luego introducir la glosa y el importe en bolivianos.</p> <p>9. El usuario agrega el gasto introducido.</p> <p>11.1 Si el usuario desea agregar un nuevo gasto, vuelve al paso 6.</p> <p>11.2 El usuario finaliza el registro de gastos.</p>	<p>3. El sistema despliega el formulario para registro de gastos, solicitando todos los datos necesarios.</p> <p>4. El sistema despliega únicamente las unidades que han sido presupuestadas.</p> <p>7. El sistema despliega las partidas que han sido presupuestadas para la unidad organizacional seleccionada</p> <p>10.1 Si el monto introducido no es válido, el sistema despliega un cuadro de información".</p> <p>10.2 Si el monto introducido es válido, el sistema agrega al gasto a una tabla de gastos, habilitando un nuevo cuadro para añadir un nuevo gasto.</p> <p>12.1 Si, el código de formulario, la fecha y el tipo de gasto son correctos, el sistema registra el gasto,</p> <p>12.2 Si la validación es incorrecta, el sistema despliega un cuadro de información.</p>	
POSTCONDICIONES	<p>El sistema web despliega el cuadro de "documentos de gastos" efectuados en el periodo actual.</p> <p>El sistema registra la fecha, hora, IP de usuario y nombre de usuario.</p>	

Tabla 3.22: Caso de Prueba "Registro de gastos"

Fuente: Elaboración propia

CASO DE PRUEBA	Registro de modificación presupuestaria	
ROLES	Usuario Operativo	
TAREAS	T9, T10	
PRECONDICIONES	Las partidas a ser modificadas deben haber sido previamente presupuestadas.	
DESCRIPCIÓN		
PETICIONES	RESPUESTAS	
<p>1. El usuario accede al sistema web.</p> <p>2. El usuario selecciona la opción “Modificación” en el menú principal.</p> <p>4. El usuario selecciona la partida a ser disminuida, para luego introducir el monto correspondiente y confirmar acción.</p> <p>6. El usuario selecciona una o varias unidades, a la vez introduce los montos a ser disminuidos y confirma acción.</p> <p>8. El usuario selecciona la partida a ser incrementada, para luego introducir el monto correspondiente y confirmar acción.</p> <p>9. El usuario selecciona una o varias unidades, a la vez introduce los montos a ser incrementados.</p>	<p>3. El sistema muestra el formulario, de modificación presupuestaria, dividido en dos partes: partidas a ser incrementadas y partidas a ser disminuidas</p> <p>5. El sistema despliega todas las unidades que tienen presupuestada la partida seleccionada.</p> <p>7. 1 Si los montos introducidos son correctos, despliega el cuadro de partidas disminuidas.</p> <p>7. 1 Si los montos introducidos son incorrectos, despliega un cuadro de información.</p> <p>8. El sistema despliega todas las unidades.</p> <p>10.1 Si los montos introducidos no son válidos, el sistema despliega un cuadro de información”.</p> <p>10.2 Si los montos introducidos son válidos, el sistema procede a guardarlos, cerrando el formulario de modificación presupuestaria.</p>	
POSTCONDICIONES	<p>El sistema web actualiza los registros presupuestarios para la unidad, efectuando los incrementos y disminuciones registradas.</p> <p>El sistema registra la fecha, hora, IP de usuario y nombre de usuario.</p>	

Tabla 3.23: Caso de Prueba “Registro de modificación presupuestaria”

Fuente: Elaboración propia

CASO DE PRUEBA	Consulta de reportes	
ROLES	Usuario Operativo, Usuario Administrador	
TAREAS	T11, T12	
PRECONDICIONES	Se debe llevar a cabo el registro presupuestario y el registro de gastos.	
DESCRIPCIÓN		
PETICIONES	RESPUESTAS	
<p>1. El usuario accede al sistema web.</p> <p>2. El usuario selecciona la opción "Reportes" en el menú principal, para luego indicar el tipo de reporte.</p> <p>4.1 El usuario indica un rango de fechas y el tipo de filtrado organizacional.</p> <p>4.2 El usuario indica un rango de fechas, el tipo de filtrado organizacional y una partida de gasto.</p> <p>6. El usuario selecciona una unidad, grupo o componente organizacional.</p> <p>7. El usuario solicita al sistema generar el reporte.</p>	<p>3.1 El sistema despliega el cuadro de control para emisión de Reporte de ejecución presupuestaria.</p> <p>3.1 El sistema despliega el cuadro de control para emisión de Reporte de libro mayor por partida.</p> <p>5. El sistema despliega las unidades, grupos o componentes organizacionales, de acuerdo al filtrado seleccionado</p> <p>8. El sistema despliega el reporte solicitado, habilitando la opción a exportar en formato PDF.</p>	
POSTCONDICIONES	El sistema registra la fecha, hora, IP de usuario y nombre de usuario que accedió al reporte.	

Tabla 3.24: Caso de Prueba "Consulta de reportes"

Fuente: Elaboración propia

CASO DE PRUEBA	Administración de usuarios y partidas de gasto	
ROLES	Usuario Administrador	
TAREAS	T5, T6, T7, T8, T9, T10	
PRECONDICIONES	Debe existir al menos un usuario registrado con rol de Administrador.	
DESCRIPCIÓN		
PETICIONES	RESPUESTAS	
<p>1. El usuario accede al sistema web.</p> <p>2. El usuario selecciona la opción "Administración" en el menú principal, para luego indicar si desea administrar partidas o usuarios.</p> <p>4.1 El usuario puede realizar una búsqueda por código de partida o descripción o añadir una nueva partida.</p> <p>4.1 El usuario puede realizar una búsqueda por nombre de usuario, nombre real o unidad a la que pertenece. También puede añadir un nuevo usuario.</p> <p>6.1 Para ambos casos de búsqueda, los resultados obtenidos incluirán opciones de eliminación y edición.</p> <p>6.2 El usuario introduce los datos del nuevo registro y procede a guardarlos.</p>	<p>3.1 El sistema despliega el cuadro de control para administración de partidas.</p> <p>3.2 El sistema despliega el cuadro de control para administración de usuarios.</p> <p>5. 1 En caso de búsqueda de partida el sistema indica si está registrada, en caso de añadir nueva partida, despliega el formulario de ingreso de partida.</p> <p>5. 2 En caso de búsqueda de usuario indica si este está registrado, en caso de añadir nuevo usuario, despliega el formulario de ingreso de usuario.</p> <p>7.1 El sistema solicita confirmación de eliminación y edición de registros</p> <p>7.2 El sistema valida los datos introducidos y si son correctos procede a guardarlos, desplegando un mensaje de confirmación, caso contrario despliega mensaje de información y corrección de datos.</p>	
POSTCONDICIONES	Nuevas partidas pueden ser usadas en la planificación presupuestaria, como también nuevos usuarios pueden ser habilitados para ingresar al sistema, y viceversa.	

Tabla 3.25: Caso de Prueba: Administración de usuarios y partidas de gasto

Fuente: Elaboración propia

CAPITULO 4

CALIDAD DE SOFTWARE

Para medir la calidad del Sistema Web de Control y Seguimiento de la Ejecución Presupuestaria, se va a emplear el estándar ISO 9126-1.

4.1 ISO 9126-1

4.1.1 CONFIABILIDAD

El análisis de confiabilidad se basara en que el sistema opera en puntos discretos del tiempo. Se debe considerar que:

- El sistema entero falla si uno de sus componentes falla, esto es que el caso de que el sistema presente sus componentes en serie, para lo cual se tiene.

$$P = P1 P2 P3..... Pn$$

Donde:

P_i = Probabilidad de que el componente i -esimo falle

P = Probabilidad de que el sistema falle.

- El sistema falla si y solo si todos sus componentes fallan. Esto en el caso que el sistema presente todos sus componentes en paralelo para lo cual se tiene:

$$R = 1 - (1-R1) (1-R2) (1-R3) (1-Rn)$$

Donde:

R1: Confiabilidad del módulo de Presupuesto.

R2: Confiabilidad del módulo de Gastos.

R3: Confiabilidad del módulo de Modificación Presupuestaria.

R4: Confiabilidad del módulo de Reportes.

R5: Confiabilidad del Módulo de Control y Seguimiento de Usuarios

Asignando valores de error a cada uno de estos módulos (posibles errores de proceso). Tenemos:

R1: 99% con error de %1 = 0.01

R2: 97% con error de %3 = 0.03

R3: 95% con error de %6 = 0.05

R4: 97% con error de %3 = 0.03

R5: 99% con error de %2 = 0.01

Analizando todo el sistema tenemos

$$\begin{aligned} R_{\text{sistema}} &= 1 - [(1-R1) + (1-R2) + (1-R3) + (1-R4) + (1-R5)] \\ &= 1 - (1 - 0.99) (1 - 0.97) (1 - 0.95) (1 - 0.97) (1 - 0.91) \\ &= 0.87 \end{aligned}$$

Se tiene por lo tanto un 87% de confiabilidad,

Esto quiere decir que existe una probabilidad de 13% de que cualquiera de los componentes falle.

4.1.2 FUNCIONALIDAD

La funcionalidad es medida a través del **Punto Función (PF)**, que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión de usuario final de la aplicación. [Pressman, 2003]

Para medir la funcionalidad del sistema se deben determinar las siguientes cinco características:

- ✓ Número de salidas de usuario
- ✓ Número de Peticiones de Usuario
- ✓ Número de Archivos.
- ✓ Número de intercales externas

Para calcular el punto función se utiliza la siguiente relación:

$$PF = CuentaTotal * (X + Min(Y) * \sum F_i)$$

Donde:

PF: Medida de funcionalidad.

Cuenta Total: Es la suma de los siguientes datos: No de Entradas, No de Salidas, No de Peticiones, No de Archivos y No de Interfaces externas.

X: Confiabilidad del proyecto, varía entre 1% a 100%.

Min(Y): Error mínimo aceptable al de la complejidad.

$\sum F_i$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Tomando en cuenta estos pasos, inicialmente definimos los siguientes puntos:

4.1.2.1 NÚMERO DE ENTRADAS DE USUARIO

Representa cada entrada de control del usuario que llega desde la frontera externa y que proporciona diferentes datos de aplicación.

No	ENTRADAS DE USUARIO	CANTIDAD
1	Pantalla de Búsqueda de Presupuesto	5
2	Pantalla de Registro de Presupuesto	13
3	Pantalla de Búsqueda de Gasto	5
4	Pantalla de Registro de Gastos	10
5	Pantalla de Modificación Presupuestaria	10
6	Pantalla de Autenticación de Usuarios	1
7	Pantalla de Búsqueda de Usuarios	3
8	Pantalla de Búsqueda de Partidas	3
9	Pantalla Reporte Ejecución Presupuestaria	4
10	Pantalla Reporte Libro Mayor por Partida	5
Total		59

Tabla 4.1: Cálculo del Número de Entradas de Usuario

Fuente: Elaboración Propia

4.1.2.2 NÚMERO DE SALIDAS DE USUARIO

Representa cada salida de información referente a la aplicación. En este contexto las salidas se refieren a informes, pantallas, mensajes de error.

No	SALIDAS DE USUARIO	CANTIDAD
1	Alerta de usuario no identificado	1
2	Alerta de montos inválidos	6
3	Aviso de confirmación de eliminación de registros	4
4	Reporte Versión Imprimible Ejecución Presupuestaria	1
5	Reporte Versión Imprimible Libro Mayor por Partida	1
6	Aviso de confirmación inserción de datos	3
7	Aviso de error al insertar registros	3
Total		19

Tabla 4.2: Cálculo del Número de salidas de usuario

Fuente: Elaboración Propia

4.1.2.3 NÚMERO DE PETICIONES DE USUARIO

Se representa cada combinación, entrada/ salida única, donde una entrada causa y genera una salida inmediata.

No	ARCHIVO	CANTIDAD
1	Búsqueda de Presupuesto Anual por Unidad	1
2	Listado de Gastos por Unidad	1
3	Listado de Gastos por Periodo	1
4	Reporte Vista Preliminar Ejecución Presupuestaria	1
5	Reporte Vista Preliminar Libro Mayor por Partida	1
6	Búsqueda de Usuarios por cualquier atributo	1
7	Búsqueda de Partidas por cualquier atributo	1
Total		7

Tabla 4.3: Cálculo del número de peticiones de usuario

Fuente: Elaboración Propia

4.1.2.4 NÚMERO DE ARCHIVOS

Representa cada archivo que puede ser una parte en una gran base de datos o un archivo independiente.

No	ARCHIVO	CANTIDAD
1	Presupuesto Anual	1
2	Gastos/Ejecución presupuestaria	1
3	Modificaciones al Presupuesto	1
4	Clasificador de Partidas	1
5	Log de Acciones	1
6	Unidades Organizacionales	1
Total		6

Tabla 4.4: Cálculo del número de archivos

Fuente: Elaboración Propia

4.1.2.5 NÚMERO DE INTERFACES EXTERNAS

Representan todas las interfaces legibles por la máquina por ejemplo: archivos de datos, archivos de datos, en cinta o discos que son utilizados para transmitir información a otro sistema.

No	ARCHIVO	CANTIDAD
1	CD's	1
2	Flash Memory	1
Total		2

Tabla 4.5: Cálculo de Interfaces externas

Fuente: Elaboración Propia

La tabla 4.6 resume todo lo mencionado anteriormente:

PARÁMETRO DE MEDIDA	CUENTA		FACTOR DE PONDERACIÓN			=	TOTAL
			SIMPLE	MEDIO	COMPLEJO		
No de entradas de usuario	59	X	3	4	6	=	236
No de salidas de usuario	19	X	4	5	7	=	95
No de peticiones de usuario	7	X	3	4	6	=	28
No de archivos	6	X	7	10	15	=	60
No de interfaces externas	2	X	5	7	10	=	14
Cuenta Total							433

Tabla 4.6: Cálculo de la Cuenta Total

Fuente: Elaboración Propia

Cada organización que utiliza métodos de puntos desarrolla criterios para determinar si una entrada es denominada *simple*, *media* o *compleja*. No obstante la determinación de la complejidad es algo subjetivo. Para nuestro caso utilizamos un factor de ponderación **medio** [Pressman, 2003]. Los valores de ajuste de complejidad se obtienen de la siguiente tabla:

No	CARACTERÍSTICAS DEL SISTEMA	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
		0	1	2	3	4	5	
1	Requiere el sistema de copias de seguridad y de recuperación						X	5
2	Se requiere comunicación de datos					X		4
3	Existen funciones de procesamiento distribuido		X					1
4	El rendimiento es crítico			X				2
5	Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado					X		4
6	Se requiere entrada de datos en línea	X						0
7	Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas opciones.						X	5
8	Se actualizan los datos maestros de forma interactiva						X	5
9	Son complejas las entradas, las salidas, los archivos o las peticiones		X					1
10	Es complejo el procesamiento interno				X			3
11	Se ha diseñado el código reutilizable					X		4
12	Están incluidas en el diseño la conversión y la instalación			X				2
13	Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones						X	5
14	Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario						X	5
FACTOR DE COMPLEJIDAD								46

Tabla 4.7: Cálculo del Factor de Complejidad

Fuente: Elaboración Propia

Luego para calcular el punto Función (PF) se emplea la siguiente relación

$$PF = CuentaTotal * (X + Min(Y) * \sum F_i)$$

$$PF = CuentaTotal * (0.65 + 0.01 * 46)$$

Con $X=0.65$ (Pie de corrección), $Min(Y)=0.01$ (Error de la confiabilidad del sistema)

$$\sum F_i = 46 \text{ y } CuentaTotal = 433$$

Reemplazando:

$$PF = CuentaTotal * (X + Min(Y) * \sum F_i)$$

$$PF = 433 * (0.65 + 0.01 * 46)$$

$$PF = 480,63$$

Ahora realizamos la comparación de funcionalidad del sistema con el PF máximo que se puede alcanzar es decir que $\sum F_i = 70$.

$$PF = 433 * (0.65 + 0.01 * 70)$$

$$PF = 584,55$$

Con máximos valores de ajuste a la complejidad se tiene que la funcionalidad real es:

$$FUNCIONALIDAD = \left(\frac{480,63}{584,55} \right) * 100$$

$$FUNCIONALIDAD = 82,22\%$$

El Sistema Web para Control y Seguimiento de la Ejecución Presupuestaria tiene una funcionalidad del 82.22 %.

Este resultado se traduce, como una medida de magnitud objetiva y cuantitativa respecto a la magnitud del Sistema, desde el punto de vista del usuario final, es decir 82%

4.1.3 FIABILIDAD

La fiabilidad es una característica fundamental para el reutilización del software, para su estimación se toma en cuenta la frecuencia y magnitud de los errores de sistema. Una ventaja de haber utilizado una metodología ágil es que la magnitud de los errores es mínima, ya que la temprana producción de código proporciona retroalimentación constante.

Para la obtención de la fiabilidad se aplica la siguiente ecuación:

$$\text{Fiabilidad} = 1 - \left(\frac{\# \text{Errores}}{\# \text{Lineas deCodigo}} \right)$$

Para la estimación de # Errores se desarrollo un formulario de entrega para cada entregable funcional durante la fase de puesta en marcha.

Reemplazando valores, obtenemos:

$$\text{Fiabilidad} = 1 - \left(\frac{45}{5230} \right) = 0.9913$$

$$\text{Fiabilidad} = 0.99666 * 100\% = 99.13$$

$$\text{Fiabilidad} = 99.13 \%$$

De esta manera, el Sistema Web posee un coeficiente de 99.13% fiabilidad, dicho valor se traduce en que código fuente del Sistema Web es reutilizable en un 99%.

4.1.4 PORTABILIDAD

La portabilidad es una característica que indica la capacidad del sistema de funcionar correctamente independientemente de la plataforma tecnológica sobre la cual opera.

Para la estimación, se aplica de la siguiente ecuación:

$$\text{Portabilidad} = 1 - \left(\frac{\# \text{ dias para portar el sistema}}{\# \text{ dias para implementar sistema}} \right)$$

Reemplazando valores se tiene:

$$\text{Portabilidad} = 1 - \left(\frac{1}{20}\right) = 0.966666$$

$$\text{Portabilidad} = 0.966666 * 100\% = 96.66\%$$

$$\text{Portabilidad} = 96.6\%$$

El Sistema Web posee un coeficiente de portabilidad de 96.6%. Esto se traduce como la capacidad del mismo para funcionar un 96%.6 de la totalidad de plataformas tecnológicas.

4.1.5 FLEXIBILIDAD

Esta característica hace referencia al coste de realizar modificaciones al sistema web, para su estimación se tomó en cuenta modificaciones anteriores sobre distintos módulos registrando el tiempo promedio por modificación:

Para la obtención de la flexibilidad, se aplica la siguiente ecuación:

$$\text{Flexibilidad} = 1 - 0.05 (\# \text{ medio de días} - \text{hombre por cambio})$$

Reemplazando valores se tiene:

$$\text{Flexibilidad} = 1 - 0.05 (3-1) = 0.9$$

$$\text{Flexibilidad} = 90\%$$

De esta manera se puede decir que el sistema en, cuanto al coste de modificación, es flexible en un 90 %, del total estimado por la modificación.

4.1.6 RESULTADOS

CARACTERISTICAS	RESULTADO
Usabilidad	90%
Funcionalidad	82.22%
Fiabilidad	99.13%
Portabilidad	96.6%
Flexibilidad	90%
Evaluación de calidad total	91.59

Tabla 4.7: Calidad total ISO 9126-1

Fuente: Elaboración Propia

El factor de calidad total está directamente relacionado con el grado de satisfacción de un usuario que ingresa al sistema.

4.2 COCOMO

Las características del Sistema Web permiten aplicar el Modelo Empírico de Análisis de costo y beneficio COCOMO

Para hallar el esfuerzo, se tiene la siguiente ecuación:

$$E = a * (KLDC)^b \text{ [personas/mes]}$$

Donde:

- **E** es el esfuerzo expresado en personas por mes.
- **a** Constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (**a=2.94**)
- **b** Es el factor exponencial para considerar la economías y deseconomías de escala.
- **KLDC** es un número estimado de código fuente en miles de líneas

Reemplazando los datos en la ecuación, se tiene:

$$E = 2.94 * (3)^{0.7} = 6.34$$

$$E = 6 \text{ [personas/mes]}$$

El esfuerzo aplicado para la realización del sistema, es de 6 personas por mes. Para la obtención del tiempo empleado para el desarrollo del sistema, se hace uso de la siguiente ecuación:

$$T = c * E^d \text{ [meses]}$$

Donde:

- T es el tiempo de desarrollo expresado en meses
- c, d son constantes empíricas
- E es el esfuerzo expresado en personas por mes.

Reemplazando los datos en la ecuación, se tiene:

$$T = 2.4 * (6)^{0.4} = 4.9$$

$$T = 5 \text{ [meses]}$$

El tiempo aproximado de desarrollo del Sistema Web, es de 5 meses.

Los costos de este sistema se distribuyen de la siguiente manera:

Descripción	Cantidad	# de Personas	Costo	Costo total
Análisis y Diseño	1 Mes	2	\$us 560	\$us 1120
Programación	5	6	\$us 290	\$us 8700
Capacitación	20 Hrs.	1	\$us 10	\$us 200
TOTAL				\$ 10.020

Bs. 70.140

Tabla 4.8: Análisis de costos del proyecto

Fuente: Elaboración Propia

4.3 VAN-TIR

También conocido valor actualizado neto (en inglés *Net present value*), cuyo acrónimo es VAN (en inglés NPV), es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión, dado que el presente proyecto se llevó a cabo en una institución del estado, con naturaleza de contribución social y no habiendo inversión inicial, la interpretación de los resultados nos darán indicadores de rendimiento y eficiencia.

Tasa de descuento 3%				
AÑOS	PAGOS	AHORROS	RESULTADO	VALOR ACTUAL
0	24.500		-24.500	-24.500
1		5.000	5.000	4854.36
2		7.000	7.000	6598.17
3		10.000	10.000	9151.41
4		13.000	13.000	11.550.33
	24.500	35.000	10.500	VAN=7654.27
				TIR=13.52%

Tabla 4.9: Análisis de Eficiencia y Rendimiento basado en el modelo VAN-TIR

Fuente: Elaboración Propia

El valor de la TIR, significa para este caso, un aumento del 13.52% en la eficiencia de los procesos de la ejecución presupuestaria, en un tiempo de 4 años.

Luego, podemos estimar de manera subjetiva:

$$\frac{C}{D} = \sum \frac{ahorro}{C} = \frac{35000}{24500} = 1.42$$

De este resultado podemos concluir que, en la realización de este proyecto; por cada boliviano invertido se ahorra 0.42 Bs.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Luego de haber concluido la elaboración del presente proyecto, se ha llegado a las siguientes conclusiones:

- ✓ Se ha logrado aplicar la metodología WebML para el análisis, diseño y desarrollo del producto, cumpliendo con cada una de sus fases.
- ✓ Se ha llevado a cabo los procesos y tarea de acuerdo al cronograma establecido, se han desarrollado los siguientes módulos.
 1. Módulo de Presupuesto
 2. Módulo de Gastos
 3. Modificación Presupuestaria
 4. Módulo de Reportes
 5. Módulo Control y Seguimiento de Usuarios.
 6. Modulo de Administración de Partidas, Unidades y Usuarios
- ✓ Se han aplicado métricas de calidad de Software, y análisis de costo y beneficio en términos de rendimiento y eficiencia.

5.2 RECOMENDACIONES

Las siguientes recomendaciones tienen como propósito respaldar la implementación de proyectos similares.

- ✓ Se recomienda la implementación de mecanismos de seguridad, para el acceso a la información del sistema a fin de garantizar el correcto funcionamiento del producto, como por ejemplo, el seguimiento continuo al registro de ingresos y acciones del Sistema Web.

- ✓ Se debe realizar un continuo mantenimiento al sistema, para conseguir un buen desempeño del producto.
- ✓ Es importante que cualquier proyecto similar al presente, cuente con el respaldo de la documentación de procesos funcionales del sistema.
- ✓ Se recomienda el uso de una metodología ágil, y un lenguaje de modelado específico, para proyectos de similar naturaleza, donde los requerimientos no están definidos desde el inicio del proyecto.



BIBLIOGRAFIA

- [Ceri et al., 2000] Ceri, S., Fraternali, P, Bongio 2000. – Lenguaje de Modelado Web (WebML): Un lenguaje para diseñar sitios Web, Computer Networks. USA
- [Baresi, Garzotto, Paolini 2001] Baresi L., Garzotto F., Paolini P. 2001 Extendiendo UML para el modelado Web Modeling Web Applications. Pittsburgh, USA.
- [Jeffries, R. & Anderson 2001] “Addison-Wesley. 2001 – Programación Extrema Instalada”.
- [Ian Sommerville, 2001] Ian Sommerville, “*Ingeniería de Software*”, 2001, Sexta Edición. USA
- [Newkirk, J. & Martin R.C 2001] Newkirk, J., Martin R.C, 2001 “Prácticas de Programación Extrema”. USA
- [Wake & W.C. 2002] “Addison-Wesley. 2002. Exploración de Programación Extrema”. USA
- [César F.& Acebal 2002] César F.& Acebal 2002 – Programación Extrema (XP): un nuevo método de desarrollo de software. Depto. de Informática, Área de Lenguajes y Sistemas Informáticos, Universidad de Oviedo, USA

- [Hans Van Vliet, 2002] Hans Van Vliet, 2002 “Principios y Prácticas de la Ingeniería de Software”, Tercera edición, USA
- [Brambilla &Manolescu, 2002] Brambilla &Manolescu, 2002 – Diseño de Aplicaciones Web, Ediciones “Morgan Kaufmann” Publicado en San Francisco, USA
- [Beck, K. 2003] “Addison Wesley, 2003 - “Una explicación de la programación extrema. Aceptar el cambio”, USA
- [Pressman, 2003] Pressman Roger – 2003: “Ingeniería de Software – Un enfoque Práctico”, 5ta Edición, McGraw-Hill, España
- [Cabezas, 2004] Luis Miguel Cabezas Granado, 2004 - Manual Imprescindible de PHP 5”: “Ediciones Anaya Multimedia (GRUPO ANAYA, S.A.), Madrid – España.
- [Pastor, 2008] Oscar Pastor, 2008, Metodologías de Diseño de Aplicaciones Web, Universidad Politécnica de Valencia, España
- [INTECO, 2009] Ken Schwaber, 2009, Ingeniería del software: metodologías y ciclos de vida. Laboratorio Nacional de Calidad del Software de INTECO (Instituto Nacional de Tecnologías de la Comunicación), España.

REFERENCIAS DE INTERNET

- [Beck & Beedle, 2001] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, Febrero 2001
Manifesto for Agile Software Development
<http://agilemanifesto.org/>
- [Steward Baird, 2002] Prentice Hall, Professional Technical Reference
Marzo 22, 2002 - XP: A Project Management
<http://www.phptr.com/articles/article.asp?p=2606>
- [Soto & Palma, 2004] Soto R., Palma W. – 2004: "Propuesta de un modelo navegacional"
http://www.inf.ucv.cl/rsoto/papersPUV/Propuesta_de_un_modelo_navegacional.pdf
- [Martin Fowler, 2005] Martin Fowler, 2005 - La Nueva Metodologia
<http://martinfowler.com/articles/newMethodology.html>

ANEXOS



ANEXO A – GLOSARIO

WEBML (Siglas en Ingles de Lenguaje de Modelado WEB)

Lenguaje conceptual para diseño de alto nivel de aplicaciones web intensivas en el manejo de datos. WebML soporta una colección de conceptos poderosos que posibilitan un diseño de alto nivel y provee especificaciones gráficas para producir una descripción (a nivel abstracto) de la aplicación Web.

WebML combina técnicas de modelado Entidad Relación con UML, se basa en la distribución de nodos en los niveles del hipertexto sobre las páginas del nivel de presentación.

EXTREME PROGRAMMING XP (Siglas en Ingles de Programación Extrema)

Extreme Programming (XP) surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Las metodologías de desarrollo de software tradicionales.

XP es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo

SITIO WEB.

Sitio Web (Web Site), es una colección de archivos creados para Internet (páginas, gráficos, multimedia) sobre un mismo tema. Se puede acceder a un sitio a través de su dirección o URL, desde donde se puede llegar a las demás páginas o archivos en su sitio. Un sitio Web no necesariamente debe localizarse en el sistema de cómputo. Los documentos que integran el sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país.

El único requisito es que el equipo en el que se residan los documentos esté conectado a la red mundial de Internet. Este equipo de cómputo o Servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios.

NBSCI (NORMAS BÁSICAS DEL SISTEMA DE CONTABILIDAD INTEGRADA)

Las Normas Básicas del Sistema de Contabilidad Integrada (NBSCI), constituyen el instrumento técnico que establece los principios y las normas contables que proporcionan al Sistema de Contabilidad Integrada (SCI) una base conceptual única y uniforme a ser observada en la preparación de Estados Financieros del sector público.

NBSP (NORMAS BÁSICAS DEL SISTEMA DE PRESUPUESTO)

Conjunto ordenado y coherente de principios, disposiciones normativas, procesos e instrumentos técnicos, que tiene por objeto la formulación, aprobación, ejecución, seguimiento y evaluación, de los presupuestos públicos, según los programas y proyectos definidos en el marco del Plan General de Desarrollo Económico y Social del estado plurinacional de Bolivia.

Asimismo tiene por objeto la generación y obtención de información confiable, útil y oportuna sobre la ejecución presupuestaria, para la evaluación del cumplimiento de la gestión pública.

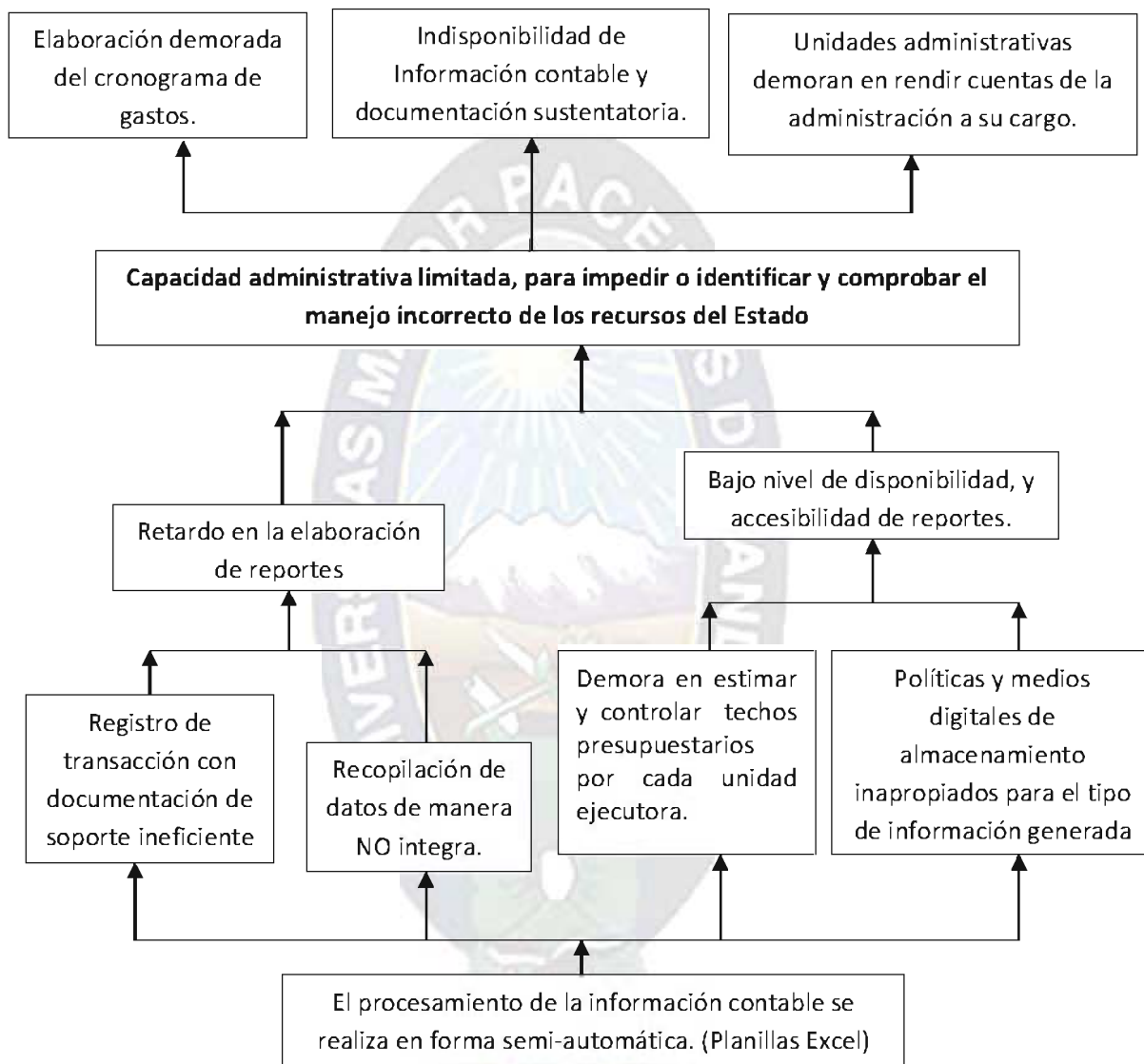
ANEXO B – ANALISIS DE INVOLUCRADOS

GRUPOS	INTERESES	PROBLEMAS PERCIBIDOS	RECURSOS Y MANDATOS
Dirección Administrativa Financiera	Generar información contable desagregada por cada unidad.	No se disponen de medios tecnológicos adecuados	Automatizar el ingreso de registros contables
Unidades Ejecutoras	Precisan estado actualizado de techo presupuestario.	Demora al establecer los límites máximos de endeudamiento.	Optimizar la generación de reportes, en función al tiempo
Dirección General de Asuntos Administrativos	Recepciona, revisa y remite la Solicitud de Modificación Presupuestaria para su análisis correspondiente	Demora al clasificar el tipo de Modificación si corresponde a un Presupuesto Adicional, a un Traspaso Interinstitucional o si se trata de un Traspaso Intrainstitucional.	Emitir Reportes en el tiempo mínimo, y formato requerido por esta Dirección

Fuente: Elaboración Propia (2011)

ANEXO C – ARBOL DE PROBLEMAS

Efectos



Causas

ANEXO D – ARBOL DE OBJETIVOS

