

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Análisis y diseño del módulo Presentación del sistema del Convenio Integral de Cooperación Cuba-Venezuela.

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

Autores: Martin Villalón Cruzata.

Yoilan Zayas Heras.

Tutor: Ing. Juan Carlos Montané Izaguirre.

Ciudad de la Habana, 2008-2009

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

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

Yoilan Zayas Heras

Martin Villalón Cruzata

(Autor)

(Autor)

Juan Carlos Montané Izaguirre

(Tutor)

Agradecimientos

A nuestros familiares por contribuir a nuestra formación y por la confianza depositada en nosotros.

A nuestro tutor, por su abnegada y constante preocupación por este trabajo, por su experiencia y profesionalismo a la hora de inculcarnos los conocimientos.

A las personas que crearon nuestra querida universidad, que se convirtió durante 5 años en nuestro hogar y por las incontables experiencias aquí vividas.

A los profesores, compañeros y amigos que ayudaron de una u otra forma al desarrollo de este trabajo.

A todos muchas gracias...

Dedicatoria

A ustedes, sí, mis padres, porque son los inspiradores de este logro alcanzado, porque son ustedes la mejor escuela que he tenido, gracias por confiar en mí.

A mis adorables hermanitas, que siempre están al pendiente de mí, por tenerme en lo más alto, a ustedes también gracias por confiar en mí. Por supuesto cuñita eres mi hermano, gracias por todo lo que has hecho por mí, gracias mi brother.

A mi familia que de una forma u otra siempre me ha tendido la mano, a ustedes también va esta dedicatoria. A mi tío Nelson que sus consejos los llevo conmigo siempre, gracias por ser así conmigo.

A mis hermanos de la 313, ustedes son lo máximo, sin ustedes fuera uno más en este mundo, gracias por ser los mejores amigos. A mi compañero de tesis Martín por ser el mejor jefe que voy a tener en mi vida, gracias por tenderme tu mano.

A tí, mi hermosa princesa, por ser la persona más dulce que ha tocado mi corazón, por todo el inmenso apoyo que siempre me has dado, por todos tus consejos, por enseñarme tantas cosas, por ser mi inspiración para todo, por mostrarme verdaderamente quien soy, gracias mi niña, gracias por confiar en mí. Te amo.

A mis dos principitos Brayitan y Chris, lucharé siempre por ser ejemplo para ustedes.

A todos con cariño y respeto, a mis padres, los mejores del mundo, a mis dos adorables hermanas, a mi hermosa princesa.

COX1234

Dedicatoria

A mis padres por hacerme la persona que soy, por sentir suyos mis nervios, mis problemas; por apoyarme y confiar en mí en todo momento, ustedes son los mejores.

A mi hermana Nalvis: mona, eres la mejor hermana del mundo, mi amiga y mi guía, te quiero mucho.

A mi sobrinito Ray: por convertirte en la alegría de todos en la casa.

A mi hermana Niovis: siempre te voy a tener en el lugar que te corresponde y te mereces.

Al Cory, compañero de proyectos y tesis, amigo incondicional.

A mi niñita: Ari eres lo más lindo que me pasó en la UCI, gracias por apoyarme y aguantar mis malcriadeces, te amo mi chiquitica.

A todos mis amigos y personas maravillosas que me he tenido la suerte de conocer en mi vida.

Todo lo que soy es gracias a ustedes

Martin

Resumen

La República de Cuba y la República Bolivariana de Venezuela, en aras de intensificar sus relaciones diplomáticas en los marcos del ALBA, mantienen en conjunto un acuerdo de colaboración. Dicho acuerdo comprende la concepción y ejecución de un gran número de proyectos, en el marco de la Comisión Mixta Cuba-Venezuela.

El control de los proyectos se efectúa de forma semiautomática, mediante reuniones de coordinación, por vía telefónica o por correo electrónico entre las partes implicadas, lo cual dificulta la comunicación y manipulación de los detalles de los proyectos ejecutados. Para darle soporte a la gestión a los mismos, se crea el Proyecto Convenio Cuba-Venezuela (CCV), el cual consta de 7 módulos entre los cuales se encuentra el módulo Presentación.

El proceso de presentación constituye la etapa donde se elaboran los proyectos, se trazan las metas y objetivos, se destinan los recursos que se invertirán en la ejecución de los mismos y se someten a un proceso de revisión transitando por niveles superiores. Finalmente se dejan listos para su ejecución una vez que se les asigna el financiamiento.

En el presente trabajo se realiza el análisis y diseño del módulo Presentación con el objetivo de obtener un sistema bien especificado. Para ello se identificaron, analizaron y especificaron los requerimientos y casos de uso del sistema. Además se generaron los artefactos correspondientes al modelo de diseño: subsistemas, paquetes, modelo de datos, diagramas de clases y diagramas de iteración exactamente los diagramas de secuencia. Finalmente se validan los artefactos obtenidos, tanto en el modelo del sistema como en el modelo de diseño en función de la calidad.

Índice de Figuras

Figura 1 Proceso unificado de desarrollo de software (fases y flujos de trabajos).....	7
Figura 2 Actividades del diseño en XP.....	11
Figura 3 Fases de MSF.....	12
Figura 4 Elementos de UML.....	29
Figura 5 Relaciones de dependencia de UML.....	29
Figura 6 Relaciones de asociación de UML.	29
Figura 7 Relación de generalización de UML.....	30
Figura 8 Actores del sistema.....	52
Figura 9 Diagrama de casos de uso del sistema.....	58
Figura 10 Subsistemas de diseño del Módulo Presentación del sistema CCV	78
Figura 11 Paquetes de diseño del Módulo Presentación del sistema CCV	79
Figura 12 Diagrama de clases de diseño, CU Gestionar Propuesta de Proyecto.	80
Figura 13 Diagrama de clases de diseño, CU Gestionar Plan Operativo.....	81
Figura 14 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Crear.	82
Figura 15 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Modificar.....	82
Figura 16 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Eliminar.	83
Figura 17 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Aceptar.....	83
Figura 18 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Crear.....	84
Figura 19 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Modificar.	85
Figura 20 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Eliminar.....	85
Figura 21 Diagrama de clases persistentes.	86
Figura 22 Modelo Entidad Relación.	87
Figura 23 Expansión del módulo Presentación.	91
Figura 24 Concentración hacia el módulo Presentación.....	91
Figura 25 Relaciones de uso de las clases.	93
Figura 26 Número de clases por categorías.....	96

Índice

Agradecimientos	I
Dedicatoria.....	II
Resumen	IV
Índice de Figuras	V
Índice	VI
Introducción	1
Capítulo 1. Fundamentación teórica.....	5
1.1 Introducción	5
1.2 Evolución del análisis y diseño del software. Estado Actual.....	5
1.3 Análisis y diseño en las diferentes metodologías	6
1.3.1 Rational Unified Process (RUP)	6
1.3.2 Extreme Programming (XP).....	10
1.3.3 Microsoft Solution Framework (MSF).....	11
1.4 Ingeniería de requisitos.....	13
1.5 Patrones de casos de uso y patrones de diseño	18
1.5.1 Patrones de casos de uso	19
1.5.2 Patrones de diseño	20
1.6 Métricas para el diseño	22
1.7 Lenguaje de modelado para el análisis y el diseño	28
1.8 Lenguajes de programación web	30
1.9 Herramientas CASE para el análisis y el diseño	32
1.9.1 Enterprise Architect	32
1.9.2 Rational Rose	33
1.9.3 Visual Paradigm.....	34
1.10 Conclusiones.....	35
Capítulo 2. Solución propuesta. Análisis y diseño del sistema	37
2.1 Introducción	37
2.2 Requerimientos de software	37
2.2.1 Requerimientos funcionales.....	38

2.2.2	Requerimientos no funcionales.....	49
2.3	Actores del sistema.....	51
2.4	Casos de usos.....	53
2.5	Diagrama de casos de uso del sistema.....	57
2.6	Descripción de casos de uso del sistema	58
2.7	Subsistema de diseño.....	77
2.8	Paquetes de diseño	78
2.9	Diagrama de clases de diseño.....	79
2.10	Diagramas de secuencia.....	81
2.11	Diagrama de clases persistentes.....	85
2.12	Modelo de datos.....	86
2.13	Conclusiones.....	88
Capítulo 3.	Análisis de los resultados.....	89
3.1	Introducción.....	89
3.2	Validación del modelo del sistema	89
3.3	Métricas de diseño a nivel de componentes.....	90
3.3.1	Métrica de acoplamiento.....	90
3.3.2	Métrica de cohesión.....	92
3.4	Métricas orientadas a clases. Tamaño de clase (TC).....	95
3.5	Conclusiones	96
Conclusiones Generales		98
Recomendaciones		99
Bibliografía.....		100
Anexos.....		102
Glosario de términos.....		116

Introducción

Antecedentes y problemática existente

En la actualidad la República de Cuba y la República Bolivariana de Venezuela, se encuentran inmersas en un acuerdo de colaboración en los marcos del ALBA, que intensifica las relaciones diplomáticas de ambos países. Esto conlleva que anualmente se compendien en conjunto, en el marco de la Comisión Mixta Cuba-Venezuela, una cantidad determinada de proyectos, los cuales no son controlados de una manera eficiente.

El proceso de elaboración, concepción, contratación y seguimiento de los proyectos presentados, en la Comisión Mixta Cuba-Venezuela, se efectúan de forma manual mediante reuniones de coordinación, por vía telefónica o por correo electrónico entre las partes implicadas, lo cual dificulta la comunicación y manipulación de los detalles de los proyectos ejecutados.

El control de los proyectos del Convenio Integral de Cooperación Cuba-Venezuela (CCV) parte desde la concepción y presentación de estos. Transitan por las fases de financiamiento, contratación y en todo momento se debe conocer su estado.

El proceso de presentación es de suma importancia. En este se elaboran los proyectos, se trazan las metas y objetivos, se destinan los recursos que se invertirán en la ejecución de los mismos y se someten a un proceso de revisión transitando por niveles superiores. Finalmente se dejan listos para su ejecución una vez que se les asigna el financiamiento.

Surge la necesidad de diseñar un sistema informático que sea capaz de automatizar el proceso de presentación de proyectos del convenio Cuba–Venezuela. De aquí que el problema científico a solucionar quede definido de la siguiente forma:

Problema Científico Investigativo

¿Cómo transformar los procesos del negocio para el módulo Presentación del sistema CCV en elementos que puedan ser implementados?

Objeto de estudio

Proceso de desarrollo del software.

Campo de acción

Análisis y diseño del módulo Presentación del sistema CCV.

Objetivo

Desarrollar el análisis y diseño del módulo Presentación para el sistema CCV.

Hipótesis

Si se desarrollan adecuadamente los procesos del negocio del módulo Presentación del sistema CCV se propiciará una adecuada implementación del mismo.

Tareas de la investigación

1. Realizar un estudio del proceso de desarrollo de software, de las herramientas y técnicas de análisis y diseño de software más usado.
2. Identificar los requerimientos.
3. Especificar el sistema.
4. Diseñar elementos del sistema.
5. Validar especificación del sistema.
6. Validar el diseño.

Métodos y técnicas de investigación a utilizar**Métodos Científicos****Teóricos:**

- Histórico – lógico: Para el estudio del estado del arte de los sistemas de gestión, del proceso de desarrollo de software y las técnicas de diseño de software más usados.
- Modelación: Para la creación de modelos y diagramas que reflejen la lógica del módulo Presentación en su análisis y diseño.
- Hipotético – deductivo: Para la confección de la hipótesis.

Empíricos:

- Entrevista: Para interactuar con los clientes e identificar los requerimientos del sistema.

Resultados esperados

- Modelo del sistema.
- Modelo de diseño.

Estructura de la tesis

El trabajo está estructurado por tres capítulos, los cuales están conformados como se muestra a continuación:

Capítulo 1. Fundamentación teórica.

El capítulo trata aspectos teóricos y técnicos de análisis y diseño. Se muestra una síntesis de la evolución su estado actual. Se tratan aspectos concernientes a la ingeniería de requisitos y a las métricas para el diseño que aseguran la factibilidad y la calidad del modelo de sistema y el modelo de diseño. Se exponen los principales patrones existentes para el desarrollo del diseño, así como las metodologías, herramientas y lenguaje más usados en el desarrollo de software a nivel mundial. Finalmente se muestra cuales son las técnicas, y herramientas que se usarán en el desarrollo del análisis y diseño del módulo de Presentación del sistema CCV.

Capítulo 2. Solución propuesta. Análisis y Diseño del sistema.

El capítulo expone la solución que se propone para la realización del módulo Presentación del sistema CCV. Parte de la aplicación de técnicas de ingenierías de requisitos para obtener las funcionalidades y restricciones del sistema a partir de los procesos del negocio. Se especifican los artefactos pertinentes

para elaborar el modelo del sistema. Por último se analizan los casos de uso, sus especificaciones y requerimientos para modelar los artefactos imprescindibles para obtener un adecuado modelo de diseño.

Capítulo 3. Análisis de los resultados.

El capítulo muestra los procedimientos y métodos utilizados para medir la factibilidad de los artefactos obtenidos, tanto en el modelo de sistema como en el de diseño. Para la validación de los artefactos del modelo del sistema se muestran las actas legales de liberación a diferentes niveles, mientras que para la validación de los artefactos del modelo de diseño se aplican métricas de diseño a nivel de componentes.

1.1 Introducción

En el presente capítulo se tratan aspectos teóricos y técnicos necesarios para llevar a cabo un adecuado análisis y diseño. Se mostrará una síntesis de la evolución y estado actual del análisis y el diseño. Se tratan aspectos concernientes a la ingeniería de requisitos y a las métricas para el diseño que aseguran la factibilidad y la calidad del modelo de sistema y el modelo de diseño y la concepción de los casos de uso (CU). Se exponen los principales patrones existentes para el desarrollo del diseño, así como las metodologías, herramientas y lenguaje más usados en el desarrollo de software a nivel mundial. Finalmente se muestra cuales son las técnicas, y herramientas que se usarán en el desarrollo del análisis y diseño del módulo Presentación del sistema CCV.

1.2 Evolución del análisis y diseño del software. Estado Actual

El término de análisis de software en su principio con un enfoque estructurado surge como complemento a otra problemática, el diseño estructurado. A finales de la década de los 60 y principio de los 70 aparecen las primeras publicaciones sobre este tema, exponiéndose la necesidad de hacer un análisis y diseño previo a la implementación de un software. El término de análisis estructurado fue ideado por Douglas Ross, pero muchos otros expertos en el tema hicieron importantes contribuciones al perfeccionamiento del mismo. En la década de los 80, Ward y Mellor y más tarde Hatley y Pirbhai introdujeron las aplicaciones en tiempo real, con las cuales se consiguió un método de análisis más robusto para resolver problemas informáticos. A finales de la década de los 80 con el auge de los mecanismos orientado a objetos, surgen un gran número de técnicas de modelado de análisis y diseño, pero esta vez con este nuevo enfoque. No es hasta que Grady Booch, James Rumbaugh e Ivar Jacobson, comenzaron a colaborar en fusionar las mejores técnicas de cada uno de sus métodos de análisis y diseño orientados a objetos, dando paso a lo que se conoce hoy en día como Lenguaje Unificado de Modelado (UML) del inglés Unified Modeling Language, el cual se ha convertido en el método más usado en los últimos años (Pressman, 2005). En la actualidad gracias al uso del UML junto a herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) de desarrollo que facilita el modelado del análisis y diseño

en el desarrollo de software aplicando metodologías de desarrollo de software como es el caso de RUP, XP u otras de las existentes, se ha logrado un desarrollo de las técnicas de análisis y diseño y con ello un mayor perfeccionamiento en el proceso de desarrollo de software. En el epígrafe que le continúa (1.3), se abordará con más detalles el enfoque del análisis y diseño en algunas de las metodologías más usadas en la actualidad en el desarrollo de software.

1.3 Análisis y diseño en las diferentes metodologías

El desarrollo de un software debe regirse por una guía que indique en cada momento de su ciclo de vida quien debe hacer qué, cómo y cuándo. El análisis y diseño del software en construcción es de suma importancia y es desarrollado con diferentes enfoques según la metodología de desarrollo de software usada. En este epígrafe se analizarán los principales elementos que exponen en el tema del análisis y diseño, algunas de las metodologías de desarrollo de software más usadas del mundo, como son: Rational Unified Process (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF).

1.3.1 Rational Unified Process (RUP)

RUP es una metodología de desarrollo de software que da la guía para transformar los requisitos que se obtienen una vez definida la necesidad de automatización y mejoras que requiere y necesita el cliente, en casos de uso que posteriormente se convertirán en conjunto de funcionalidades que le dan estructura al sistema desarrollado y que trae resultados visibles al usuario o cliente. Está definido en función de flujos de trabajos y fases (**Figura 1**), las cuales llevan el software desde su concepción hasta su entrega al usuario final.

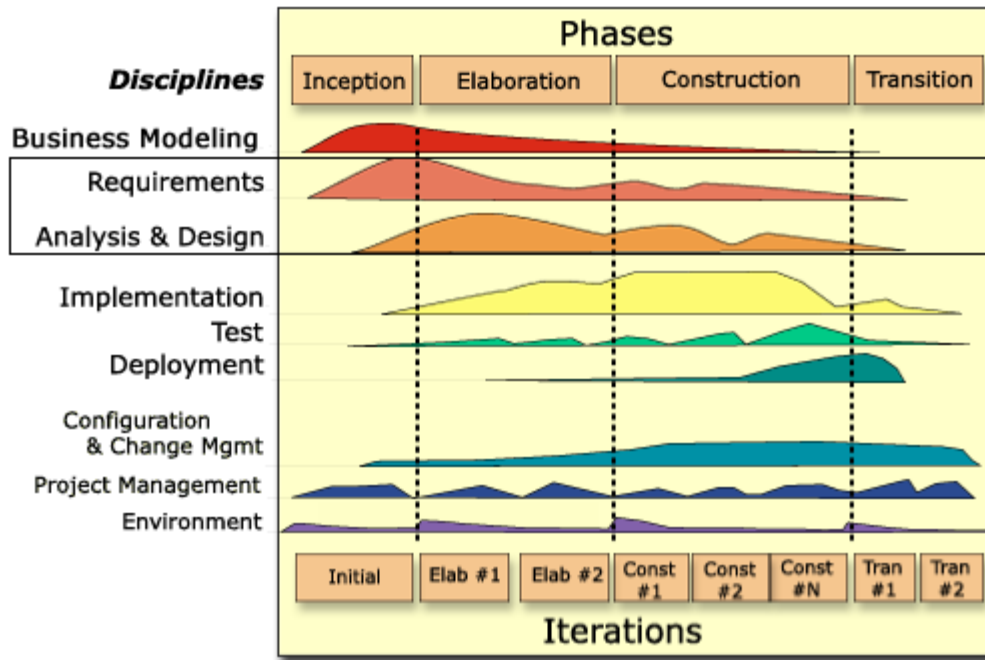


Figura 1 Proceso unificado de desarrollo de software (fases y flujos de trabajos).

Para RUP el análisis y diseño se concreta a través de sus flujos de trabajos. Parte desde el modelado del negocio, donde se modelan los procesos del negocio, luego comienza la etapa de requerimientos, donde se identifican las funcionalidades que se automatizaran del negocio definido, luego inicia el análisis y diseño el cual tiene como entrada el modelo del sistema definido en el flujo de trabajo anterior. Define que en primer lugar se debe determinar una arquitectura candidata, luego a partir del modelo del sistema hacer el análisis de los casos de uso determinados y por último se deben diseñar los componentes pertinentes.

A continuación se listaran los trabajadores que participan en el flujo de trabajo de requerimientos y en el de análisis y diseño, con los artefactos que genera cada uno de ellos, así como las actividades que guían el desarrollo de estos a lo largo del flujo de trabajo:

Tabla 1 Flujo de trabajo de requerimientos.

Trabajador	Artefactos	Actividades
------------	------------	-------------

Analista de sistema	<ul style="list-style-type: none"> • Plan de administración de requerimientos. • Glosario. • Visión. • Especificaciones suplementarias. • Modelos de casos de uso. 	<ul style="list-style-type: none"> • Capturar un vocabulario común. • Encontrar actores y casos de uso. • Desarrollo del plan de administración de requerimientos. • Administrar las dependencias. • Elicitar requerimientos. • Estructurar el modelo de casos de uso.
Arquitecto de software		<ul style="list-style-type: none"> • Priorizar casos de uso.
Especificador de requerimientos	<ul style="list-style-type: none"> • Casos de uso. • Especificación de requerimientos de software. • Paquetes de casos de uso. • Requerimientos de software. 	<ul style="list-style-type: none"> • Detallar los casos de uso. • Detallar los requerimientos de software.
Revisor técnico		<ul style="list-style-type: none"> • Revisar los requerimientos.

Este flujo de trabajo tiene como principales objetivos encontrar y especificar los requerimientos del sistema, así como agruparlos en casos de usos, encontrar los actores y conformar el modelo del sistema (Rational Software, Corporation, 2003).

Tabla 2 Flujo de trabajo de análisis y diseño.

Trabajador	Artefactos	Actividades
Arquitecto de software	<ul style="list-style-type: none"> • Documento de arquitectura de software. • Modelo de análisis. • Modelo de diseño. • Modelo de despliegue. 	<ul style="list-style-type: none"> • Análisis de la arquitectura. • Identificación de mecanismos y elementos de diseño. • Incorporar elementos de diseño existentes. • Describir la arquitectura.
Diseñador	<ul style="list-style-type: none"> • Análisis de clases. • Diseño de clases. • Clases de pruebas. • Realización de casos de uso. • Subsistemas de diseño. • Paquetes de diseño. 	<ul style="list-style-type: none"> • Analizar casos de uso. • Diseñar casos de uso. • Diseñar subsistemas • Diseñar clases. • Diseñar elementos de pruebas.
Diseñador de Base de datos	<ul style="list-style-type: none"> • Modelo de datos 	<ul style="list-style-type: none"> • Diseñar base de datos.
Diseñador de pruebas	<ul style="list-style-type: none"> • Diseño de pruebas 	<ul style="list-style-type: none"> • Diseñar las pruebas.
Diseñador de interfaz de usuario	<ul style="list-style-type: none"> • Prototipos de interfaz de usuario. • Mapa de navegación 	<ul style="list-style-type: none"> • Diseñar Prototipos de interfaz de usuario. • Diseñar mapa de navegación.

Este flujo de trabajo tiene como principales objetivos transformar los requerimientos en un diseño objetivo del sistema, evolucionar hasta una arquitectura robusta y lograr la adaptación del sistema con el desarrollo de la posterior implementación (Rational Software, Corporation, 2003).

1.3.2 Extreme Programming (XP)

Programación extrema del inglés Extreme Programming (XP) es una de las conocidas metodologías de desarrollo de software ágiles. El desarrollo con XP se efectúa de forma incremental, sus principios fundamentales se pueden resumir en los siguientes:

- Comunicación: Comunicación total a todos los niveles de trabajo. Se trabaja en grupos de dos personas por ordenador pero con total comunicación en todos los momentos entre todos los grupos.
- Usuario: El usuario siempre está en mente. Se han de satisfacer sus necesidades pero nada más.
- Simplicidad: Lo más simple es lo mejor, funciona mejor, más rápido, es más adaptable, más barato y además es más entendible.
- YAGNI: “You aren’t gonna need it” (No lo vas a necesitar). No hagas nada que creas que en el futuro puede ser útil porque probablemente no lo vas a necesitar. Es una pérdida de tiempo.
- OAOO: “Once and only once” (Una única vez). Las cosas se hacen una sola vez.

Estos principios se ven complementados en lo que llaman “sus 12 buenas prácticas”: Planificación, Versiones Pequeñas, Sistema Metafórico. (Metaphor), Diseños simples, Testeos Continuos, Refactoring, Programación en parejas, Propiedad colectiva del código, Integración continua, 40 horas por semana, El cliente en su sitio, Estándares de codificación.

XP plantea que el análisis y diseño no se debe ausentar en un ciclo de desarrollo de software. El análisis se efectúa de forma muy ligera, mientras que el diseño (**Figura 2**), si se lleva a profundidad como en la mayoría de las metodologías. El análisis se transforma en la exposición por parte del cliente de las “user-stories”, para lograr un entendimiento de lo que se quiere. Luego continua el diseño global del sistema, en el que se profundiza hasta el nivel necesario para que los desarrolladores sepan exactamente que van hacer (Cortizo Pérez, et al., 2004).

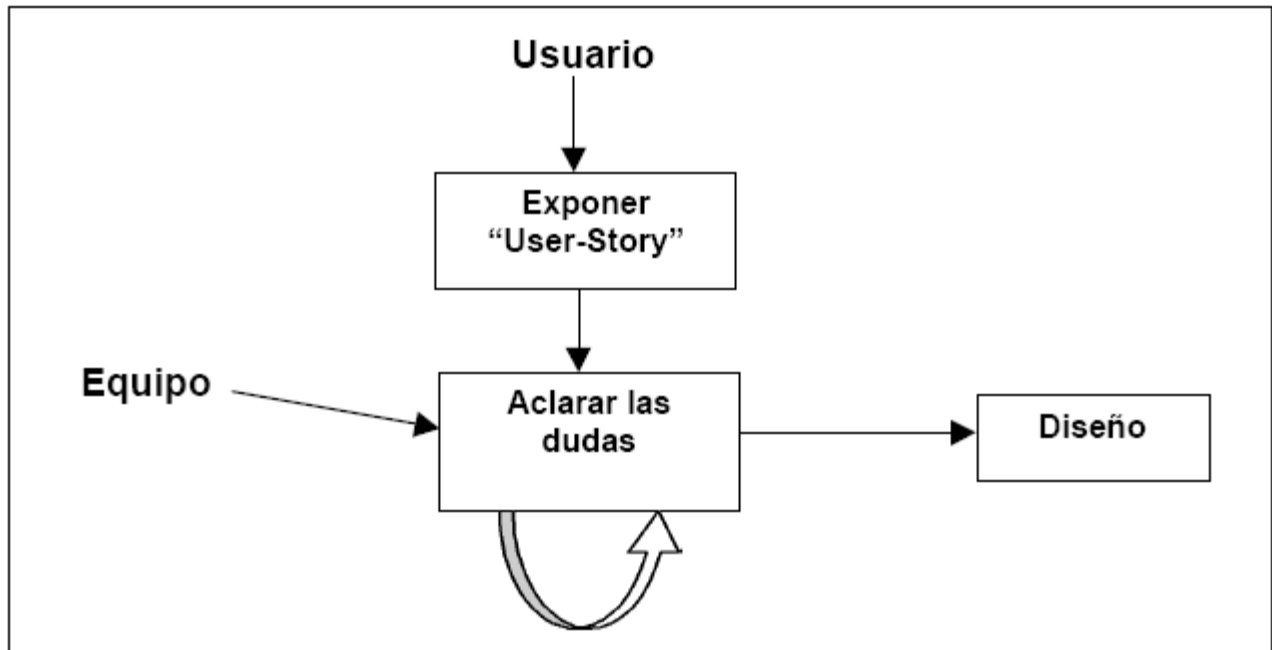


Figura 2 Actividades del diseño en XP.

1.3.3 Microsoft Solution Framework (MSF)

MSF es una metodología de desarrollo de software muy flexible. Se centra en el modelo de cascada, usando puntos de control para el paso entre sus fases y en el modelo espiral para refinar continuamente los requerimientos y las estimaciones del proyecto. Además se compone de varios modelos, encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.

El proceso de desarrollo en MSF consta de fases: Visión, Planeación, Desarrollo, Estabilización e Implantación; las cuales junto a los modelos mencionados anteriormente, complementan el ciclo de desarrollo de software en esta metodología. (Microsoft)



Figura 3 Fases de MSF.

MSF enfoca el análisis y diseño a través del Modelo de Diseño del Proceso en la fase de planeación. El mismo está diseñado para distinguir entre los objetivos empresariales y las necesidades del cliente. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores (Universidad Simon Bolivar, 1999).

Diseño Conceptual: Establece los conceptos que especifican las necesidades de los usuarios. Se debe comenzar a estructurar la solución propuesta, para ello propone la elaboración de los siguientes artefactos:

- Perfil de los usuarios: Especifica la ubicación, las capacidades y las expectativas, de los usuarios.
- Escenarios de los usuarios: Describen qué sucede en la ejecución de una tarea en particular; especifican cómo son los procesos, las funciones y los procedimientos.

Diseño Lógico: Estructura y organiza los elementos de la solución propuesta, para ello propone la elaboración de los siguientes artefactos:

- Diseño de la Interfaz de Usuario: Presenta los elementos y lineamientos que conforman el diseño de la interfaz de usuario.

- Componentes de la Solución: Establece los elementos involucrados en la solución, así como sus relaciones.
- Bases de Datos Lógica: Especificación lógica de las Bases de Datos que conforman o con las que interactúa la solución.

Diseño Físico: En esta se aplican las restricciones tecnológicas a la solución del diseño lógico, para ello propone la elaboración de los siguientes artefactos:

- Restricciones de Tecnología: Especifica la tecnología utilizada.
- Implementación de la Interfaz del Usuario: Muestra la apariencia de la solución.
- Arquitectura de la solución: Presenta la vista de implantación de la solución.

1.4 Ingeniería de requisitos

La ingeniería de requisitos (IR), como disciplina iniciadora del proceso de desarrollo de software, tiene como principal objetivo garantizar la especificación de un sistema que cumpla con las expectativas y necesidades del cliente. En esta se identifican dos aspectos fundamentales, el primero, cual es el propósito del sistema que se va a desarrollar y el segundo, el contexto en el que será usado.

La IR brinda la guía para lograr un enlace entre las necesidades reales de los clientes, usuarios y otros involucrados al sistema. Consiste en un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un medio que necesite de un software y convertir la declaración de estas en una descripción completa, precisa y documentada de los **requerimientos** del sistema.

Pero, ¿qué es un requerimiento?

El Institute of Electrical and Electronics Engineers (IEEE) define requerimiento como una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Una representación documentada de una condición o capacidad de un sistema (IEEE Standards Association, 2004).

La IR da la vía para determinar los requerimientos, analizarlos para lograr una solución viable, especificarlos, validarlos y gestionarlos. (Pressman, 2005).

Como toda disciplina de ingeniería, esta se divide en etapas con el fin de hacer un análisis minucioso de los requerimientos del sistema. Varios son los autores que han incursionado en la IR, los cuales tienen coincidencias y desavenencias en la forma de desarrollar esta vital fase del proceso de desarrollo del software. A nivel internacional cuando se habla de IR existe un entendimiento común en cuanto a la Identificación, Análisis y Negociación, Especificación, Validación y Gestión de Requisitos, como principales etapas de la IR, las cuales se especifican a continuación por la importancia que se le concede para el desarrollo del sistema CCV.

Etapas de la ingeniería de requisitos.

La IR se divide en etapas lógicamente bien definidas, las cuales especifican los pasos, las tareas y las técnicas que se deben emplear para interactuar con los clientes y especificar correctamente el sistema. El buen entendimiento de cada una de las etapas de la IR y su correcta aplicación durante la primera fase del desarrollo del software, propicia una vía efectiva para mantener una fluida comunicación entre los involucrados en el desarrollo del proyecto y una guía para especificar en detalles el sistema solicitado.

Elicitación/Identificación de requisitos

Es la etapa inicial, en la que se debe investigar con los clientes o involucrados, como los sistemas o productos se ajustan a las necesidades del negocio y como el sistema o producto será usado. Para alcanzar un entendimiento efectivo con el cliente. Para su desarrollo se recomiendan las siguientes tareas (Durán Toro, et al., 2000):

- Obtener información sobre el dominio del problema y el sistema actual.
- Preparar y realizar las reuniones de elicitación/negociación.
- Identificar/revisar los objetivos del sistema.
- Identificar/revisar los requisitos de almacenamiento de información.
- Identificar/revisar los requisitos funcionales.
- Identificar/revisar los requisitos no funcionales.
- Priorizar objetivos y requisitos.

Para obtener los resultados esperados por la aplicación de las tareas expuestas anteriormente se hace necesaria la aplicación de las siguientes técnicas:

Entrevistas:

Es la técnica de elicitación más utilizada, consta de tres fases, preparación, realización y análisis. Para realizar una entrevista con resultados adecuados se debe estudiar el dominio del problema con anterioridad, se deben seleccionar las personas que se van a entrevistar, tener con claridad el objetivo de y contenido de la misma y planificarlas; todo esto se elabora en la fase de preparación. Durante la realización de la entrevista no debe faltar: la apertura, el desarrollo con el uso de un lenguaje natural y preguntas abiertas y por ultimo una conclusión, en la que incluye una recapitulación para verificar que no hubo mal entendidos en el desarrollo de la entrevista. Luego se debe proceder hacer un análisis detallado de la entrevista realizada, documentando y organizando las notas que se adquirieron en su desarrollo, para que sea verificada luego por el entrevistado (Durán Toro, et al., 2000).

Desarrollo conjunto de aplicaciones JAD (Joint Application Development):

Es una alternativa a las entrevistas individuales, se desarrolla en reuniones en grupos en períodos de corta duración. Consta de cuatro principios fundamentales, dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que se ve es lo que se obtiene). Su ventaja sobre las entrevistas tradicionales radica en que:

- Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- Todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada, no sólo los ingenieros de requisitos.
- Implica más a los clientes y usuarios en el desarrollo.

Consta de 3 fases: adaptación, celebración de las sesiones JAD, conclusión (Durán Toro, et al., 2000).

Tormenta de ideas (Brainstorming)

Es una técnica de reuniones en grupo que tiene como principal objetivo generar ideas en un ambiente libre de críticas o juicios. Requiere poca organización, generalmente estructurado por menos de diez participantes incluido un jefe, encargado de iniciar el debate. Ayuda a generar diferentes vistas de un problema y formular diferentes soluciones. Se estructura de cuatro fases: preparación, generación, consolidación y documentación (Durán Toro, et al., 2000).

Casos de uso

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Jacobson. Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables (Durán Toro, et al., 2000).

Análisis de requisitos

Luego de obtenidos los requisitos se procede hacer un análisis de los mismos. Durante esta etapa se agrupan por categorías y se organizan en subconjuntos, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad, y se clasifican en base a las necesidades de los clientes. Para realizar un adecuado análisis de los requisitos Pressman plantea las siguientes preguntas (Pressman, 2005):

- ¿Cada requisito es consistente con objetivos generales del sistema?
- ¿Tienen todos los requisitos especificados el nivel adecuado de abstracción?
- ¿El requisito es necesario o representa una característica añadida que puede no ser esencial a la finalidad del sistema?
- ¿Cada requisito está delimitado y sin ambigüedad?
- ¿Cada requisito tiene procedencia? Es decir, existe un origen (general o específico) conocido para cada requisito.
- ¿Existen requisitos incompatibles con otros requisitos?
- ¿Es posible lograr cada requisito en el entorno técnico donde se integrará el sistema o producto?
- ¿Se puede probar el requisito una vez implementado?

La meta fundamental de esta etapa está en solucionar cualquier contradicción o ambigüedad que existan en los requisitos expuestos por los clientes, con el objetivo de lograr requisitos concretos que cumplan con sus necesidades.

Especificación de requisitos

Esta etapa tiene como principal objetivo la descripción detallada de los requisitos a través de un documento, o cualquier medio gráfico o matemático que indique como debe ser el comportamiento del sistema a desarrollar, cuáles son sus particularidades y funciones requeridas. El resultado final de la especificación de requisitos debe ser la especificación del sistema, la cual describe la información que fluye por el sistema, y delimita cada elemento y característica de este (Pressman, 2005).

Validación de requisitos

La validación de requisitos es la encargada de medir la calidad de los requisitos especificados. "La validación de requisitos examina las especificaciones para asegurar que todos requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a estándares establecidos para el proceso, el proyecto y el producto" (Pressman, 2005). Existen varias técnicas para la validación de requisitos, como son: las revisiones técnicas formales, el prototipado y otras. Para determinar posibles errores en los requisitos descritos Pressman plantea las siguientes preguntas (Pressman, 2005):

- ¿Está el requisito claramente definido? ¿Pueden interpretarse mal?
- ¿Está identificado el origen del requisito (por ejemplo: persona, norma, documento)? ¿El planteamiento final del requisito ha sido contrastado con la fuente original?
- ¿El requisito está delimitado en términos cuantitativos?
- ¿Que otros requisitos hacen referencia al requisito estudiado? ¿Están claramente identificados por medio de una matriz de referencias cruzadas o por cualquier otro mecanismo?
- ¿El requisito incumple alguna restricción definida?
- ¿El requisito es verificable? Si es así, ¿se puede efectuar pruebas (algunas veces llamadas criterios de validación) para verificar el requisito?
- ¿Se puede seguir el requisito en el modelo del sistema que se ha desarrollado?

- ¿Se puede localizar el requisito en el conjunto de objetivos del sistema/producto?
- ¿Está el requisito asociado con rendimientos del sistema o con su comportamiento y han sido establecidas claramente sus características operacionales? ¿El requisito está implícitamente definido?

Comprobar la fiabilidad, flexibilidad y calidad de los requisitos, asegura que se inicie la elaboración de un sistema con los objetivos bien definidos, que se sepa que es lo que se tiene que hacer para satisfacer las necesidades del cliente.

Gestión de requisitos

Los cambios de los requerimientos durante el desarrollo del sistema son inevitables, por tal razón se deben someter a un proceso de gestión. “La gestión de requerimientos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento” (Pressman, 2005).

Para la gestión de requisitos se procede primeramente a identificar los requerimientos y luego se lleva el control de estos a través de matrices de trazabilidad basadas en varios conceptos, dentro de estas se tiene:

- Matriz de seguimiento de características.
- Matriz de seguimiento de orígenes.
- Matriz de seguimiento de dependencias.
- Matriz de seguimiento de subsistemas.
- Matriz de seguimiento de interfaces.

El control de los cambios de los requisitos es vital para el buen desarrollo del sistema, por lo que es recomendable además de las actividades expuestas anteriormente para la gestión de los requisitos, tener en cuenta otras de las actividades pertenecientes a la Gestión de Configuración del Software (GCS), que son de gran utilidad para la gestión de los requisitos (Pressman, 2005).

1.5 Patrones de casos de uso y patrones de diseño

En la actualidad tanto los patrones de casos de uso como los patrones de diseño son muy utilizados en el desarrollo de un software. El uso de técnicas y soluciones ante diferentes problemáticas durante el desarrollo del software garantizan una reducción considerable de los errores a cometer. La utilización de

patrones durante el análisis y el diseño se considera una buena práctica, por tal razón a continuación se abordan especificaciones de los principales patrones que se deben usar para el desarrollo del análisis y el diseño.

1.5.1 Patrones de casos de uso

La experiencia en la utilización de casos de uso en los proyectos de los sistemas más diversos ha evolucionado en un conjunto de patrones a través de los años. Dado un problema a resolver éstas técnicas han demostrado ser la solución preferida por la comunidad del desarrollo de software. Se presentan a modo de herramientas que permiten resolver los problemas que se les planteen a los asistentes de una forma ágil y sistemática.

A continuación una breve descripción de los patrones de casos de uso más usados:

- **CRUD: Completo (CRUD: Complete)**

Es un patrón de estructura, sus siglas significan *crear, leer, actualizar y eliminar* del inglés *Create-Read-Update-Delete*. Este consiste en identificar un caso de uso, llamado “Administrar Información”, que modele las diferentes operaciones que pueden ser realizadas en una misma identidad, tales operaciones serían las que indican sus siglas. Este patrón debe ser usado cuando todos los flujos contribuyan a un mismo valor del negocio y que sean cortos y sencillos.

- **Extensión concreta (Concrete Extension or Inclusion: Extension)**

Es un patrón de estructura y consiste en dos casos de uso y una relación de extensión relacionada entre ellos. El caso de uso extendido es concreto lo cual quiere decir que puede ser instanciado por sí mismo como también extender el caso de uso base. Este último puede ser concreto o abstracto. El patrón es aplicable cuando el flujo de uno puede extender el flujo de otro caso de uso como también ser ejecutado por sí mismo.

- **Inclusión concreta (Concrete Extension or Inclusion: Inclusion)**

Es un patrón de estructura, en este hay una relación de inclusión entre un caso de uso base al caso de uso incluido. Este último puede ser instanciado por sí mismo y el caso de uso base puede ser abstracto o concreto. Este patrón es utilizado cuando el flujo de datos de un caso de uso puede ser incluido en el flujo de otro caso de uso y ejecutarse por sí solo.

- **Reglas de negocio (Business Rules: Static Definition)**

Es un patrón de estructura. Es aplicado en casos de uso que modelan servicios que son afectados por las reglas de negocios definidas en la organización. Las reglas son descritas en un documento por separado. Su uso es apropiado cuando no hay necesidad de cambios dinámicos en las reglas de negocio. (Övergaard, et al., 2004)

1.5.2 Patrones de diseño

Por otra parte los patrones de diseño se adentran más en lo específico, o sea estos son utilizados en componentes y en clases individuales (MITOPENSOURCEWARE Universia, 2001).

Un patrón de diseño es:

- una solución estándar para un problema común de programación
- una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios
- un proyecto o estructura de implementación que logra una finalidad determinada
- un lenguaje de programación de alto nivel
- una manera más práctica de describir ciertos aspectos de la organización de un programa
- conexiones entre componentes de programas
- la forma de un diagrama de objeto o de un modelo de objeto

Las cualidades de un patrón de diseño están definidas por su *encapsulamiento* y *abstracción* pues cada patrón encapsula un problema bien definido y su solución en un dominio particular. La *extensión* y *variabilidad* también forman parte de las cualidades de un patrón ya que cada uno debería ser abierto por extensión o parametrización por otros patrones, de tal forma que puedan aplicarse juntos para solucionar un problema. Cada patrón una vez aplicado genera un contexto resultante, el cual concuerda con el contexto inicial de uno o más de uno de los patrones del catálogo siendo así la *generatividad* y *composición* otra de sus cualidades. Finalmente el *equilibrio* permite que cada patrón deba realizar algún tipo de balance entre sus efectos y restricciones (Larman, 1999) .

Los patrones se clasifican según su propósito en:

Patrones de Creación: Tratan la creación de instancias o sobre qué objetos un objeto delegará responsabilidades.

- **Abstract Factory:** Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta.
- **Builder:** Permite a un objeto construir un objeto complejo especificando sólo su tipo y contenido.
- **Factory Method:** Define una interfaz para crear un objeto dejando a las subclases decidir el tipo específico al que pertenecen.
- **Prototype:** Permite a un objeto crear objetos personalizados sin conocer su clase exacta a los detalles de cómo crearlos.
- **Singleton:** Garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a él.

Patrones Estructurales: Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad, describiendo así la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

- **Adapter:** Convierte la interfaz que ofrece una clase en otra esperada por los clientes.
- **Bridge:** Desacopla una abstracción de su implementación y les permite variar independientemente.
- **Composite:** Permite gestionar objetos complejos e individuales de forma uniforme.
- **Decorator:** Extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes.
- **Facade:** Simplifica los accesos a un conjunto de objetos relacionados proporcionando un objeto de comunicación.
- **Flyweight:** Usa la compartición para dar soporte a un gran número de objetos de grano fino de forma eficiente.
- **Proxy:** Proporciona un objeto con el que se controla el acceso a otro objeto.

Patrones de Comportamiento: Tratan la interacción y cooperación entre clases. Organizan, manejan y combinan comportamientos.

- **Chain of Responsibility:** Evita el acoplamiento entre quien envía una petición y el receptor de la misma.
- **Command:** Encapsula una petición de un comando como un objeto.

- Interpreter: Dado un lenguaje define una representación para su gramática y permite interpretar sus sentencias.
- Iterator: Acceso secuencial a los elementos de una colección.
- Mediator: Define una comunicación simplificada entre clases.
- Memento: Captura y restaura un estado interno de un objeto.
- Observer: Una forma de notificar cambios a diferentes clases dependientes.
- State: Modifica el comportamiento de un objeto cuando su estado interno cambia.
- Strategy: Define una familia de algoritmos, encapsula cada uno y los hace intercambiables.
- Template Method: define un esqueleto de algoritmo y delega partes concretas de un algoritmo a las subclases.
- Visitor: Representa una operación que será realizada sobre los elementos de una estructura de objetos, permitiendo definir nuevas operaciones sin cambiar las clases de los elementos sobre los que opera (Wesley, 2003).

Es aconsejable utilizar patrones de diseño desde un principio si hay una comprensión total del dominio y el problema, así como lo es utilizar un algoritmo eficiente desde el comienzo. Los patrones de diseño pueden incrementar o disminuir la capacidad de comprensión de un diseño o de una implementación, disminuirla al añadir accesos indirectos o aumentar la cantidad de código, disminuirla al regular la modularidad, separar mejor los conceptos y simplificar la descripción. Una vez que aprenda el vocabulario de los patrones de diseño le será más fácil y más rápido comunicarse con otros individuos que también lo conozcan.

Los patrones en general tienen como principal objetivo transmitir experiencia, es de suma importancia que estos estén al alcance de los desarrolladores de software.

1.6 Métricas para el diseño

La medición es esencial para cualquier disciplina de ingeniería y la ingeniería de software no es una excepción. Una métrica puede ser cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software. Éstas ayudan a la evaluación de los modelos de análisis y diseño, proporcionan una indicación de la complejidad de los diseños procedimentales y del

código fuente, y ayudan a realizar pruebas más efectivas. En las métricas para el modelo de diseño se genera la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del Sistema de Información. Proporcionan al diseñador una mejor visión interna y ayudan a que el diseño evolucione a un nivel superior de calidad. Se dividen en métricas de diseño arquitectónico, métricas de diseño a nivel de componente y métricas orientadas a clases.

Métricas de diseño arquitectónico

Éstas se concentran en las características de la estructura del programa dándole énfasis a la estructura arquitectónica y en la eficiencia de los módulos. Éstas métricas son llamadas de caja negra, ya que no requieren ningún conocimiento del trabajo interno de ningún modo en particular del sistema.

Card y Glass proponen tres medidas de complejidad del software:

- Complejidad estructural. $S(i)$, de un módulo i se define de la siguiente manera:

$$S(i) = f_{out}^2(i)$$

donde $f_{out}(i)$ es la expansión del módulo i .

- Complejidad de datos. $D(i)$ proporciona una indicación de la complejidad en la interfaz interna de un módulo i y se define como:

$$D(i) = v(i) / [f_{out}(i) + 1]$$

donde $v(i)$ es el número de variables de entrada y salida del módulo i .

- Complejidad de sistema. $C(i)$, se define como la suma de las complejidades estructural y de datos, y se define como:

$$C(i) = S(i) + D(i)$$

A medida que crecen los valores de complejidad, la complejidad arquitectónica o global del sistema también aumenta (Pressman, 2005).

Por otra parte Henry y Kafura proponen la métrica de complejidad de expansión-concentración, que considera las estructuras de datos que recoge (concentra) o actualizan (expansión). También es representada una fórmula para definir esta métrica.

Fenton sugiere varias métricas de morfología simples las cuales permiten comparar diferentes arquitecturas de programa mediante un conjunto de dimensiones directas (Pressman, 2005).

Métricas de diseño a nivel de componentes

Éstas se concentran en las características internas de los componentes del software, de ahí que son llamadas métricas de caja blanca, e incluyen medidas de la cohesión, acoplamiento y complejidad del módulo. Estas medidas ayudan al desarrollador de software a juzgar la calidad de un diseño a nivel de componentes. Puede aplicarse una vez que se haya desarrollado un diseño procedimental o pueden retrasarse hasta tener disponible el código fuente (Pressman, 2005).

Métricas de cohesión

Bieman y Ott definen una colección de métricas que se definen con cinco conceptos y medidas:

- Porción de datos. Dicho simplemente, una porción de datos es una marcha atrás a través de un módulo que busca valores de datos que afectan a la localización del módulo en el que empezó la marcha atrás. Debería resaltarse que se pueden definir tanto porciones de programas (que se centran en enunciados y condiciones) como porciones de datos.
- Símbolos léxicos (tokens) de datos. Las variables definidas para un módulo pueden definirse como señales de datos para el módulo.
- Señales de unión. El conjunto de señales de datos que se encuentran en uno o más porciones de datos.
- Señales de super-unión. Las señales de datos comunes a todas las porciones de datos de un módulo.
- Cohesión. La cohesión relativa de una señal de unión es directamente proporcional al número de porciones de datos que liga (Pressman, 2005).

Bieman y Ott desarrollaron métricas para las cohesiones funcionales fuertes, para las cohesiones funcionales débiles y pegajosidad. Éstas pueden ser interpretadas de la siguiente manera:

“Todas estas métricas de cohesión tienen valores que van desde 0 a 1. Tienen un valor de 0 cuando un procedimiento tiene más de una salida y no muestra ningún atributo de cohesión indicado por una métrica particular. Un procedimiento sin señales de super-uniión, sin señales comunes a todas las porciones de datos, no tiene una cohesión funcional fuerte (no hay señales de datos que contribuyan a todas las salidas). Un procedimiento sin señales de unión, es decir, sin señales comunes a más de una porción de datos (en procedimientos con más de una porción de datos), no muestra una cohesión funcional débil y ninguna adhesividad (no hay señales de datos que contribuyan a más de una salida). La cohesión funcional fuerte y la pegajosidad se obtienen cuando las métricas de Bieman y Ott toman un valor máximo de 1.” (UDELAB, 2009)

Para ilustrar el carácter de estas métricas, se debe la métrica para la cohesión funcional fuerte:

$$CFF(i) = SU(SA(i))/señales (i)$$

donde $SU(SA(i))$ denota señales de super-uniión (el conjunto de señales de datos que se encuentran en todas las porciones de datos de un módulo i). Como la relación de señales de super-uniión con respecto al número total de señales en un módulo i aumenta hasta un valor máximo de 1, la cohesión funcional del módulo también aumenta 1.

Métricas de acoplamiento

El acoplamiento de módulo proporciona una indicación de la “conectividad” de un módulo con otros módulos, datos globales y entorno exterior. Dhama ha propuesto una métrica para el acoplamiento del módulo que combina el acoplamiento de flujo de datos y de control: acoplamiento global y acoplamiento de entorno. Las medidas necesarias para calcular el acoplamiento de módulo se definen en términos de cada uno de los tres tipos de acoplamiento apuntados anteriormente.

Para el acoplamiento de flujo de datos y de control:

d_i = número de parámetros de datos de entrada

c_i = número de parámetros de control de entrada

d_o = número de parámetros de datos de salida

c_o = número de parámetros de control de salida

Para el acoplamiento global

g_d = número de variables globales usadas como datos

g_c = número de variables globales usadas como control

Para el acoplamiento de entorno:

w = número de módulos llamados (expansión)

r = número de módulos que llaman al módulo en cuestión (concentración)

Usando estas medidas, se define un indicador de acoplamiento de módulo, m_c de la siguiente manera:

$$m_c = k/M$$

donde $k = 1$ es una constante de proporcionalidad.

$$M = d_i + a * c_i + d_o + b * c_o + g_d + c * g_c + w + r$$

donde:

$$a=b=c=2$$

Usando estas medidas, se define un indicador de acoplamiento de módulo, y cuanto mayor es el valor de este indicador, menor es el acoplamiento del módulo (Pressman, 2005).

Métricas de complejidad

Se pueden calcular una variedad de métricas del software para determinar la complejidad del flujo de control del programa. Muchas de éstas se basan en una representación denominada grafo de flujo, un grafo es una representación compuesta de nodos y enlaces (también denominados filis). Cuando se dirigen los enlaces (aristas), el grafo de flujo es un grafo dirigido.

McCabe identifica un número importante de usos para las métricas de complejidad, donde pueden emplearse para predecir información sobre la fiabilidad y mantenimiento de sistemas software, también se alimentan la información durante el proyecto de software para ayudar a controlar la actividad de diseño, en las pruebas y mantenimiento, proporcionan información sobre los módulos del software para ayudar a resaltar las áreas de inestabilidad (Pressman, 2005).

“La métrica de McCabe proporciona una medida cuantitativa para probar la dificultad y una indicación de la fiabilidad última. Estudios experimentales indican una fuerte correlación entre la métrica de McCabe y el número de errores que existen en el código fuente, así como el tiempo requerido para encontrar y corregir dichos errores. McCabe también defiende que la complejidad ciclomática puede emplearse para proporcionar una indicación cuantitativa del tamaño máximo del módulo.”

Métricas de diseño de interfaz

“Sears sugiere la conveniencia de la representación como una valiosa métrica de diseño para interfaces hombre-máquina. Una IGU (Interfaz Gráfica de Usuario) típica usa entidades de representación, iconos gráficos, texto, menús, ventanas y otras para ayudar al usuario a completar tareas. Para realizar una tarea dada usando una IGU, el usuario debe moverse de una entidad de representación a otra. Las posiciones absolutas y relativas de cada entidad de representación, la frecuencia con que se utilizan y el “costo” de la transición de una entidad de representación a la siguiente contribuirán a la conveniencia de la interfaz.” (Pressman, 2005)

Para calcular la representación óptima de una IGU, la superficie de la interfaz (el área de la pantalla) se divide en una cuadrícula. Cada cuadro de la cuadrícula representa una posible posición de una entidad de la representación.

La conveniencia de la representación es empleada para la valoración de diferentes distribuciones propuestas de IGU y la sensibilidad de una representación en particular a los cambios en las descripciones de tareas (por ejemplo, cambios en la secuencia y/o frecuencia de transiciones). Es importante apuntar que el árbitro final debería ser la respuesta del usuario basada en prototipos de IGU. Nielsen Levy afirma; que puede haber una posibilidad de éxito si se prefiere la interfaz basándose exclusivamente en la opinión del usuario ya que el rendimiento medio de tareas de usuario y su satisfacción con la IGU están altamente relacionadas (Pressman, 2005) (UDELAB, 2009).

Métricas orientadas a clases

La clase es la unidad principal de cualquier sistema orientado a objeto. Esto dice, que tanto las medidas y métricas para una clase individual, la jerarquía de clases y las colaboraciones de las clases resultarán de gran utilidad al ingeniero que desee estimar la calidad de un diseño.

Tamaño de clase (TC)

El tamaño general de una clase se puede determinar siguiendo los planteamientos a continuación:

- El número total de operaciones (tanto operaciones heredadas como operaciones privadas de la instancia) que están encapsuladas dentro de la clase.
- El número de atributos (tanto atributos heredados como atributos privados de la instancia) que están encapsulados en la clase.

Si existen valores grandes de TC éstos estarán demostrando que una clase puede tener demasiada responsabilidad, lo cual reduciría la reutilización de la clase y hará complicada la implementación y la prueba. De forma contraria sucede si los valores TC son de menor valor.

Por otra parte es necesaria una evaluación concreta de las métricas mediante los umbrales. Algunos especialistas plantean la clasificación de la siguiente manera:

Tabla 3 Clasificación de las clases.

Clasificación	Valores de los umbrales
Pequeño	≤ 20
Medio	$>20 \leq 30$
Grande	>30

Finalmente se calcula los promedios correspondientes a los diferentes valores para tener una estimación general del sistema.

1.7 Lenguaje de modelado para el análisis y el diseño

El Lenguaje Unificado de Modelado (UML) es un sistema de notación que se ha estandarizado en el mundo dentro del proceso de desarrollo de software que permite a los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema, tener un entendimiento común del producto que se quiere obtener, entendimiento que es de vital importancia para que cualquier sistema se pueda desarrollar eficientemente. Creado por Grady Booch, James Rumbaugh e Ivar Jacobson, el UML visualiza, especifica, construye y documenta un sistema de software.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- ✓ Elementos: Los elementos son abstracciones de cosas reales o ficticias (objeto, acciones, etc.).
- ✓ Relaciones: Relacionan los elementos entre sí.
- ✓ Diagramas: Son colecciones de elementos con sus relaciones.

Los *elementos* pueden ser estructurales, de comportamiento, de agrupación o de anotación. Los estructurales están compuestos por actores, casos de uso, clases y objetos; los de comportamiento no son más que los mensajes y los de agrupación son los paquetes.

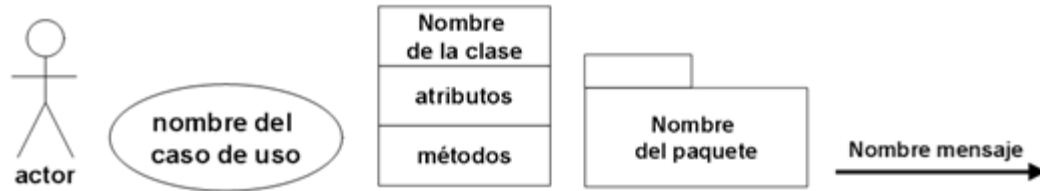


Figura 4 Elementos de UML.

Existen varios tipos de *relaciones*:

Dependencia: Es una relación semántica entre dos elementos, en la cual una modificación puede afectar a la semántica de otro elemento. Existen varios tipos de dependencia predefinidas que se especifican mediante estereotipos `<<extend>>` o `<<include>>` para casos de uso.

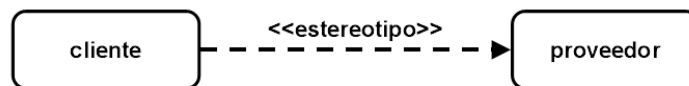


Figura 5 Relaciones de dependencia de UML.

Asociación: Es una relación estructural entre dos elementos, que describen las conexiones entre ellos (pueden ser bidireccional). Pueden ser de agregación o de composición.

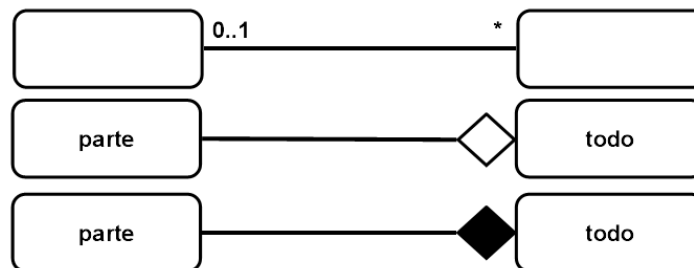


Figura 6 Relaciones de asociación de UML.

Generalización: Es una relación taxonómica entre un elemento más general (padre) y otro más específico (hijo). Se utiliza en diagramas de clases y en diagramas de casos de uso.

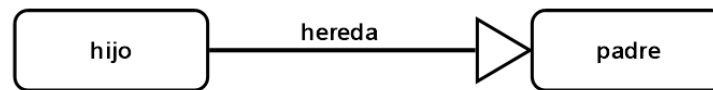


Figura 7 Relación de generalización de UML.

Está constituido por un conjunto de *diagramas* y proporciona características que permiten organizar y extender los diagramas. Están los diagramas de estructura estática, los de comportamiento y los de implementación.

- **Diagramas de estructura estática**
 - ✓ Diagramas de casos de uso.
 - ✓ Diagrama de clases.
 - ✓ Diagrama de objetos.
- **Diagramas de comportamiento**
 - ✓ Diagrama de interacción.
 - ✓ Diagrama de secuencia.
 - ✓ Diagrama de colaboración.
 - ✓ Diagrama de actividad.
 - ✓ Diagrama de estados.
- **Diagramas de implementación**
 - ✓ Diagrama de componentes.
 - ✓ Diagrama de despliegue.

Es importante señalar que un modelo UML indica qué es lo que supuestamente hará el sistema, no precisamente cómo lo hará. Pero constituye una guía y una representación de las especificaciones del sistema, ayudando a tomar decisiones para lograr un sistema que cumpla con todas las expectativas del cliente y los usuarios.

1.8 Lenguajes de programación web

Desde los inicios de Internet surgieron diferentes demandas por los usuarios y se dieron soluciones mediante la utilización de lenguajes estáticos. Al paso del tiempo, las tecnologías fueron desarrollándose y

surgieron nuevos problemas para crear nuevas soluciones. De esta manera aparecieron los lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios de forma más amigable y conectarse a diferentes bases de datos. En la actualidad existen numerosos lenguajes de programación para desarrollar la web, tanto lenguajes de lado cliente como de lado servidor.

Cuando se habla de lenguajes de lado cliente, son aquellos capaces de ser digeridos directamente por el navegador sin necesidad de un pre-tratamiento.

Entre los lenguajes de lado cliente se encuentran:

- **HTML** (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales), es un lenguaje estático para el desarrollo de sitios web, desarrollado por World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones htm, html. Entre sus ventajas se encuentra que es admitido por todos los exploradores, sencillo y de archivos pequeños. Como desventaja resalta que la interpretación de cada navegador puede ser diferente.
- **Javascript** es un lenguaje interpretado creado por Brendan Eich en la empresa Netscape Communications. El código Javascript puede ser integrado dentro de las páginas web. El W3C para evitar incompatibilidades diseñó un estándar para la modelación de objetos del documento, más conocido como DOM (en inglés Document Object Model). Entre sus ventajas está su scripting seguro y fiable y contradictoriamente su debilidad es su visibilidad al alcance de cualquier usuario.

Por otra parte se encuentran los lenguajes de lado servidor que son aquellos lenguajes reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

Entre los lenguajes de lado servidor se encuentran:

- **ASP.NET** es un lenguaje comercializado por Microsoft. Fue desarrollado para resolver las limitantes de su antecesor ASP. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Creado para desarrollar tanto web sencillas o grandes aplicaciones. Para su funcionamiento se necesita tener instalado IIS (Internet Information Server) con el Framework .Net. Entre sus ventajas se encuentra que es un lenguaje completamente orientado a objetos, su velocidad de respuesta del servidor y seguridad entre otras. El consumo de recursos es su desventaja.

- **JSP** (Java Server Pages) desarrollado por Sun Microsystems, está orientado a desarrollar páginas web en Java. Comparte ventajas similares a la de ASP.NET y fue desarrollado principalmente para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Se necesita tener instalado un servidor Tomcat para su funcionamiento. Comparado mucho por su potencialidad con ASP.NET este se destaca por ser multiplataforma. Es un tanto complejo en su aprendizaje.
- **PHP** es un lenguaje de script interpretado utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Lenguaje multiplataforma y fácil de aprender y no requiere definición de tipos de variables, siendo estas sus mayores ventajas. Entre sus desventajas se encuentra su pobre programación orientada a objetos para aplicaciones grandes, además de dificultar la organización por capas de la aplicación.

La integración entre lenguajes de lado servidor y de lado cliente hacen posible la implementación y con esta la solución a muchos de los problemas actuales (World Wide Web Consortium, 1994).

1.9 Herramientas CASE para el análisis y el diseño

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador), son aplicaciones que facilitan el desarrollo de software, reduciendo el esfuerzo, el costo y el tiempo, además de estructurar la documentación asociada a los artefactos generados. A continuación se exponen las principales características de tres de las herramientas CASE más utilizadas en el mundo en la actualidad.

1.9.1 Enterprise Architect

Es una herramienta CASE orientada a objetos que provee su alcance para el desarrollo de sistemas, administración de proyectos y análisis de negocios. Maneja totalmente el ciclo de vida de desarrollo de software, utilizando el UML como lenguaje de modelado. Facilita y soporta el levantamiento de requerimientos, el análisis y diseño, las pruebas y mantenimiento del software en desarrollo. Soporta un impresionante rango de lenguajes de desarrollo, incluyendo ActionScript, C, C++, C# y VB.NET, Java,

Visual Basic 6, Python, PHP, XSD, WSDL y otros más. Gestiona la ingeniería de código, normal e inversa, además de una efectiva documentación compatible con Microsoft Word (SPARX System Pty L td, 2000-2007).

Dentro de las funcionalidades que soporta el Enterprise Architect están:

- Diagramas UML.
- CU, Modelos Lógico, Dinámico y Físico.
- Extensiones personalizadas para modelado de procesos.
- Documentación de alta calidad compatible con MS Word.
- Modelado de Datos.
- Ingeniería directa de Base de Datos a DDL e ingeniería inversa de Base de Datos desde ODBC.
- Soporte de pruebas.
- Multi-usuario, con sistema de seguridad.

Se integra a través de plug-ins con herramientas como:

- Eclipse.
- Visual Estudio.Net.

1.9.2 Rational Rose

Es una herramienta case que soporta el flujo completo de desarrollo de software. Está basada en la metodología de desarrollo RUP. Da soporte al UML. Es un entorno de modelado que permite generar código a partir de modelos ADA, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Facilita la documentación de los artefactos generados en el proceso de desarrollo de software. Estructura el proyecto en función de las vistas de la arquitectura que plantea la metodología de desarrollo RUP, sirviendo de guía y estructura para el desarrollo del proyecto (IBM) (GSInnova, 2007).

Dentro de las funcionalidades que soporta el Rational Rose están:

- Diagramas UML.
- Desarrollo Orientado al Modelado.
- La generación de código ADA, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.

- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

Se integra con herramientas Integrated Development Environment (IDE) como:

- Borland JBuilder versiones 7.0 a 10.0
- Sun Forte for Java Community y Enterprise Editions
- Microsoft Visual Studio 6
- Microsoft Visual Studio 2003
- Microsoft Visual Studio 2005
- Wind River Tornado
- Green Hills MULTI

1.9.3 Visual Paradigm

Es una herramienta CASE que utiliza UML: como lenguaje de modelado. Soporta el ciclo de vida completo de desarrollo de software, ayuda a una rápida construcción de aplicaciones de calidad y a un menor coste. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Visual Paradigm International, 2005).

Dentro de las funcionalidades que soporta el Visual Paradigm están:

- Administración de requisitos.
- Modelado de procesos del negocio.
- Modelado de base de datos.
- Generación de código.
- Ingeniería inversa.

Se integra con herramientas Java como:

- Eclipse.
- JBuilder.
- NetBeans/sun ONE.

- Weblogic Workshop.
- JDeveloper.
- IntelliJ IDE.

Visual Paradigm da la posibilidad de lograr una mejor integración entre todos los involucrados en el desarrollo del software, brindándole la posibilidad de organizar los diagramas y documentación asociada al desarrollo del proyecto.

1.10 Conclusiones

En este capítulo se profundizó en el estudio de la evolución del análisis y diseño, así como su estado actual. Se reflejaron los patrones de casos de uso y los de diseño más usados. Se analizaron varias metodologías de desarrollo de software. Las métricas para el modelo de diseño fueron analizadas y detalladas. Se estudió sobre el lenguaje de modelado UML y los lenguajes de programación tanto de lado cliente como de lado servidor más usados. Por último se realizó un análisis de diferentes herramientas CASE, enfatizando en las facilidades que brindan.

Después de tener estos elementos se arribó a las siguientes conclusiones:

- Se utilizará la metodología de desarrollo RUP, por ser una guía eficaz para el desarrollo del análisis y diseño del módulo Presentación del sistema CCV.
- De la ingeniería de requisitos se aplicarán las etapas de identificación, análisis y especificación de requerimientos, aplicando cada una de las actividades que estas proponen, partiendo de los procesos del negocio ya definidos, como entrada a la elicitación de requerimientos.
- Se aplicarán patrones de casos de uso, pues brindarán gran ayuda en la identificación y confección del diagrama de casos de uso del módulo Presentación del sistema CCV.
- Se aplicarán los patrones de diseño necesarios para no cometer errores tradicionales en el diseño del módulo Presentación del sistema CCV.
- Se utilizará UML como lenguaje de modelado, por permitir todo el modelado necesario para el análisis y el diseño del módulo Presentación del sistema CCV.
- Se utilizará Java como lenguaje de programación el cual fue seleccionado por el equipo de arquitectura del sistema CCV.

- Se utilizará Visual Paradigm como herramienta CASE de desarrollo, pues permitirá el modelado de los artefactos del modelo del análisis y del modelo de diseño del módulo Presentación del sistema CCV.

2.1 Introducción

El presente capítulo muestra la solución que se propone para la realización del análisis y diseño del módulo Presentación del sistema CCV. Se parte de la aplicación de técnicas de ingenierías de requisitos para obtener las funcionalidades y restricciones del sistema a partir de los procesos del negocio. Luego se especifican los artefactos pertinentes para elaborar el modelo del sistema. Por último se analizan los casos de uso, sus especificaciones y requerimientos para modelar los artefactos imprescindibles para obtener un adecuado modelo de diseño.

Modelo del sistema.

Entre los principales objetivos que tiene el modelo de sistema, está definir los requerimientos funcionales y no funcionales que tendrá el mismo. Una vez definidas las fronteras del sistema se encontrarán los actores, posteriormente se obtendrán los casos de uso y una descripción detallada de cada uno de ellos, que junto al diagrama de casos de uso del sistema, darán cabida a una posterior modelación de diseño.

Modelo de diseño

El modelo de diseño tiene como principal propósito comprender, modelar y documentar todos los aspectos referentes a los requerimientos funcionales y no funcionales del sistema, además de lograr una entrada adecuada a las actividades de implementación. Para su ejecución se procede a identificar los subsistemas, paquetes y clases de diseño, además de distribuir el comportamiento de los casos de uso entre los elementos de diseño identificados.

2.2 Requerimientos de software

Teniendo como entrada los procesos de negocio del módulo Presentación, junto a la interacción con los clientes del sistema y aplicando técnicas para la elicitación de requisitos, se obtuvieron los requerimientos que debe cumplir el sistema, así como las restricciones necesarias.

Se han identificado las necesidades reales de los usuarios y las demandas de los clientes, con el fin de obtener una aplicación que sufrague las necesidades de los mismos. Los requerimientos consensados constituyen un compromiso de obligatorio cumplimiento por parte del equipo de desarrollo con los clientes de la aplicación, en lo que a desarrollo de funcionalidades se refiere.

Los requerimientos obtenidos persiguen llegar a un entendimiento entre el cliente y el equipo de desarrollo de las condiciones que debe presentar el producto desde el punto de vista funcional. Los mismos se agruparon en dos categorías, funcionales y no funcionales. Además de llevar un proceso de control de la calidad de la especificación de los mismos.

2.2.1 Requerimientos funcionales

Como resultado de las técnicas de identificación de requisitos aplicadas se obtuvieron las siguientes condiciones o capacidades que el sistema debe cumplir:

RF01.01 Proponer apertura de un período de Presentación de Proyectos

El sistema permitirá que una de las dos Secretarías Técnicas, ya sea de la parte cubana o la parte venezolana puedan proponer la apertura de un nuevo período de Presentación de Proyectos, donde se presentan proyectos del convenio entre ambos países.

RF01.01.01 Insertar datos del período de Presentación de Proyectos.

El sistema permitirá la creación de un nuevo período de Presentación de Proyectos para el cual los Entes Ejecutores (EE) tendrán la posibilidad de presentar los proyectos. Esta creación será notificada a la Secretaría Técnica (ST) contraparte para su aprobación o rechazo. Los atributos que van a conformar el período de Presentación de Proyectos son:

- Nombre de la mixta.
- Fecha en que se abre la mixta.
- Fecha en que se cierra.
- Lugar en el que se va a celebrar la mixta (ciudad, país).
- Si se realizará pre mixta o no.

En caso de ser afirmativa la anterior opción se debe indicar los datos:

- Fecha de apertura de la pre mixta.

- Lugar en el que se va a celebrar la pre mixta (ciudad, país)
- Fecha de cierre de la pre mixta.

RF01.01.02 Rechazar propuesta del período de Presentación de Proyectos

El sistema permitirá que la ST contraparte rechace la propuesta de apertura del período de Presentación de Proyectos, enviando una notificación y la propuesta rechazada con una nota donde especifica el motivo del rechazo de la propuesta.

RF01.01.03 Aceptar propuesta del período de Presentación de Proyectos

El sistema permitirá que la Secretaría Técnica contraparte acepte la propuesta de apertura del período de Presentación de Proyectos enviando una notificación a todos los implicados en el convenio. A partir de este momento los ministerios podrán comenzar a presentar los proyectos concebidos por los EE y ellos.

RF01.01.04 Notificar apertura de un período de Presentación de Proyectos

El sistema permitirá que una vez que estén de acuerdo las dos ST en la apertura de un período de Presentación de Proyectos, se enviará una notificación con los datos de apertura (especificados en el RF01.01.01) a todos los EE y Ministerios (M) informándoles la apertura del mismo.

RF01.01.05 Modificar datos de un período de Presentación de Proyectos

El sistema permitirá modificar los datos de un período de Presentación de Proyectos, en el transcurso del tiempo donde se están poniendo de acuerdo las ST en la apertura hasta su aprobación, así como después de que ya esté aprobada por ambas partes.

Además de los atributos del período que se permitirán modificar se incorpora una Nota, explicando lo modificado. Esta modificación va a generar una notificación a la ST contraparte.

Datos modificables:

- Nombre de la mixta.
- Fecha en que se abre la mixta.
- Fecha en que se cierra.
- Lugar en el que se va a celebrar la mixta (ciudad, país).
- En caso de que el período tenga pre mixta:
 - Fecha de apertura de la pre mixta.
 - Lugar en el que se va a celebrar la pre mixta (ciudad, país)
 - Fecha de cierre de la pre mixta.

En caso de que el período no tenga pre mixta:

- El sistema permitirá adicionar los datos de la pre mixta, lo que representa un modificación al período de presentación.
- Nota de modificación.

RF01.01.06 Aceptar modificación del período de Presentación de Proyectos

El sistema permitirá que la ST contraparte acepte la modificación del período de Presentación de Proyectos, enviando una notificación con los datos del período (especificados en el RF01.01.01) a todos los implicados en el convenio.

RF01.01.07 Notificar modificación de un período de Presentación de Proyectos

El sistema permitirá que se notifique cualquier modificación que se produzca por alguna de las partes en los datos del período de Presentación de Proyectos que se está concibiendo entre las dos ST.

RF01.01.08 Rechazar modificación del período de Presentación de Proyectos

El sistema permitirá que la ST contraparte rechace la modificación de apertura del período de Presentación de Proyectos con una nota del porque del rechazo de la modificación y se envía una notificación a la otra ST.

RF01.02 Concebir proyectos

El sistema permitirá que se cree un nuevo proyecto, así como su modificación, eliminación y aceptación o rechazo de esta modificación o eliminación.

RF01.02.01 Proponer proyecto

El sistema permitirá que un EE le proponga a otro un proyecto para realizarlos en conjunto, esta propuesta se hará mediante la ficha del proyecto, para la cual se requieren al menos los siguientes datos:

- Nombre del proyecto
- Entes Ejecutores
- Ministerios
- Responsables
- Objetivos
- Fundamentación

En la propuesta además de estos datos el sistema permitirá la entrada del plan operativo, que está definido por las actividades y los recursos que se le asignan a cada una para su futura ejecución.

Se enviará una notificación de la propuesta de proyectos a todos los implicados en el mismo.

RF01.02.02 Rechazar propuesta de proyecto

El sistema permitirá que el EE contraparte de un proyecto propuesto pueda rechazar el proyecto que le están proponiendo, agregando una nota explicando el por qué del rechazo. El sistema enviará una notificación automática a todos los implicados del rechazo.

RF01.02.03 Notificar rechazo de propuesta de proyecto

El sistema permitirá que se notifique el rechazo a todos los implicados de la propuesta de proyecto que le hicieron al EE que rechaza.

RF01.02.04 Aceptar propuesta de proyecto

El sistema permitirá que el EE contraparte de un proyecto propuesto pueda aceptar el proyecto que le están proponiendo, enviando una notificación a todos los implicados y dando paso al llenado de la ficha entre los EE implicados.

RF01.02.05 Notificar aceptación de propuesta de proyecto

El sistema permitirá que se notifique a los EE implicados en un proyecto, la aceptación de la propuesta que se ha elaborado.

RF01.02.06 Modificar una propuesta de proyecto

El sistema permitirá que se modifique una propuesta de proyecto cuando esta haya sido rechazada por el EE al cual le fue propuesto, dando la posibilidad de poder seleccionar otro EE contraparte.

RF01.02.07 Eliminar propuesta de proyecto

El sistema permitirá eliminar una propuesta de proyecto una vez que se haya rechazado por el ente al cual le fue propuesto.

RF01.02.08 Modificar ficha de proyecto

El sistema permitirá que los EE, M y ST modifiquen la ficha del proyecto, esta operación la pueden realizar cualquiera de las dos partes enviándose una notificación a todos los implicados, para modificar un proyecto se editarán todos los campos que se permiten cambiar de la ficha del proyecto y que se muestran a continuación:

- Nombre del proyecto.

- Ministerios por Cuba.
- Entes Ejecutores por Cuba.
- Ministerios por Venezuela.
- Entes Ejecutores por Venezuela.
- Responsable por Cuba.
- Responsable por Venezuela.
- Objetivos.
- Fundamentación.
- Duración.
- Plan Operativo.

RF01.02.09 Rechazar modificación de ficha de proyecto

El sistema permitirá que se rechace la modificación de un proyecto enviando una notificación a todos los implicados, este rechazo estará complementado de una nota que explica el motivo del rechazo de la modificación.

RF01.02.010 Notificar rechazo de modificación de ficha de proyecto

El sistema permitirá notificar a los implicados en el proyecto el rechazo de la modificación que se produjo sobre el proyecto.

RF01.02.011 Aceptar modificación de ficha de proyecto

El sistema permitirá que el EE, los M y las ST acepten la modificación de un proyecto.

RF01.02.012 Notificar aceptación de modificación de ficha de proyecto

El sistema permitirá notificar a los implicados en el proyecto que la modificación ha sido aceptada.

RF01.02.013 Eliminar ficha de proyecto

El sistema permitirá que los EE, los M y las ST puedan eliminar la ficha de un determinado proyecto.

RF01.02.014 Rechazar eliminación de ficha de proyecto

El sistema permitirá que se rechace la eliminación de un proyecto enviando una notificación a todos los implicados, este rechazo estará complementado de una nota que explica el motivo del rechazo de la modificación.

RF01.02.015 Notificar rechazo de eliminación de ficha de proyecto

El sistema permitirá notificar a los implicados en el proyecto el rechazo de la eliminación que se produjo sobre el proyecto.

RF01.02.016 Aceptar eliminación de proyecto

El sistema permitirá que el EE, los My las ST acepten la eliminación de un proyecto.

RF01.02.017 Notificar aceptación de eliminación de ficha de proyecto

El sistema permitirá notificar a los implicados en el proyecto que la eliminación ha sido aceptada y que se ha eliminado el proyecto.

RF01.02.018 Enviar ficha del proyecto a los Ministerios

El sistema permitirá que uno de los EE proponga el envío del proyecto a ser revisado por los M.

RF01.02.019 Aceptar envío de proyecto a los Ministerios

El sistema permitirá que el EE contraparte acepte el envío del proyecto a ser revisado por los M, siendo este finalmente enviado y enviando una notificación a todos los implicados.

RF01.02.020 Notificar aceptación de envío de proyecto a los Ministerios

El sistema permitirá notificar a los implicados que el envío del proyecto a los M ha sido aceptado.

RF01.02.021 Rechazar envío de proyecto a los Ministerios

El sistema permitirá que el EE contraparte rechace el envío del proyecto a ser revisado por los M, con una nota que explica el porqué del rechazo.

RF01.02.022 Notificar rechazo de envío de proyecto a los Ministerios

El sistema permitirá notificar a los implicados en el proyecto, que el envío del proyecto a los M ha sido rechazado.

RF01.02.023 Rechazar proyecto por los Ministerios

El sistema permitirá que los M rechacen un proyecto determinado, especificando en una nota el porqué del rechazo.

RF01.02.024 Notificar rechazo de proyecto por los Ministerios

El sistema permitirá que se notifique a todos los implicados el rechazo de un proyecto por parte de los M.

RF01.02.025 Aceptar proyecto por los Ministerios

El sistema permitirá que los M acepten un proyecto propuesto por los EE, quedando de esta manera listos para ser presentados en un período de Presentación de Proyectos.

RF01.02.026 Notificar aceptación de de proyectos por los Ministerios

El sistema permitirá que se notifique a todos los implicados en el proyecto, que el mismo haya sido aceptado por los M.

RF01.02.027 Definir plan operativo

El sistema permitirá definir un plan operativo de un proyecto en el cual se definen las actividades y los recursos asociados a cada actividad con el gasto asociado a cada recurso según su tipo.

RF01.02.028 Insertar actividad

El sistema permitirá insertar una nueva actividad al plan operativo, la misma requiere de los diferentes datos:

- Nombre de la actividad
- Fecha de inicio
- Fecha de fin
- Ejecutor
- Responsable
- Receptor
- Recursos

RF01.02.029 Asignar recursos humanos a una actividad

El sistema permitirá que se asignen recursos humanos a una actividad, para que la ejecuten y poder de esta forma registrar el gasto por concepto. A continuación se definen los siguientes datos para la asignación de estos recursos:

- Categoría
- Cantidad
- Tiempo
- Viáticos
- Pasaje
- Descripción

RF01.02.030 Asignar recursos materiales

El sistema permitirá que se asignen recursos materiales a la actividad para poder registrar el gasto por concepto de recursos materiales en el proyecto. A continuación se definen los siguientes datos para la asignación de estos recursos:

- Nombre del recurso
- Precio por cada unidad
- Cantidad
- Monto Total
- Descripción

RF01.02.031 Asignar gastos administrativos

El sistema permitirá que se asignen gastos administrativos a la actividad para poder registrar el gasto por concepto de gastos administrativos en el proyecto. A continuación se definen los siguientes datos para la asignación de estos recursos:

- Nombre del recurso
- Monto
- Descripción

RF01.02.032 Asignar otros tipos de recursos

El sistema permitirá que se asignen otros recursos a la actividad para poder registrar el gasto por concepto de otros recursos en el proyecto. A continuación se definen los siguientes datos para la asignación de estos recursos:

- Nombre del recurso
- Monto
- Tipo de recursos
- Descripción

RF01.02.033 Modificar actividad

El sistema permitirá que se modifiquen las actividades que conforman el plan operativo de un proyecto, se permitirán modificar todos los datos que se solicita en (RF01.02.028).

RF01.02.034 Modificar recursos humanos de una actividad

El sistema permitirá que se modifiquen los RRHH asociados a una actividad del plan operativo, se permitirán modificar todos los datos que se solicita en (RF01.02.030).

RF01.02.035 Modificar recursos materiales

El sistema permitirá que se modifiquen los recursos materiales asociados a una actividad del plan operativo, se permitirán modificar todos los datos que se solicita en (RF01.02.031).

RF01.02.036 Modificar gastos administrativos

El sistema permitirá que se modifiquen los recursos materiales asociados a una actividad del plan operativo, se permitirán modificar todos los datos que se solicita en (RF01.02.032).

RF01.02.037 Modificar otros tipos de recursos

El sistema permitirá que se modifiquen otros tipos de recursos asociados a una actividad del plan operativo, se permitirán modificar todos los datos que se solicita en (RF01.02.033).

RF01.02.038 Agregar tipo de recurso

El sistema permitirá agregar un nuevo tipo de recurso que no existe en el sistema para el proyecto que se está definiendo.

RF01.03 Presentar proyectos

El sistema permitirá que los M propongan la presentación de los proyectos que se concibieron con anterioridad, los cuales deben ser aprobados para su presentación por el M contraparte que está implicado.

RF01.03.01 Proponer Presentación de Proyectos

El sistema permitirá que uno de los M escoja los proyectos a presentar de los que se han concebido con anterioridad y le envíe la propuesta al Ministerio contraparte para que acepte o rechace su presentación.

RF01.03.02 Aceptar Presentación de Proyectos

El sistema permitirá que el M contraparte acepte la presentación del proyecto que le propuso el otro M implicado en el proyecto.

RF01.03.03 Notificar aceptación de Presentación de Proyectos

El sistema permitirá que se envíe una notificación una vez que se acepte la presentación del proyecto al Ministerio que lo propuso.

RF01.03.04 Rechazar Presentación de Proyectos

El sistema permitirá que el M contraparte rechace la presentación del proyecto que le propuso el otro M implicado en el proyecto, agregando una nota que explica el porqué del rechazo.

RF01.03.05 Notificar rechazo de Presentación de Proyectos

El sistema permitirá que se envíe una notificación una vez que se rechace la presentación del proyecto, para que el otro M conozca del rechazo de su propuesta de presentación.

RF01.04 Presentar proyectos expeditos

El sistema permitirá que los M puedan proponer la Presentación de Proyectos que por alguna razón no entraron en la fecha requerida del período de Presentación de Proyectos y por su importancia es necesario que se presenten.

RF01.04.01 Proponer Presentación de Proyectos Expeditos

El sistema permitirá que los ministerios propongan la presentación de algún proyecto expedito.

RF01.04.02 Notificar proposición de Presentación de Proyectos Expeditos

El sistema permitirá notificar al M contraparte implicado, la propuesta de Presentación de Proyectos expeditos, para que sea evaluado.

RF01.04.03 Rechazar proposición de Presentación de Proyectos Expeditos

El sistema permitirá rechazar la proposición de presentar algún proyecto expedito, agregando una nota donde explica el por qué del rechazo.

RF01.04.04 Notificar rechazo de proposición de Presentación de Proyectos Expeditos

El sistema permitirá notificar al M contraparte del rechazo de la proposición de presentación del Proyecto expedito que se evalúa.

RF01.04.05 Aceptar Presentación de Proyectos Expeditos

El sistema permitirá que el M contraparte acepte la proposición del proyecto expedito, el cual pasaría a ser evaluado por las ST para su aprobación.

RF01.04.06 Notificar aceptación de Presentación de Proyectos Expeditos

El sistema permitirá notificar a todos los implicados que el proyecto expedito solicitado para su presentación fue aceptado.

RF01.05 Evaluar Proyectos

El sistema permitirá que las ST evalúen los proyectos que se hayan presentados en la mixta y en la pre mixta en caso que se haya efectuado, para de esta forma darles paso al proceso de asignar financiamiento por parte de la ST venezolana.

RF01.05.01 Aceptar proyecto

El sistema permitirá que las ST acepten los proyectos que se presentaron en el período de Presentación de Proyectos, el proyecto queda aceptado y listo para financiamiento si es aceptado por las dos ST.

RF01.05.02 Notificar aceptación de proyecto

El sistema permitirá notificar la aceptación del proyecto por parte de las ST a todos los implicados.

RF01.05.03 Rechazar proyecto

El sistema permitirá que las ST rechacen los proyectos que se presentaron en el período de Presentación de Proyectos, si una de las dos ST rechaza el proyecto, ya este queda rechazado con una nota que explica el por qué del rechazo.

RF01.05.04 Notificar rechazo del proyecto

El sistema permitirá notificar a todos los implicados el rechazo de un proyecto determinado.

RF01.06 Cerrar el período de Presentación de Proyectos

El sistema permitirá que las ST decidan cuando será el cierre del período de Presentación de Proyectos, la cual marca el fin de la mixta, e indicando el inicio del proceso de financiamiento de los proyectos que fueron aceptados por todos los niveles en este período.

RF01.06.01 Proponer cierre del período de Presentación de Proyectos

El sistema permitirá que una de las ST proponga la fecha de cierre del período de Presentación de Proyectos.

RF01.06.02 Notificar propuesta de cierre del período de Presentación de Proyectos

El sistema permitirá notificar a la otra ST la propuesta de cierre del período de Presentación de Proyectos.

RF01.06.03 Rechazar propuesta de cierre del período de Presentación de Proyectos

El sistema permitirá que se rechace la propuesta de cierre del período de Presentación de Proyectos por parte de la ST contraparte, con una nota explicando el por qué del rechazo.

RF01.06.04 Notificar rechazo de cierre del período de Presentación de Proyectos

El sistema permitirá notificar a la ST contraparte el rechazo de la propuesta de cierre del período de Presentación de Proyectos.

RF01.06.05 Aceptar cierre del período de Presentación de Proyectos

El sistema permitirá que se acepte la propuesta de cierre del período de Presentación de Proyectos, quedando de esta forma cerrado el período de Presentación de Proyectos.

RF01.06.06 Notificar cierre del período de Presentación de Proyectos

El sistema permitirá notificar a todos los implicados en el Convenio Cuba-Venezuela que se cerró el período de Presentación de Proyectos.

RF01.07 Aprobar financiamiento

El sistema permitirá que la ST venezolana analice todos los proyectos que se aprobaron y procedan a asignar el financiamiento, asociándoselo a una o varias fuentes y teniendo en cuenta el monto que se le ha asignado a esa mixta.

RF01.08 Crear Noticias

El sistema permitirá que la ST y los M puedan crear noticias, mostrando si es ST el usuario autenticado, un listado con todas las noticias creadas por ella y por los M. Si es un usuario de Ministerio el que esta autenticado se muestra, un listado con todas las noticias creadas por él.

RF01.09 Publicar Noticia

El sistema permitirá que la ST publique las noticias creadas por ella y por los M, para esto se muestra la lista de las noticias que tienen estado de no publicadas, es decir que están creadas.

RF01.10 Editar Noticia

El sistema permitirá que una vez creada la noticia o publicada, esta puede ser editada por la ST, es decir modificar los campos que crean necesario.

RF01.11 Eliminar Noticia

El sistema permitirá que una vez creada la noticia o publicada, esta puede ser eliminada por la ST.

RF01.12 Eliminar Publicación

El sistema permitirá la eliminación de la publicación de una noticia por la ST, para esto se muestra el listado de noticias creadas por ella y por los M y al acceder al link de la que tiene estado publicada se muestra una interfaz con la opción de eliminar publicación.

RF01.13 Mostrar Noticia

El sistema permitirá mostrar las noticias creadas, que pueden estar en estado publicada o no publicada, para esto sí es ST le sale el listado de todas las noticias creadas por ella y por los M y puede acceder al link de la noticia, mostrándose la misma. Si el usuario es de M le sale el listado de todas las noticias creadas por él, ya sea en estado publicada o no publicada y puede acceder al link de la noticia, mostrándose la misma.

2.2.2 Requerimientos no funcionales

Las cualidades o propiedades que debe cumplir el sistema son:

Orientados al usuario

Seguridad:

- RNF01.01** Autenticación obligatoria y segura.
- RNF01.02** Los datos que circulan por la red no viajan en texto plano.
- RNF01.03** Acceso a la información según el rol.
- RNF01.04** Manejo de sesiones del usuario, expira la sesión en x tiempo
- RNF01.05** Debe permitir ocultar la información que aparece en la URL
- RNF01.06** Realizar salvallas periódicamente de la información contenida en la base de datos.
- RNF01.07** El sistema debe recuperarse ante fallos, ya sea por pérdida de conexión, alimentación.
- RNF01.08** Las secciones de los usuarios debe espirar después de 10 min de inactividad.

Usabilidad:

- RNF01.09** Permitir uso del teclado para realizar operaciones sobre el sistema.
- RNF01.010** Debe poseer una interfaz agradable para el cliente de acuerdo a los estándares de diseño.
- RNF01.011** Mostrar la información de forma lógica y correctamente estructurada.

Robustez:

- RNF01.012** El sistema debe estar accesible en todo momento.
- RNF01.013** El sistema debe recuperarse en poco tiempo tras cualquier anomalía.
- RNF01.014** Se debe hacer salvallas automáticas de la Base de Datos, para no perder la información.

Disponibilidad:

- RNF01.015** El sistema debe estar accesible desde internet.

Rendimiento (Tiempo de respuesta, capacidad y rendimiento):

- RNF01.016** Las páginas de la aplicación deben cargar en un tiempo inferior a 15 segundos.
- RNF01.017** Debe garantizarse que con 300 usuarios conectados concurrentemente no disminuya el rendimiento y rapidez de la aplicación.
- RNF01.018** El tiempo de carga de la aplicación debe ser de 10 a 25 segundos.

Orientados al desarrollador:**Disponibilidad:**

RNF01.019 Tener una correcta y completa configuración del entorno de trabajo.

Portabilidad:

RNF01.020 Debe permitirse ser usado en cualquier plataforma.

Adaptabilidad:

RNF01.021 El sistema debe garantizar la configuración y cambio de sus parámetros de forma fácil y rápida.

Comprensibilidad:

RNF01.022 Debe garantizarse el uso de estándares de codificación.

Requisitos para los clientes:

Hardware:

RNF01.023 CPU Intel Pentium 4.

RNF01.024 Memoria RAM Mínimo 768 Mb, Memoria RAM Máximo 1 Gb.

RNF01.025 Capacidad Disco Duro 160 Gb.

Requisitos para Servidores:

Hardware:

RNF01.026 CPU Intel Pentium 4.

RNF01.027 Memoria RAM Mínimo 1 Gb, Memoria RAM Máximo 2 Gb.

RNF01.028 Capacidad Disco Duro 160 Gb.

Software:

RNF01.029 Sistema Operativo: GNU/Linux Debian Etch 4.0.

RNF01.030 Servidor Web Apache 2.0.

RNF01.031 Servidor Web Tomcat 5.5.

2.3 Actores del sistema

Al definir las fronteras del módulo Presentación se encontraron los actores del sistema que interactúan con las funcionalidades del módulo. Para una mejor definición de los mismos se aplicó el patrón de múltiple actores para generalizar las acciones comunes que realizan varios de ellos en un actor ficticio. Los actores definidos fueron:

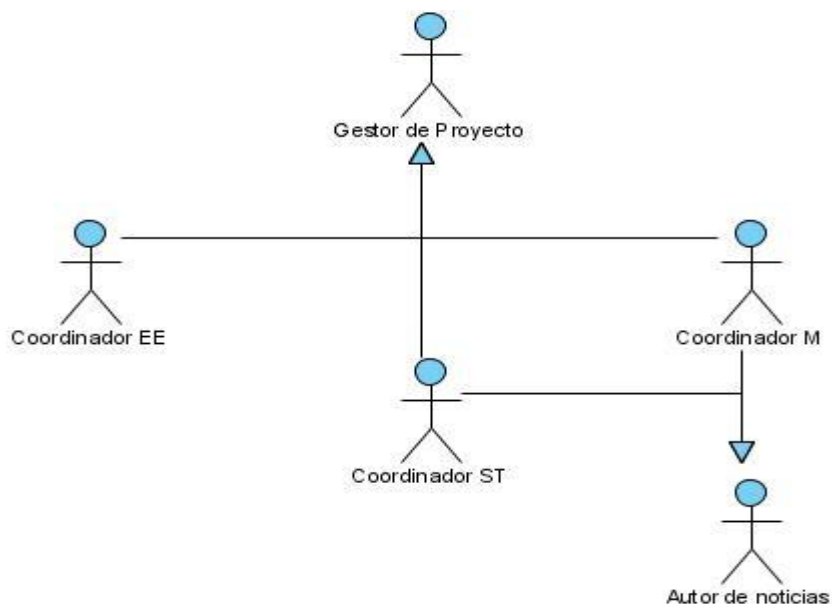


Figura 8 Actores del sistema.

Tabla 4 Descripción de los actores del sistema.

Actor	Descripción
Gestor de Proyecto	Es el actor que accede al sistema, se autentica y se le da un determinado nivel de acceso según su rol. Este actor puede, modificar y eliminar una ficha de proyecto. También puede aceptar o rechazar una modificación o eliminación que se haya realizado sobre la ficha de proyecto.

Coordinador de EE (Entidad Ejecutora)	El actor es el encargado de crear los proyectos, se comporta como Gestor de Proyecto heredando todos sus permisos para ejecutar las funcionalidades a la cual puede acceder el actor, además puede enviar los proyectos concebidos a los ministerios para que estos sean revisados.
Coordinador de M (Ministerio)	El actor puede comportarse como Gestor de Proyecto heredando todos los permisos para ejecutar las funcionalidades a la cual puede acceder el actor, además puede Aceptar o Rechazar los proyectos que son enviados a los ministerios y presentar los proyectos a las ST.
Coordinador de ST(Secretaría Técnica)	El actor puede comportarse como Gestor de Proyecto heredando todos los permisos para ejecutar las funcionalidades a la cual puede acceder el actor, además puede gestionar período de presentación, aceptar o rechazar presentación de proyectos, asignar financiamiento y administrar las noticias que son creadas por los ministerios y por las ST.
Autor de Noticias	Es el actor que accede a la funcionalidad de crear noticias.

2.4 Casos de usos

Cada forma en que los actores usan el sistema constituye un caso de uso (CU). Luego de definidos los requerimientos del sistema y aplicando patrones como es el caso de CRUD, se obtuvieron las agrupaciones de funcionalidades que aportan resultados de valor para los actores del sistema. A continuación se enuncian los CU determinados para el módulo Presentación del sistema CCV.

2.4.1 CU: Gestionar propuesta de proyecto.

Caso de Uso:	Gestionar propuesta de proyecto.
Actores:	Coordinador EE
Resumen:	El caso consiste en crear una propuesta de proyecto que será presentada a un EE contraparte, esta propuesta podrá ser modificada o eliminada si está pendiente la aprobación. La propuesta deberá ser aceptada o rechazada por el EE contraparte.
Referencias:	RF01.02.01, RF01.02.02, RF01.02.03, RF01.02.04, RF01.02.05, RF01.02.06, RF01.02.07.

2.4.2 CU: Gestionar proyecto.

Caso de Uso:	Gestionar proyecto
Actores:	Gestor de proyecto
Resumen:	El caso de uso consiste en modificar, eliminar un proyecto. También se puede aceptar o rechazar tanto una modificación como eliminación de proyecto realizada por la contraparte.
Referencias	RF01.02.08, RF01.02.09, RF01.02.010, RF01.02.011, RF01.02.012, RF01.02.013, RF01.02.014, RF01.02.015, RF01.02.016, RF01.02.017.

2.4.3 Gestionar plan operativo.

Caso de Uso:	Gestionar plan operativo
Actores:	Gestor de proyecto
Resumen:	El caso de uso consiste en crear o modificar un plan operativo para un proyecto que se esté definiendo.
Referencias:	RF01.02.027, RF01.02.028, RF01.02.029, RF01.02.030, RF01.02.031, RF01.02.032, RF01.02.033, RF01.02.034, RF01.02.035, RF01.02.036,

RF01.02.037, RF01.02.038.

2.4.4 Gestionar Período de Presentación

Caso de Uso:	Gestionar Período de Presentación
Actores:	Coordinador de ST
Resumen:	El caso consiste en crear, modificar y eliminar un período de Presentación de Proyectos. También se puede aceptar o rechazar tanto la creación como una modificación y una eliminación realizada por la contraparte.
Referencias:	RF01.01.01, RF01.01.02, RF01.01.03, RF01.01.04, RF01.01.05, RF01.01.06, RF01.01.07, RF01.01.08, RF01.06.01, RF01.06.02, RF01.06.03, RF01.06.04, RF01.06.05, RF01.06.06.

2.4.5 Enviar a Ministerios

Caso de Uso:	Enviar a Ministerios
Actores:	Coordinador EE
Resumen:	El caso de uso consiste en enviar un proyecto que ha sido aprobado por los EE responsables del proyecto a ser evaluado por los M. El EE contraparte deberá aprobar o rechazar el envío del proyecto a ser evaluado por los M.
Referencias:	RF01.02.018, RF01.02.019, RF01.02.020, RF01.02.021, RF01.02.022.

2.4.6 Aceptar o rechazar proyectos

Caso de Uso:	Aceptar o rechazar proyectos
Actores:	Coordinador M
Resumen:	El caso de uso consiste en aceptar o rechazar un proyecto que se le ha enviado a los M para su aprobación, proyecto que ha sido aprobado por los EE responsables del proyecto con anterioridad. Los M implicados en el proyecto deberán aprobar o rechazar el proyecto que se le ha enviado. El caso de uso termina con la evaluación que da el actor al proyecto.

Referencias:	RF01.02.023, RF01.02.024, RF01.02.025, RF01.02.026.
---------------------	---

2.4.7 Presentar proyectos

Caso de Uso:	Presentar proyectos
---------------------	----------------------------

Actores:	Coordinador M
-----------------	---------------

Resumen:	El caso de uso consiste en seleccionar de los proyectos que ya están concebidos los cuales serán presentados en el período de Presentación de Proyectos, es decir en la mixta o en la pre mixta en caso de existir en el período. Además el M contraparte al que propone la presentación del proyecto debe aceptar o rechazar que se presente el proyecto en el período de Presentación.
-----------------	--

Referencias:	RF01.03, RF01.03.01, RF01.03.02, RF01.03.03, RF01.03.04, RF01.03.05 RF01.04, RF01.04.01, RF01.04.02, RF01.04.03, RF01.04.04, RF01.04.05, RF01.04.05.
---------------------	--

2.4.8 Aceptar o rechazar presentación de proyecto

Caso de Uso:	Aceptar o rechazar presentación de proyecto
---------------------	--

Actores:	Coordinador ST
-----------------	----------------

Resumen:	El caso de uso consiste en evaluar los proyectos que se han presentado en el período de presentación de proyectos, las ST analizan los proyectos y aceptan o rechazan los proyectos que se han presentado en el período de Presentación de Proyectos.
-----------------	---

Referencias:	RF01.05, RF01.05.01, RF01.05.02, RF01.05.03, RF01.05.04
---------------------	---

2.4.9 Asignar financiamiento

Caso de Uso:	Asignar financiamiento
---------------------	-------------------------------

Actores:	Coordinador ST(venezolana)
-----------------	----------------------------

Resumen:	El caso de uso consiste en seleccionar los proyectos que han sido
-----------------	---

aceptados por las ST e irles asignando financiamiento para su ejecución. La ST venezolana es la que asigna el financiamiento especificando las fuentes de financiamiento. Y el monto por cada una de ellas.

Referencias: RF01.07

2.4.10 Crear Noticias

Caso de Uso: Crear Noticias

Actores: Autor de noticias

Resumen: El caso de uso consiste en crear nuevas noticias por la Secretaría Técnica (cubana o venezolana) o por los Ministerios (cubanos o venezolanos).

Referencias: RF0.1.08.

2.4.11 Administrar Noticias

Caso de Uso: Administrar Noticias

Actores: Coordinador ST

Resumen: El caso de uso consiste en publicar, editar y eliminar noticias, creadas por Secretaría Técnica (cubana o venezolana) y sus Ministerios.

Referencias: RF01.09, RF01.10, RF01.11, RF01.12, RF01.13

2.5 Diagrama de casos de uso del sistema

Determinados los requerimientos, los actores y casos de usos del sistema se estructura el diagrama de casos de usos del sistema (DCUS), que representa la relación de los actores del sistema con los casos de usos. Para la elaboración del diagrama se aplicaron los patrones de casos de uso: extensión concreta y múltiple actores. Finalmente quedo de la siguiente manera:

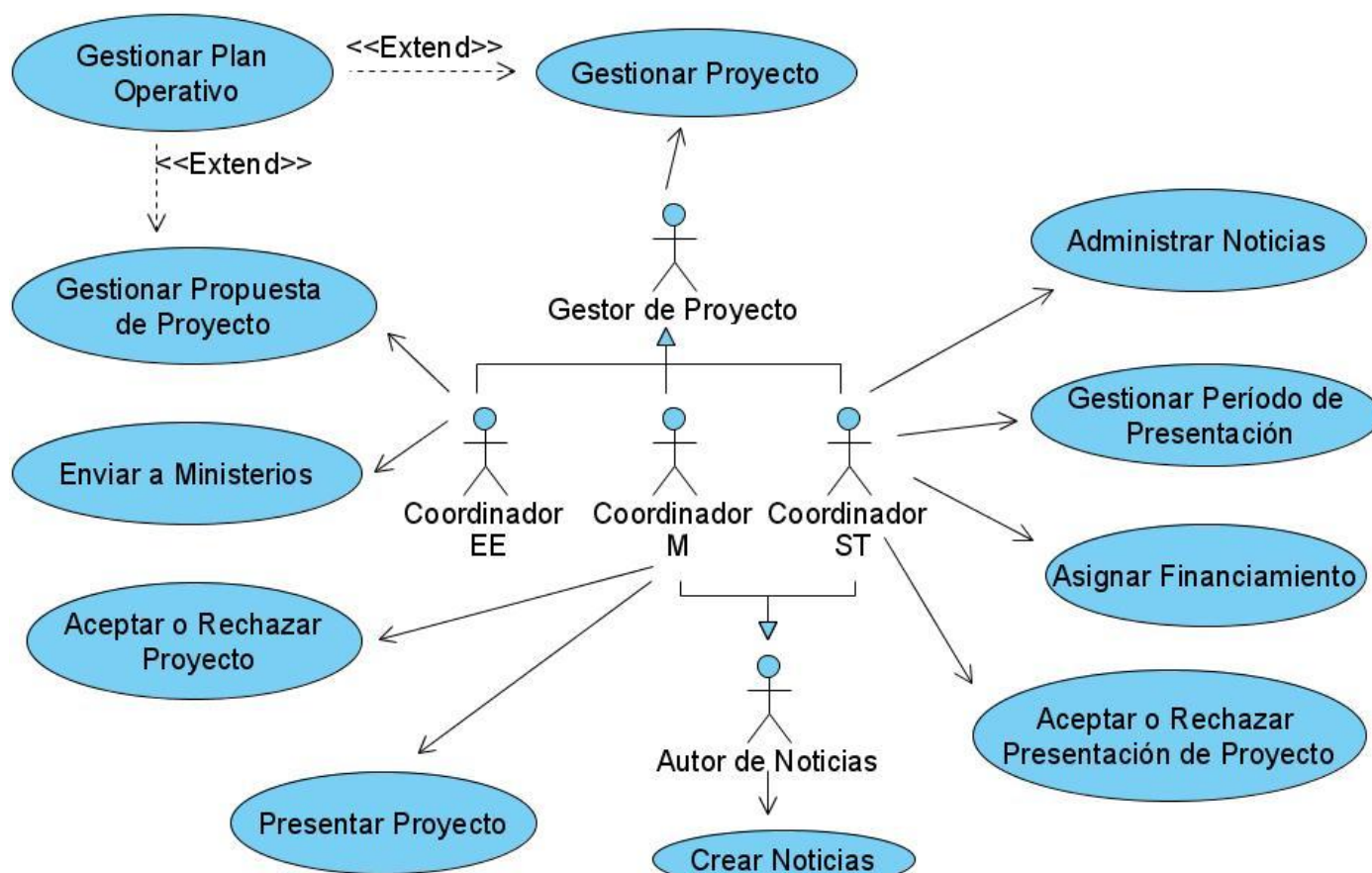


Figura 9 Diagrama de casos de uso del sistema

2.6 Descripción de casos de uso del sistema

Luego de definir los casos de uso del sistema, se procede a la descripción textual de los mismos con la intención de especificar en detalles cada una de las funcionalidades que deben ser implementadas. La descripción de los casos de uso constituye una guía para los desarrolladores y un documento de obligatorio cumplimiento en cuanto a desarrollo de funcionalidades en el sistema. Seguidamente se muestra la descripción de uno de los principales casos de uso del módulo Presentación. Para acceder a la descripción íntegra de todos los CU, ver el documento Modelo de casos de uso del sistema CCV, Módulo Presentación (Villalón Cruzata, et al., 2008).

Caso de Uso:	Gestionar propuesta de proyecto.	
Actores:	Coordinador EE (Inicia el caso de uso).	
Resumen:	El caso consiste en crear una propuesta de proyecto que será presentada a un ente contraparte, esta propuesta podrá ser modificada o eliminada si está pendiente la aprobación. La propuesta deberá ser aceptada o rechazada por el ente contraparte.	
Precondiciones:	El sistema debe estar instalado y ejecutándose correctamente. El actor debe estar autenticado con los permisos necesarios.	
Referencias	RF01.02.01, RF01.02.02, RF01.02.03, RF01.02.04, RF01.02.05, RF01.02.06, RF01.02.07.	
Prioridad	Alta	
Nivel	Usuario	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
1	El caso de uso se inicia cuando el usuario accede al módulo de Presentación.	2 El sistema muestra el menú para el módulo según el rol del usuario autenticado.
3	Accede al submenú Ficha de Proyecto.	4 El sistema muestra la lista de proyectos que se han creado donde se visualizan los siguientes datos: <ul style="list-style-type: none"> • Nombre de Proyecto. • Ministerio Contraparte. • Ente Contraparte. • Monto Total. • Estado (El proyecto se muestra en el color correspondiente a los estados definidos. Ver Documento de Estados de Proyecto).

<p>5 El actor Coordinador EE selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Nuevo proyecto. • Modificar. • Eliminar. • Ver detalles. 	
Flujo Normal de Eventos	
Sección “Crear propuesta de proyecto”	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra una pantalla con la ficha del proyecto que contiene los datos generales del proyecto.</p> <ul style="list-style-type: none"> • Nombre del Proyecto. • Duración. • Modalidad. • Ministerio responsable venezolano. • Ente Ejecutor venezolano. • Ministerio responsable cubano. • Ente Ejecutor cubano.
<p>2. Introduce los datos generales solicitados y oprime el botón “Siguiente”</p>	
	<p>3. Valida la información suministrada.</p>
	<p>4. El sistema muestra una pantalla para ingresar otros datos del proyecto.</p> <ul style="list-style-type: none"> • Fundamentación. • Descripción breve del proyecto. • Objetivos del proyecto. • Objetivos específicos del proyecto.

	<ul style="list-style-type: none">• Metas del proyecto.• Impacto del proyecto.• Población beneficiada.
5. El usuario ingresa los datos requeridos en la pantalla anterior. Oprime el botón "Siguiente".	
	6. Valida la información suministrada.
	7. Muestra una pantalla para gestionar el plan operativo. (Ver Caso de uso Gestionar Plan Operativo).
8. Oprime el botón "Finalizar".	9. El sistema crea la propuesta de proyecto y se la envía al ente contraparte para su aprobación. Guardando la propuesta en estado de aprobación por el ente contraparte.
	10. El caso de uso termina.
Prototipo de Interfaz	

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio
usuario

[NAVEGACION SECUNDARIA]

NOTICIAS

Título Noticia

En este espacio se vera un resumen de la noticia, la misma abarcará solamente dos líneas.

[mas detalles](#)

Título Noticia

En este espacio se vera un resumen de la noticia, la misma abarcará solamente dos líneas.

[mas detalles](#)

NOTIFICACIONES

Fecha	Institución	Tipo de documento	Titulo del documento	Estado del documento	
dd/mm/aaaa	Nombre Institución	Ficha de proyecto	CCV	Rechazado	
dd/mm/aaaa	Nombre Institución	Ficha de proyecto	VII Mixta	Aceptado	
dd/mm/aaaa	Nombre Institución	Mixta	Pre Mixta	Pendiente	
dd/mm/aaaa	Nombre Institución	Contrato	Contrato	Pendiente	
dd/mm/aaaa	Nombre Institución	Cronograma financiero	Cronograma	Rechazado	
dd/mm/aaaa	Nombre Institución	Solicitudes	Solicitudes RH	Aceptado	

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ▶ Presentación ▶ Documentos ▶ Fichas de proyectos usuario

Procesos

Proponer proyecto

- Otros vínculos
- Otros vínculos
- Otros vínculos

Concebir proyecto

Documentos

Fichas de proyectos

- Otros vínculos
- Otros vínculos
- Otros vínculos

Reportes

Proyectos

Fichas de proyectos

- Otros vínculos
- Otros vínculos
- Otros vínculos

Fichas de proyectos

Filtrar la búsqueda

Nombre del proyecto	Ente contraparte
<input type="text"/>	<input type="text"/>
Ministerio	Marco de aprobación
<input type="text"/>	<input type="text"/>

--	--	--	--

Nombre del proyecto	Ministerio Contraparte	Ente Contraparte	Monto Total	Estado
Convenio Cuba-Venezuela	MENPET	FUNDELEC	1,400,000.00	Espera aceptación de ente venezolano
Servicios de mecanización	MAT	Corporación venezolana agraria	2,000,000.00	Rechazado por ente venezolano
Servicios de mecanización	MAT	Corporación venezolana agraria	2,000,000.00	Proyecto en elaboración
Servicios de mecanización	MAT	Corporación venezolana agraria	2,000,000.00	Espera aceptación de ente cubano
Servicios de mecanización	MAT	Corporación venezolana agraria	2,000,000.00	Espera aceptación de ente venezolano

|< << Página 1 de 4 >> >|

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

[LOGO]	[BANNER]
Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución	
Inicio ► Presentación ► Documentos ► Fichas de proyectos ► Nuevo Proyecto ► Datos del proyecto usuario	
<div style="border: 1px solid gray; padding: 2px;">Procesos</div> <div style="border: 1px solid gray; padding: 2px;">Proponer proyecto</div> <div style="border: 1px solid gray; padding: 2px;">Concebir proyecto</div> <div style="border: 1px solid gray; padding: 2px;">Documentos</div> <div style="border: 1px solid gray; padding: 2px;">Fichas de proyectos</div> <div style="border: 1px solid gray; padding: 2px;">Reportes</div> <div style="border: 1px solid gray; padding: 2px;">Proyectos</div> <div style="border: 1px solid gray; padding: 2px;">Fichas de proyectos</div>	<p>Datos del proyecto</p> <p>Nombre del proyecto <input type="text"/></p> <p>Duración en meses <input type="text"/></p> <p>Modalidad</p> <p>Equipamientos <input type="checkbox"/></p> <p>Solución integral <input type="checkbox"/></p> <p>Asistencia técnica <input type="checkbox"/></p> <p>Suministros <input type="checkbox"/></p> <p>Ministerio venezolano responsable <input type="text"/></p> <p>Ente ejecutor venezolano <input type="text"/></p> <p>Ministerio cubano responsable <input type="text"/></p> <p>Ente ejecutor cubano <input type="text"/></p> <p style="text-align: right;"> <input type="button" value="Siguiente"/> <input type="button" value="Cancelar"/> </p>
COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela Todos los derechos reservados	

[LOGO]	[BANNER]
<input type="button" value="Presentación"/> <input type="button" value="Contratación"/> <input type="button" value="Seguimiento"/> <input type="button" value="Configuración"/> <input type="button" value="Administración"/> <input type="button" value="Misiones"/> <input type="button" value="Novedades Institución"/>	
Inicio ► Presentación ► Documentos ► Fichas de proyectos ► Nuevo Proyecto ► Datos del proyecto usuario	
<div style="border: 1px solid gray; padding: 2px;">Procesos</div> <div style="border: 1px solid gray; padding: 2px;">Proponer proyecto</div> <div style="border: 1px solid gray; padding: 2px;">Concebir proyecto</div> <div style="border: 1px solid gray; padding: 2px;">Documentos</div> <div style="border: 1px solid gray; padding: 2px;">Fichas de proyectos</div> <div style="border: 1px solid gray; padding: 2px;">Reportes</div> <div style="border: 1px solid gray; padding: 2px;">Proyectos</div> <div style="border: 1px solid gray; padding: 2px;">Fichas de proyectos</div>	<p>Datos del proyecto</p> <p>Fundamentación</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Descripción breve del proyecto</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Objetivos del proyecto</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Objetivos específicos del proyecto</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Metas del proyecto</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Impacto del proyecto</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Población beneficiada <input style="width: 80%; height: 20px;" type="text"/></p> <p style="text-align: right;"> <input type="button" value="Atrás"/> <input type="button" value="Siguiente"/> <input type="button" value="Cancelar"/> </p>
COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela Todos los derechos reservados	

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ► Presentación ► Documentos ► Fichas de proyectos ► Datos del proyecto usuario

Procesos

Proponer proyecto

Concebir proyecto

Documentos

Fichas de proyectos

Reportes

Proyectos

Fichas de proyectos

Datos del proyecto

Plan Operativo

Adicionar
 Modificar
 Eliminar

Nº	Actividad	Monto (USD)	Ejecutor	Receptor	Estado
1	Software	16,000.00	UCI	ST	Sin iniciar
2	Compra de equipos	4,000.00	UCI	ST	Sin iniciar
3	Construir edificios	0	UCI	ST	Sin iniciar
Subtotal		20,000.00			

Monto Total

< << Página 1 de 4 >> >
Mostrando 3 de 10

Atrás Finalizar Cancelar

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

Flujos Alternos

Flujo alternos a los pasos 2,5,8.

Acción del Actor	Respuesta del Sistema
a. El usuario oprime el botón “Cancelar” .	b. Elimina los datos.
	c. Muestra un mensaje de información “La acción

	ha sido cancelada”.
	d. Regresa a la vista anterior.
	e. El caso de uso termina.
Flujos alternos para Datos Incompletos	
Acción del Actor	Respuesta del Sistema
	a. Muestra el mensaje de información “Existen campos vacíos que son obligatorios, por favor, complete estos datos”.
	b. Muestra un indicador sobre los campos vacíos.
	c. Regresa al paso siguiente del Flujo Básico.
Sección “Modificar Propuesta de proyecto”	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra una pantalla con la ficha del proyecto correspondiente que contiene los datos generales del proyecto.</p> <ul style="list-style-type: none"> • Nombre del Proyecto. • Duración. • Modalidad. • Ministerio responsable venezolano. • Ente Ejecutor venezolano. • Ministerio responsable cubano. • Ente Ejecutor cubano.
2. Corrige los datos que considere necesarios y presiona el botón “ Siguiente ”.	3. Valida los datos.
	4. El sistema muestra una pantalla para

	<p>ingresar otros datos del proyecto.</p> <ul style="list-style-type: none">• Fundamentación.• Descripción breve del proyecto.• Objetivos del proyecto.• Objetivos específicos del proyecto.• Metas del proyecto.• Impacto del proyecto.• Población beneficiada.
5. Modifica los datos del proyecto que considere correcto y presiona le botón "Siguiente".	
	6. Valida la información.
7. Presiona el botón "Aceptar"	8. Modifica los datos contenidos en la propuesta de proyecto.
	9. El caso de uso termina
Prototipo de Interfaz	

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ► Presentación ► Documentos ► Fichas de proyectos ► Nuevo Proyecto ► Datos del proyecto usuario

Procesos

Proponer proyecto

Concebir proyecto

Documentos

Fichas de proyectos

Reportes

Proyectos

Fichas de proyectos

Datos del proyecto

Nombre del proyecto

Duración en meses

Modalidad

Equipamientos

Solución integral

Asistencia técnica

Suministros

Ministerio venezolano responsable

Ente ejecutor venezolano

Ministerio cubano responsable

Ente ejecutor cubano

Aceptar Siguiente Cancelar

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

Flujos Alternos

Flujo alternos a los pasos 2, 5, 8.

Acción del Actor	Respuesta del Sistema
a. El usuario oprime el botón “Cancelar”.	b. Elimina los datos.

	c. Muestra un mensaje de información “La acción ha sido cancelada”.
	d. Regresa a la vista anterior.
	e. El caso de uso termina.
Flujos alternos para Datos Incompletos	
Acción del Actor	Respuesta del Sistema
	a. Muestra el mensaje de información “Existen campos vacíos que son obligatorios, por favor, complete estos datos.”
	b. Muestra un indicador sobre los campos vacíos.
	c. Regresa al paso siguiente del Flujo Básico.
Sección “Eliminar Propuesta de Proyecto”	
Acción del Actor	Respuesta del Sistema
	1. Lanza una confirmación de eliminación preguntándole al usuario si realmente desea eliminar el proyecto en cuestión.
2. Presiona el botón “ Aceptar ”	3. Elimina la propuesta seleccionada, incluyendo el plan operativo asociado a la propuesta.
Prototipo de Interfaz	

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ▶ Presentación ▶ Documentos ▶ Fichas de proyectos usuario

Procesos

Proponer proyecto

- Otros vínculos
- Otros vínculos
- Otros vínculos

Concebir proyecto

Documentos

Fichas de proyectos

- Otros vínculos
- Otros vínculos
- Otros vínculos

Reportes

Proyectos

Fichas de proyectos

- Otros vínculos
- Otros vínculos
- Otros vínculos

Fichas de proyectos

Filtrar la búsqueda

Nombre del proyecto	Ente contraparte
<input type="text"/>	<input type="text"/>
Ministerio	Marco de aprobación
<input type="text"/>	<input type="text"/>

Nombre del proyecto	Ministerio	Contraparte	Ente Contraparte	Monto Total	Estado
Convenio Cuba-Venezuela	MENPET		FUNDELEC	1,400,000.00	Espera aceptación de ente venezolano
Servicios					Rechazado por ente venezolano
Servicios					Proyecto en elaboración
Servicios					Espera aceptación de ente cubano
Servicios de mecanización	MAT		Corporación venezolana agraria	2,000,000.00	Espera aceptación de ente venezolano

Mensaje de confirmación
 ¿Esta seguro que desea eliminar la propuesta de proyecto?

|< << Página 1 de 4 >> >|

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

Flujos Alternos

Flujo alternativo al paso 2.

Acción del Actor	Respuesta del Sistema
	a Detiene la eliminación regresando al paso anterior del flujo de normal de eventos.

Sección "Aceptar propuesta de proyecto"	
Acción del Actor	Respuesta del Sistema
1. Selecciona una propuesta de proyecto y presiona "Ver detalles".	2. Muestra una pantalla con los datos contenido en la ficha del proyecto.
3. Selecciona opción " Aceptar " del menú.	4. Cambia el estado de la propuesta a aceptada por ambos entes, pasando a ser un proyecto conjunto entre los entes.
Prototipo de interfaz	

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ► Presentación ► Procesos ► Proponer proyecto usuario

Procesos

Proponer proyecto

Concebir proyecto

Documentos

Fichas de proyectos

Reportes

Proyectos

Fichas de proyectos

Datos del proyecto [Aceptar](#) [Rechazar](#)

Nombre del proyecto	Registros y Notarías
Duración en meses	12
Modalidad	Integral
Invertir en Venezuela	9,000,000.00
Transferir a Cuba	16,000,000.00
Ministerio(s) por Venezuela	MPPIJ
Ente(s) ejecutor(es) por Venezuela	RN
Ministerio(s) por Cuba	UCI
Ente(s) ejecutor(es) por Cuba	ALBET
Fundamentación	Importante para el ALBA
Descripción breve del proyecto	
Objetivos del proyecto	Informatizar los registros y notarías
Objetivos específicos del proyecto	
Metas del proyecto	
Impacto del proyecto	
Población beneficiada	

[Siguiente](#) [Cancelar](#)

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados

Flujos Alternos

Flujo alternativo al paso 3.

Acción del Actor	Respuesta del Sistema
	a Muestra una pantalla con un cuadro de texto

	para que el ente argumente la causa o motivo del rechazo.
b Ingresa el comentario necesario y presiona el botón "Aceptar".	c Cambia el estado de la propuesta y envía la nota de rechazo al ente involucrado.

[LOGO]

[BANNER]

Presentación Contratación Seguimiento Configuración Administración Misiones Novedades Institución

Inicio ▶ Presentación ▶ Procesos ▶ Proponer proyecto usuario

Procesos

Proponer proyecto

Concebir proyecto

Documentos

Fichas de proyectos

Reportes

Proyectos

Fichas de proyectos

Datos del proyecto

[Aceptar](#) [Rechazar](#)

Nombre del proyecto	Registros y Notarías
Duración en meses	12

Nota de rechazo
✕

En esta área se introducirá una nota breve pero explicativa de el por qué se ha tomado una decisión dentro del proyecto.
 Esta nota será emitida cuando las decisiones tomadas afecten a un bien, servicio, a una o varias personas.

Enviar Cancelar

Descripción breve del proyecto	Informatizar los registros y notarías
Objetivos del proyecto	
Objetivos específicos del proyecto	
Metas del proyecto	
Impacto del proyecto	
Población beneficiada	

[Siguiente](#) [Cancelar](#)

COPYRIGHT © 2008, Convenio Integral de Cooperación Cuba Venezuela
 Todos los derechos reservados



Flujo alternativo al paso b

Acción del Actor	Respuesta del Sistema
	a Detiene el rechazo de la propuesta, regresando al paso anterior del flujo de eventos.
Poscondiciones	<ol style="list-style-type: none"> 1. La propuesta de proyecto debe quedar creada. 2. La propuesta de proyecto debe quedar modificada. 3. La propuesta de proyecto debe quedar eliminada. 4. La propuesta de proyecto debe quedar aceptada.

2.7 Subsistema de diseño

Los subsistemas de diseño constituyen una forma de estructurar los artefactos que conforman el modelo de diseño en estructuras más independientes. Los elementos del subsistema deben tener un alto grado de cohesión. El subsistema en su conjunto debe tener bajo acoplamiento con respecto a otros subsistemas, lo cual facilita su reutilización, pudiéndose convertir en componentes genéricos. Con el fin de separar los aspectos de diseño y lograr una mayor independencia en cuanto a desarrollo y a reutilización de funcionalidades, se identificaron los subsistemas relacionados con el módulo Presentación del sistema CCV.

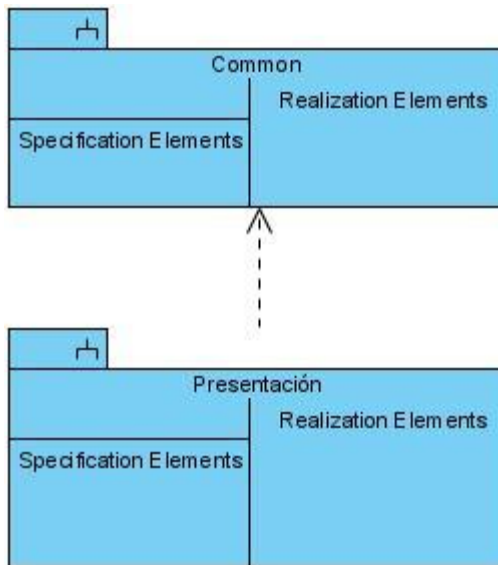


Figura 10 Subsistemas de diseño del Módulo Presentación del sistema CCV

2.8 Paquetes de diseño

El diseño de paquetes facilita la agrupación de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes relacionados de alguna manera. Constituye una división del subsistema en partes más manejables, las cuales no necesariamente tienen una interfaz definida. Con el objetivo de lograr una mejor organización y especialización de los elementos del diseño se identificaron y definieron los siguientes paquetes para el subsistema de presentación del módulo Presentación del sistema CCV.

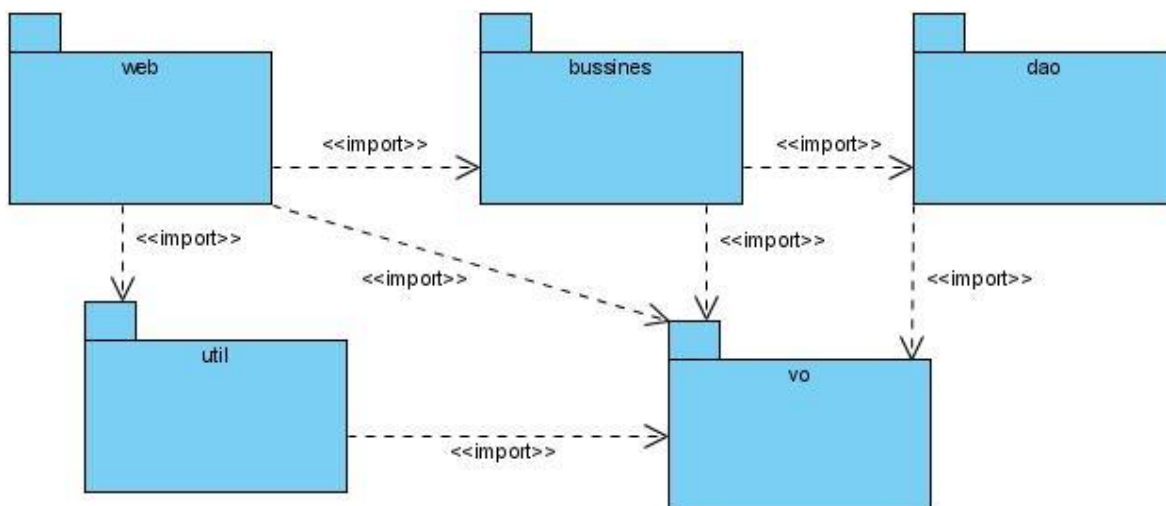


Figura 11 Paquetes de diseño del Módulo Presentación del sistema CCV

2.9 Diagrama de clases de diseño

Los diagramas de clases del diseño son los encargados de representar las relaciones entre clases, interfaces así como la colaboración entre ellos. Constituyen la vista del diseño estático de un sistema. Contienen las definiciones de las entidades de software y además de visualizar, estructurar y documentar los modelos estructurares, ayudan a construir el sistema a través de la ingeniería directa e inversa de código.

El sistema CCV por sus fines de uso se definió que debía ser una aplicación web, por tal razón en los diagramas de clases de diseño elaborados se hicieron uso de los estereotipos web, con el objetivo de ilustrar la colaboración real y la representación de todos los elementos que interactúan en la ejecución de las funcionalidades que debe brindar el sistema.

A continuación se muestran los diagramas de clases de diseño de los casos de uso principales del módulo Presentación del sistema CCV, el resto de los diagramas de clases del diseño se pueden consultar en el Documento referente al modelo de diseño del módulo Presentación del sistema CCV (Zayas Heras, et al., 2009).

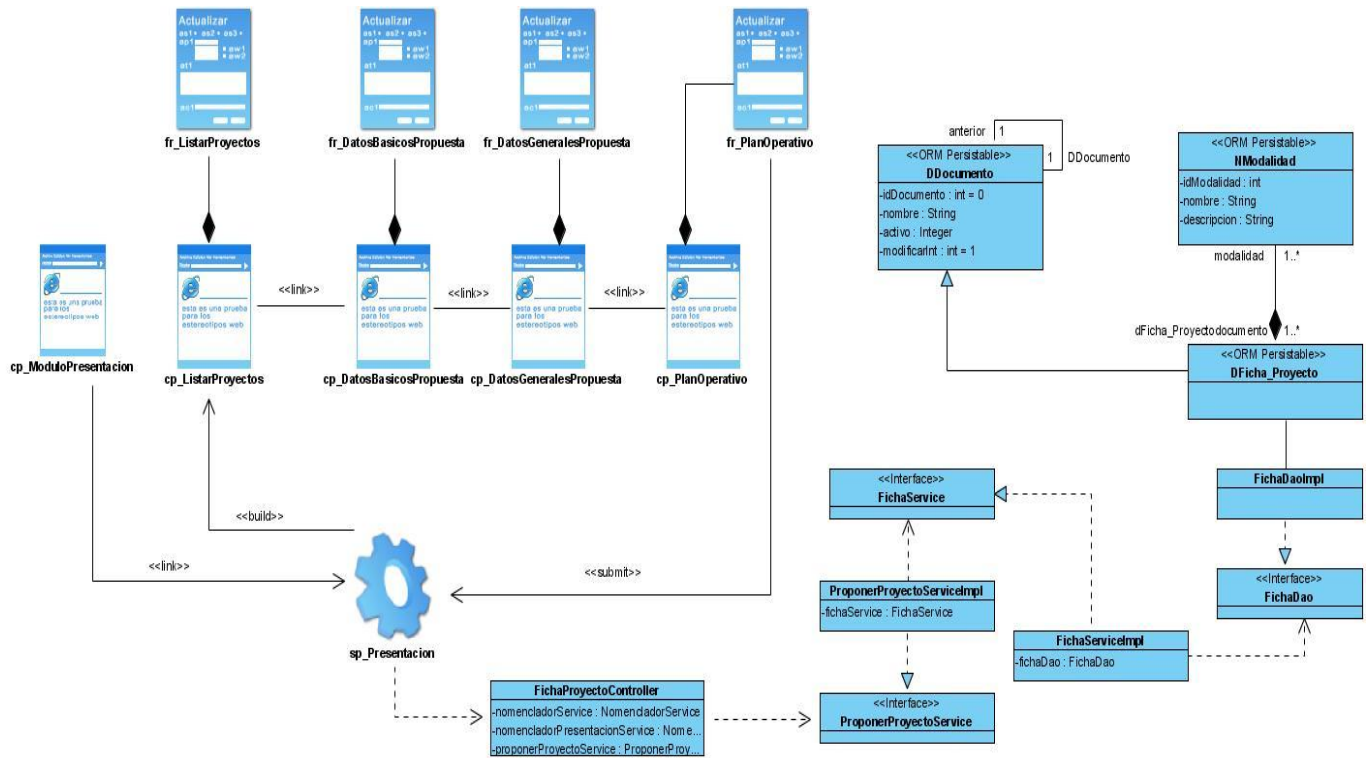


Figura 12 Diagrama de clases de diseño, CU Gestionar Propuesta de Proyecto.

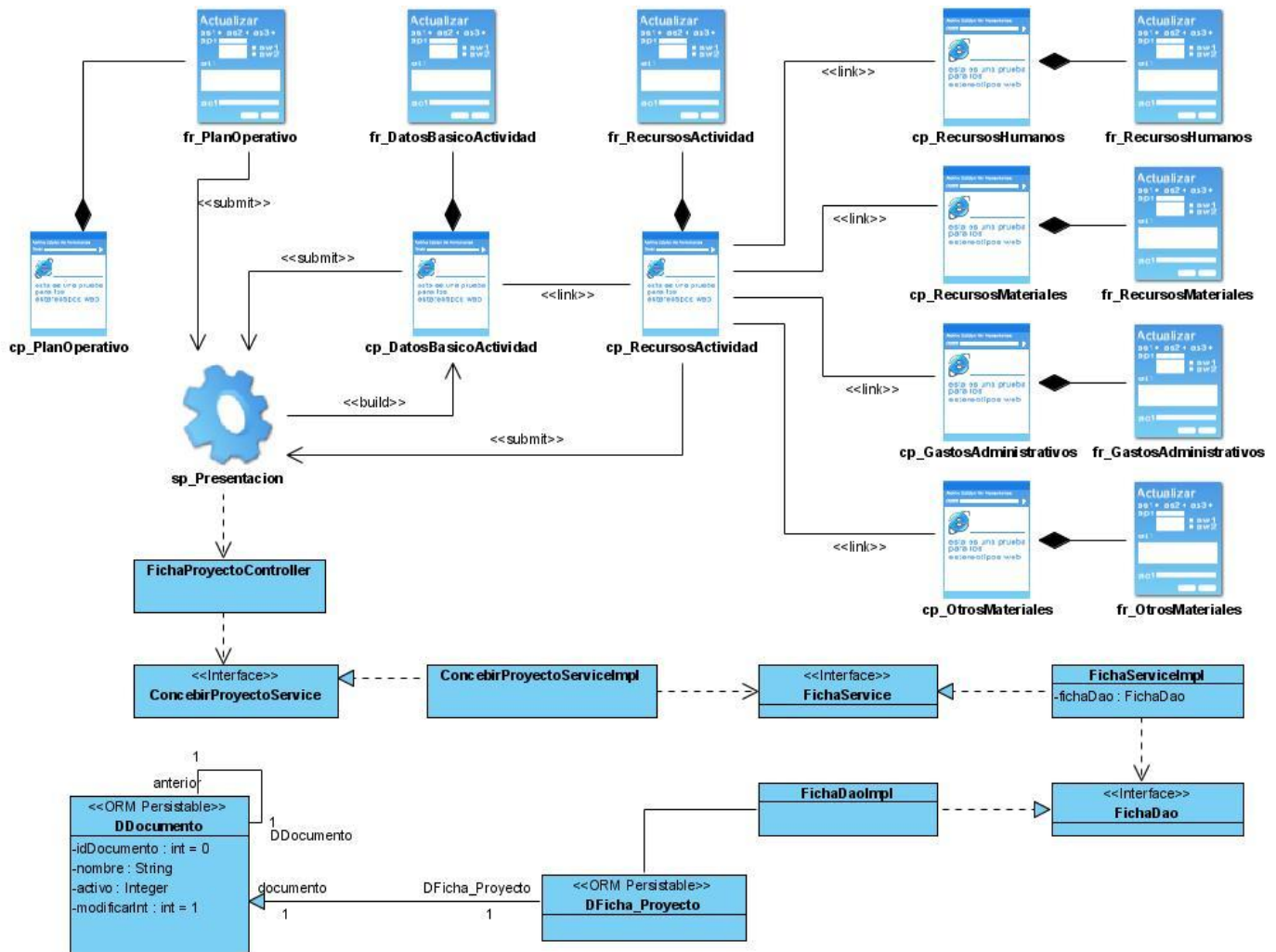


Figura 13 Diagrama de clases de diseño, CU Gestionar Plan Operativo.

2.10 Diagramas de secuencia

Para una mejor representación y entendimiento de los flujos de los casos de uso definidos, se elaboraron los diagramas de secuencia como parte de la realización de cada uno de estos. Los diagramas de secuencia muestran las interacciones entre los objetos que estarán presentes en la aplicación, ordenadas en secuencia temporal durante un escenario concreto. A los casos de uso que tienen varias secciones se le realizó un diagrama de secuencia por cada flujo para un mejor entendimiento de los mismos.

A continuación se muestran los diagramas de secuencia elaborados para los escenarios de los casos de uso principales del módulo Presentación del sistema CCV, el resto de los diagramas de secuencia del diseño se pueden consultar en el Documento referente al modelo de diseño del módulo Presentación del sistema CCV (Zayas Heras, y otros, 2009).

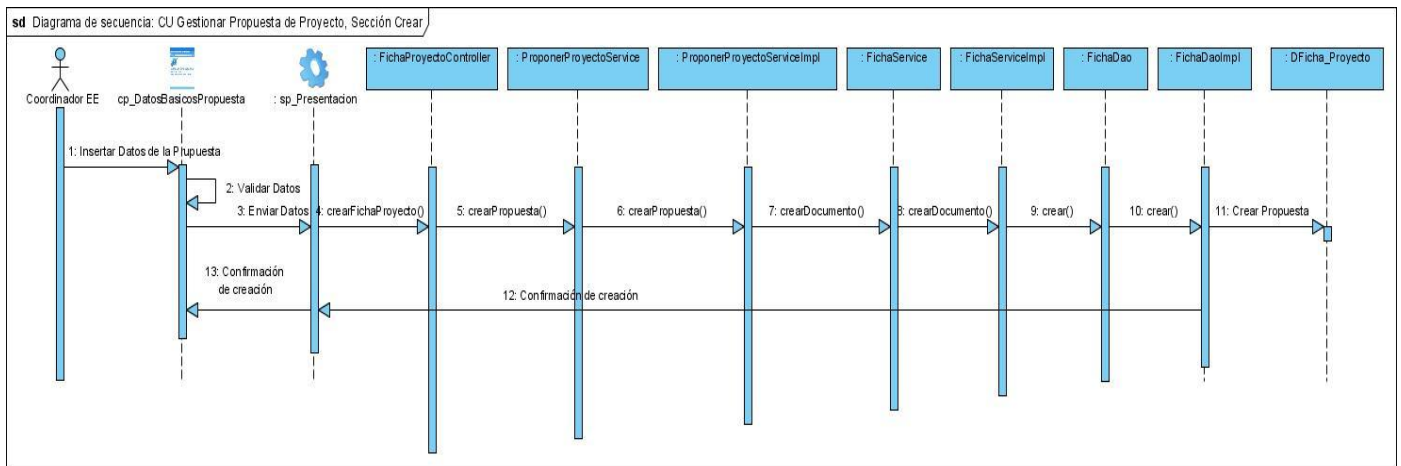


Figura 14 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Crear.

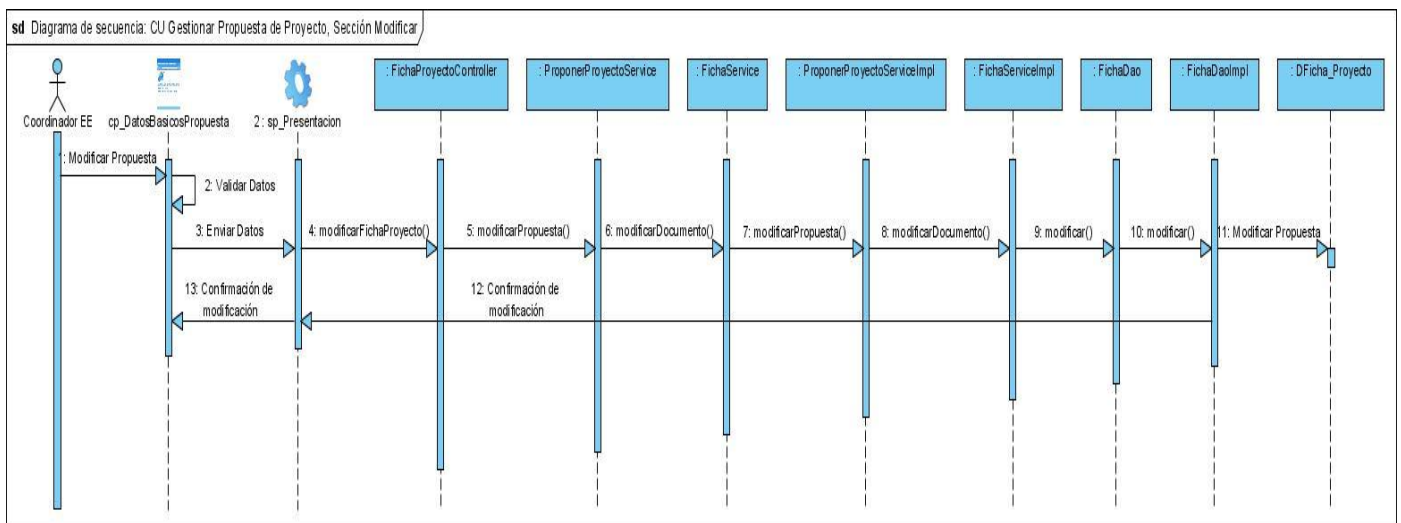


Figura 15 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Modificar.

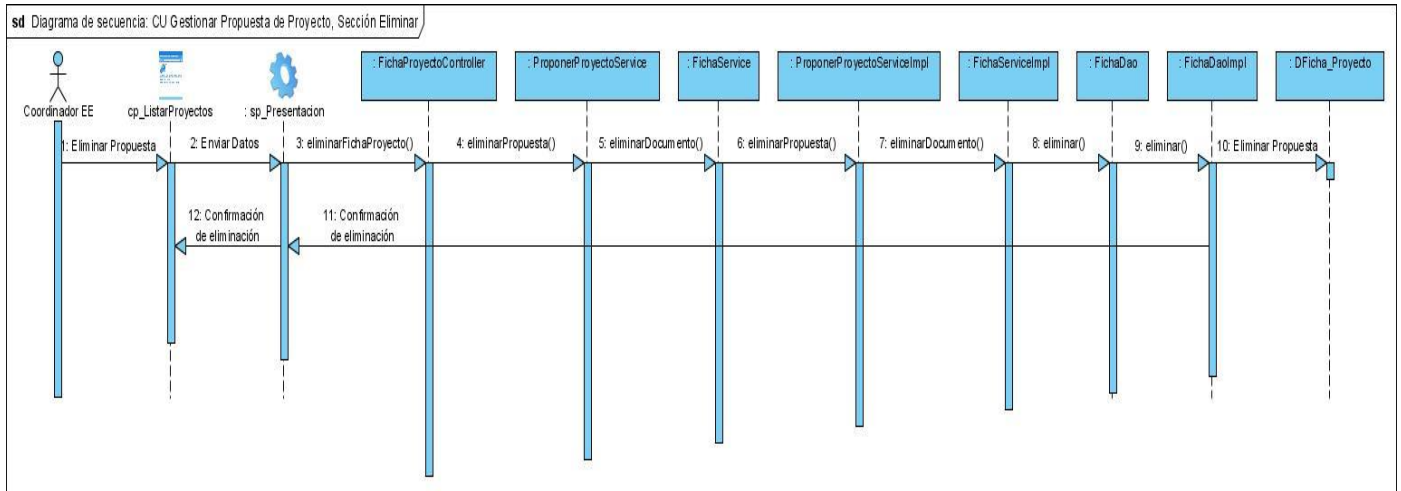


Figura 16 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Eliminar.

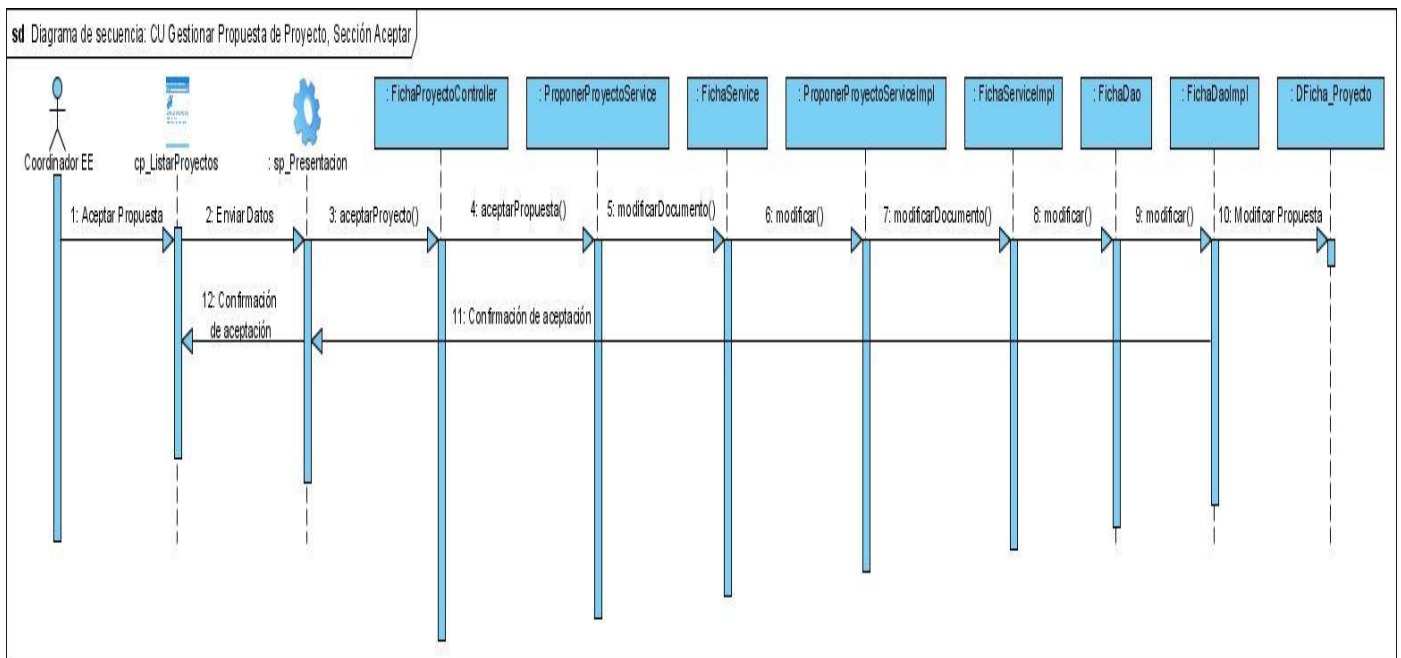


Figura 17 Diagrama de secuencia: CU Gestionar Propuesta de Proyecto, Sección Aceptar.

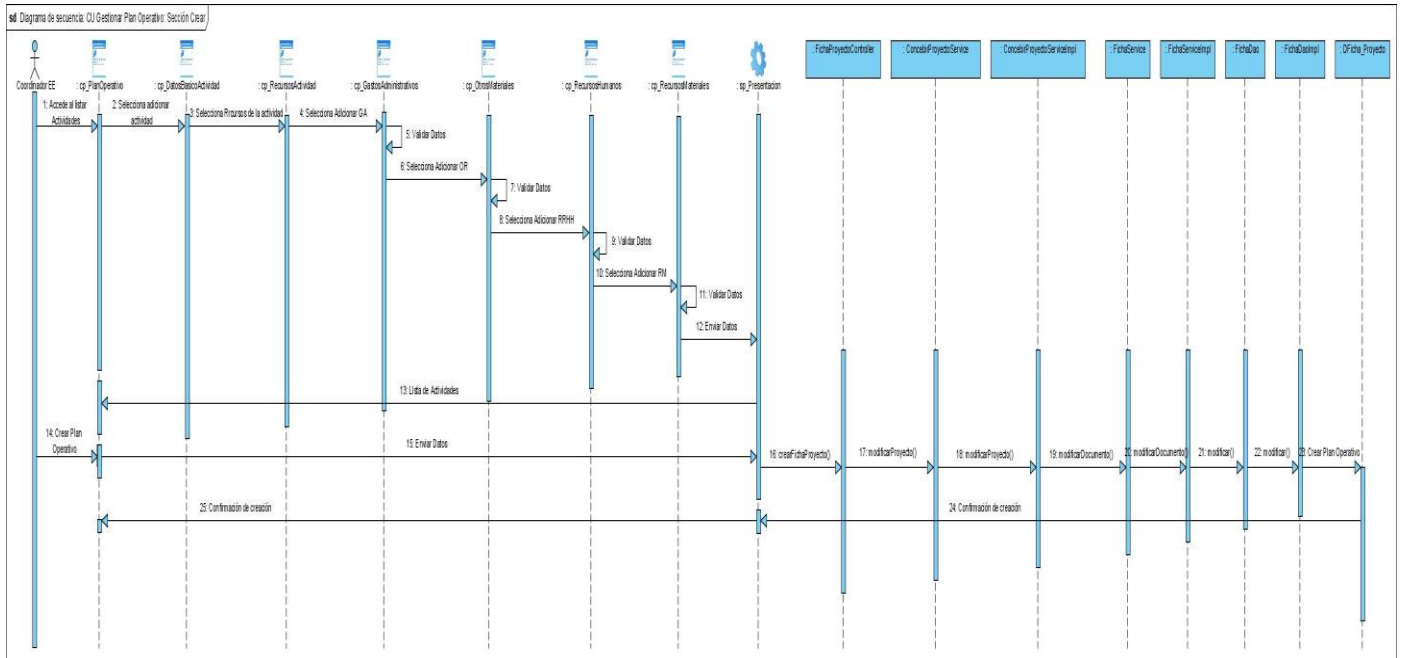


Figura 18 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Crear.

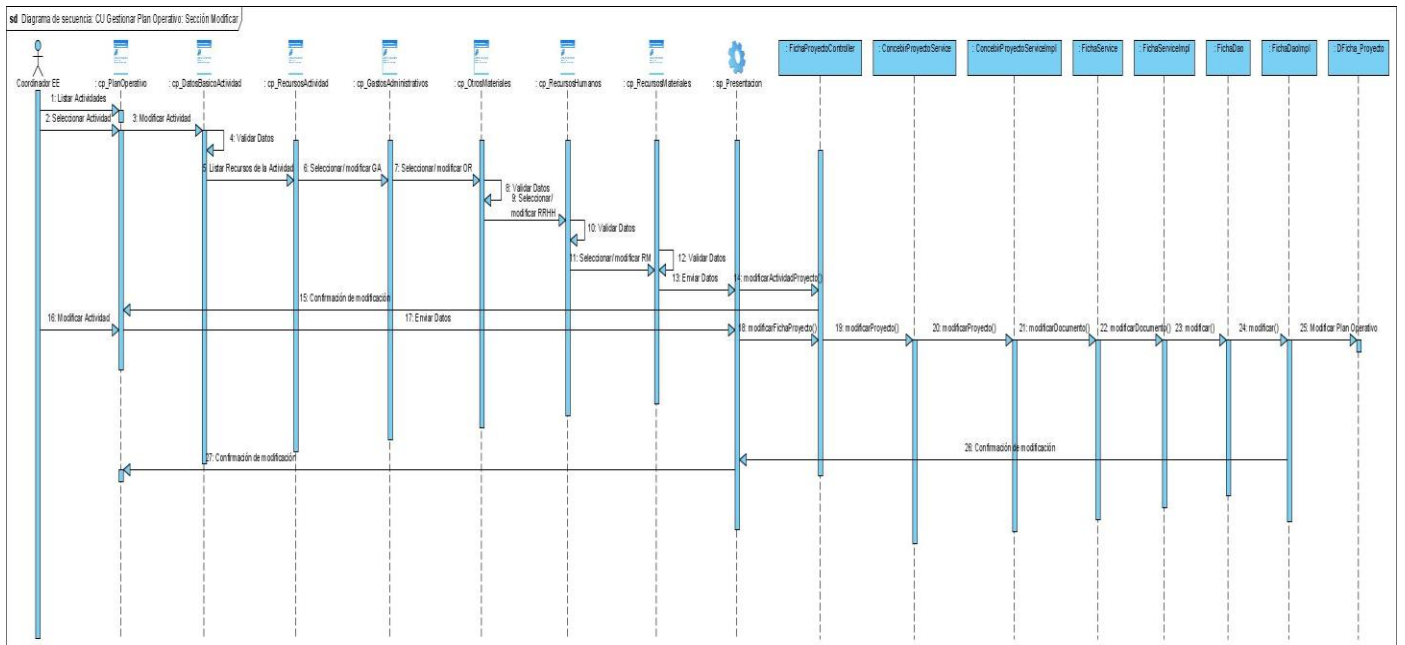


Figura 19 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Modificar.

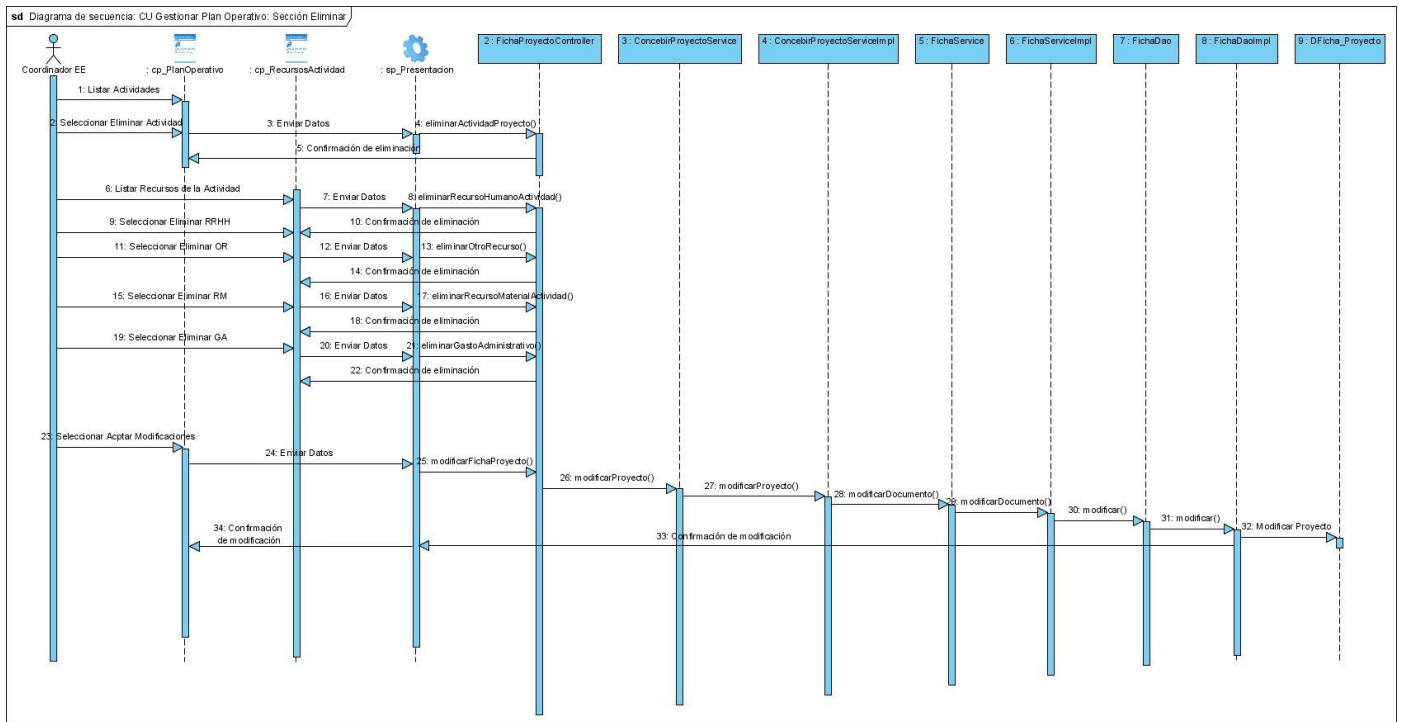


Figura 20 Diagrama de secuencia: CU Gestionar Plan Operativo, Sección Eliminar.

2.11 Diagrama de clases persistentes

Para facilitar la persistencia de los datos y para poder generar la base de datos y los objetos persistentes junto a sus ficheros de configuración que necesita el *hibernate* para establecer el mapeo de estos objetos con las tablas de la base de datos se generó el diagrama de clases persistentes. Este diagrama representa la relación de los objetos persistentes con sus respectivas relaciones, dependencias y cardinalidades. La sincronización del modelo de entidad relación con este diagrama quedó de la siguiente forma.

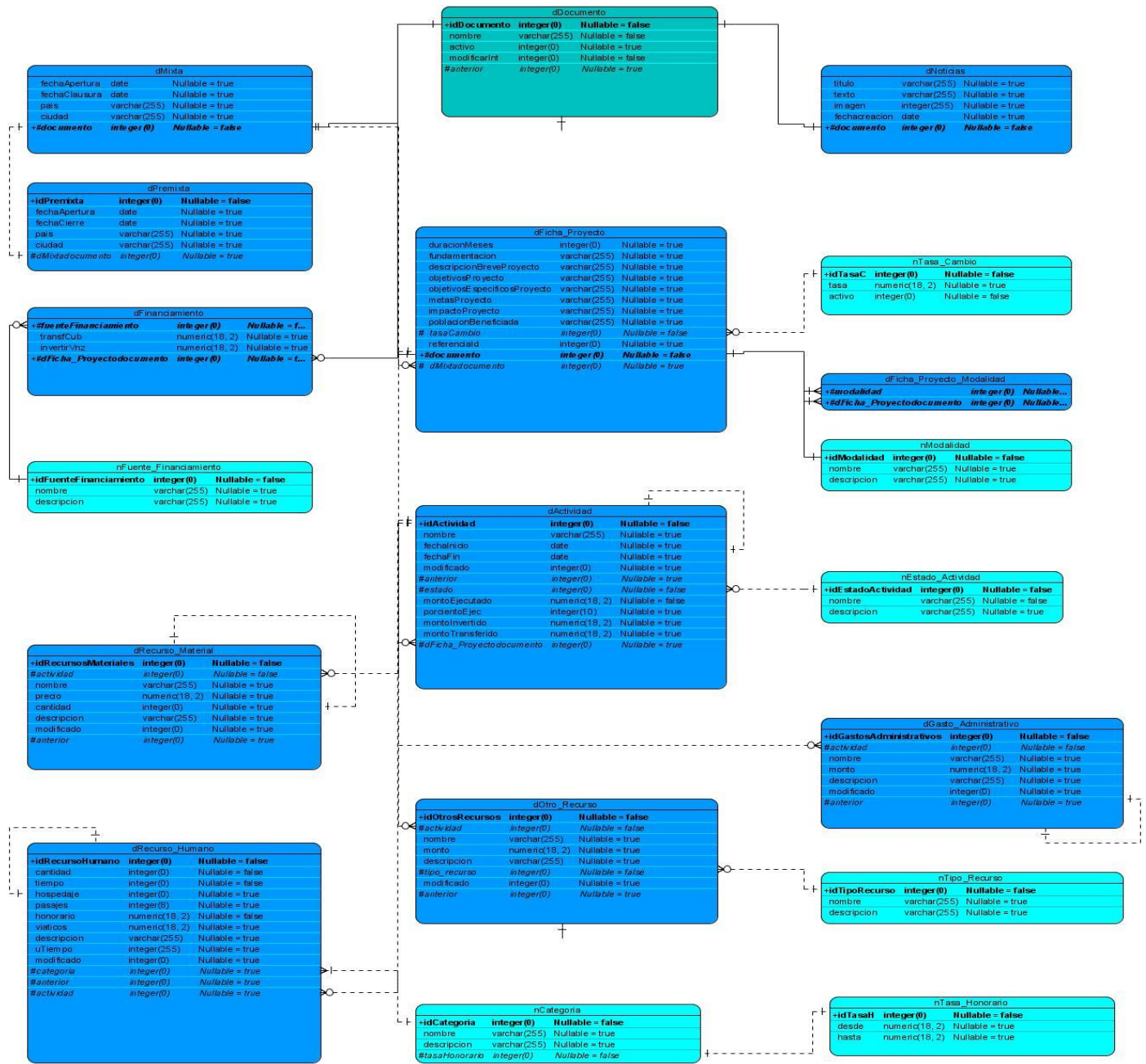


Figura 22 Modelo Entidad Relación.

2.13 Conclusiones

En este capítulo se analizaron las particularidades referentes al sistema CCV específicamente del módulo Presentación, arribándose a las siguientes conclusiones:

- El análisis de los procesos de negocio, junto a la interacción con los clientes del sistema y aplicando técnicas para la elicitación de requisitos, propició que se obtuvieran resultados satisfactorios en la identificación de los requerimientos, tanto funcionales como no funcionales que debe cumplir el sistema.
- La identificación y especificación de los artefactos del modelo del sistema facilitó un mayor entendimiento y un acuerdo común entre los clientes y los desarrolladores, en cuanto a la concepción de las funcionalidades que el sistema debe cumplir.
- La modelación de los artefactos, tanto del análisis como del diseño, utilizando visual Paradigm y UML como lenguaje de modelado, propició un mayor entendimiento entre los involucrados en el módulo Presentación del sistema CCV.
- La construcción de los artefactos del modelo de diseño permitió obtener cada uno de los elementos que soportan los requerimientos funcionales y no funcionales del sistema, siendo estos la entrada a las actividades de implementación.
- La aplicación de patrones, tanto de casos de uso como de diseño propiciaron obtener artefactos aceptables.

3.1 Introducción

El presente capítulo muestra los procedimientos y métodos utilizados para medir la factibilidad de los artefactos obtenidos, tanto en el modelo de sistema como en el de diseño. Para la validación de los artefactos del modelo del sistema se muestran las listas de chequeo que se le fueron aplicados, además de las actas legales de liberación a diferentes niveles, mientras que para la validación de los artefactos del modelo del diseño se aplican métricas a nivel de componentes y del tamaño de clases. Garantizar la calidad de los artefactos obtenidos en cada fase ayuda minimizar los errores y obtener un producto con mayor calidad que cumpla con todas las expectativas del cliente. He ahí la importancia del análisis de los resultados obtenidos en cada etapa del proceso de desarrollo de software.

3.2 Validación del modelo del sistema

Los requerimientos identificados con el cliente se especificaron en el artefacto llamado “Especificación de Requisitos”, el cual junto al “Modelo del sistema”, artefacto producto del análisis de los requerimientos e identificación de casos de uso, actores y descripción de ambos, constituyen los principales artefactos obtenidos durante el análisis del módulo Presentación del sistema CCV. Con el objetivo de garantizar su factibilidad, fueron sometidos a varias iteraciones de revisiones por parte de calidad a nivel de proyecto, facultad y universidad. Para la prueba de estos artefactos se aplicaron listas de chequeo para los requerimientos (**ver Anexo 1**) y para la descripción de los CU (**ver Anexo 2**). Durante las iteraciones de la revisión se identificaron no conformidades, las cuales fueron solucionadas por el equipo de análisis del proyecto. Como resultado se obtuvo la liberación de la documentación por parte del equipo de calidad de la universidad, como constancia de la factibilidad de los artefactos analizados.

Luego de liberados los artefactos anteriormente mencionados, fueron presentados a los clientes del sistema, los cuales procedieron a su revisión, con el objetivo de verificar que la especificación mostrada cumpliera con todas las expectativas, necesidades, condiciones y propiedades que esperaban con el desarrollo del módulo Presentación del sistema CCV. Los artefactos presentados fueron aceptados,

quedando de esta forma legalmente firmadas en el acta de aceptación (**ver Anexo 3**), de la “Especificación de requisitos” y del “Modelo del sistema” del módulo Presentación del sistema CCV, convirtiendo estos artefactos en compromisos de obligatorios cumplimiento en lo que a desarrollo de funcionalidades de la aplicación se refiere, por parte del equipo de desarrollo, para con los clientes del sistema.

Como constancia de haber realizado un adecuado proceso de identificación y especificación de requerimientos y posteriormente un adecuado análisis del sistema se muestra el acta de aceptación del módulo Presentación (**ver Anexo 4**) por parte del cliente y el acta de finalización oficial del proyecto (**ver Anexo 5**), lo cual plasma la satisfacción del cliente con el sistema CCV.

3.3 Métricas de diseño a nivel de componentes

3.3.1 Métrica de acoplamiento

Para medir el nivel de conectividad y dependencia del módulo Presentación con otros módulos se aplica la medida propuesta por Dhama (ver Capítulo 1), con el fin de determinar el nivel de independencia y reutilización que posee el módulo según el diseño propuesto.

Definiendo:

Acoplamiento de flujos de datos de control:

$d_i = 2$ (número de parámetros de datos de entrada)

$c_i = 1$ (número de parámetros de control de entrada)

$d_0 = 1$ (número de parámetros de datos de salida)

$c_0 = 1$ (número de parámetros de control de salida)

Acoplamiento global:

$g_d = 0$ (número de variables globales usadas como datos)

$g_c = 0$ (número de variables globales usadas como control)

Acoplamiento de entorno:

$w = 2$ (número de módulos llamados **Figura 23**)

$r = 4$ (número de módulos que llaman al módulo Presentación **Figura 24**)

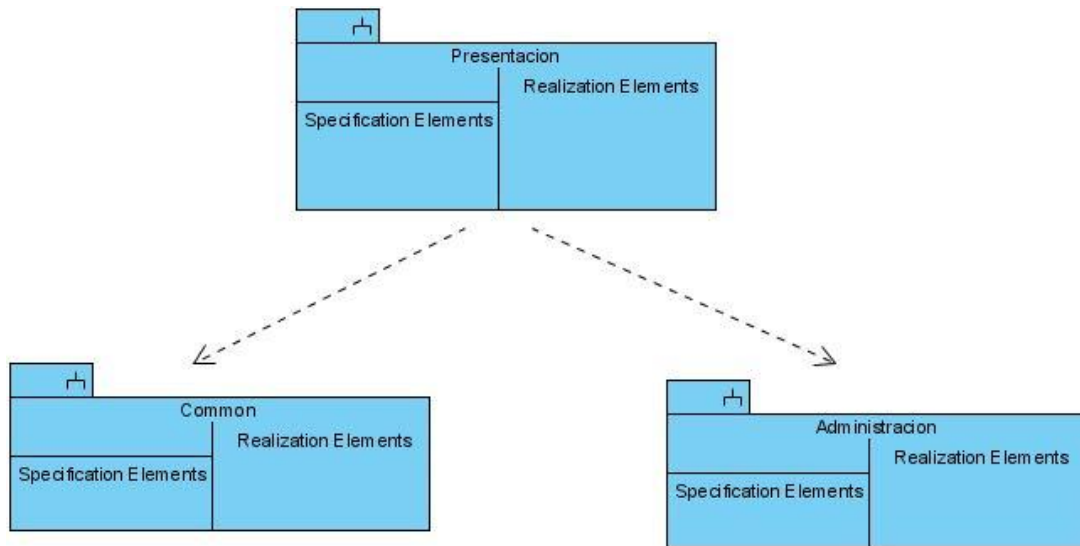


Figura 23 Expansión del módulo Presentación.

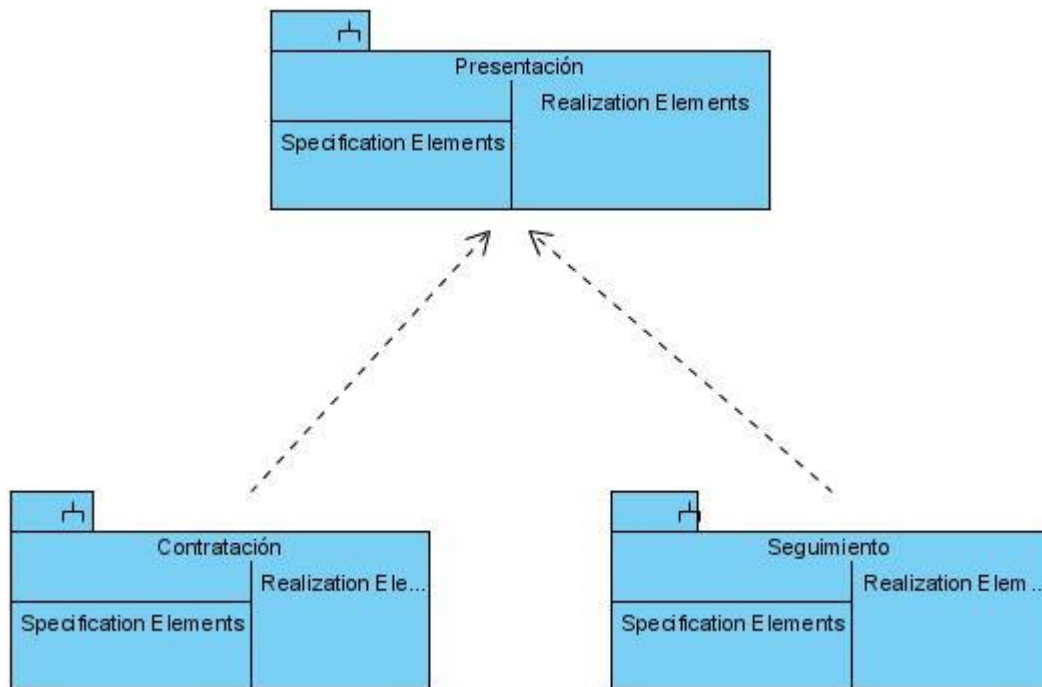


Figura 24 Concentración hacia el módulo Presentación.

Indicador de acoplamiento del módulo:

$$m_c = K / M$$

Donde $K = 1$, constante de proporcionalidad según el autor

$$M = d_i + aXc_i + d_0 + bXc_0 + g_d + cXg_c + w + r$$

$$a=b=c=2$$

$$m_c = 1 / (2 + 2X1 + 1 + 2X1 + 0 + 2X0 + 2 + 2)$$

$$m_c = 0.09$$

El valor de 0.09 del indicador de acoplamiento del módulo Presentación nos sugiere el grado de acoplamiento del mismo, lo que sugiere que existe dependencia con otros módulos para su funcionamiento, pero que la misma es mínima y no repercute por el hecho de depender de módulos básicos para el funcionamiento del sistema CCV.

3.3.2 Métrica de cohesión

Una buena práctica para el diseño lo constituye el lograr una alta cohesión, como resultado de aplicar los patrones GRASP en la construcción del mismo. Para medir el índice de cohesión de los elementos que conforman el diseño del módulo Presentación se aplicó la métrica de Bieman y Ott, la cual se detalla en el (ver Capítulo 1).

Para ello se analizaron elementos pertenecientes a los principales conceptos que plantea la métrica, dígame Porción de datos, Símbolos léxicos (tokens) de datos, Señales de unión, Señales de súper-unión y Cohesión. Para proceder al análisis de cada uno de estos conceptos se recogieron los datos que ilustran el nivel de uso de las principales clases diseñadas para el módulo Presentación (ver **Tabla 5**).

Tabla 5 Usabilidad de las clases.

Clases	Usabilidad
1.FichaDaolmpl	6
2.MixtaDaolmpl	1
3.NoticiaDaolmpl	1
4.FinanciamientoDaolmpl	1
5.FichaServiceImpl	6

6.MixtaSeviciImpl	1
7.NoticiaServiceImpl	1
8.ProponerProyectoServiceImpl	1
9.ConcebirProyectoServiceImpl	1
10.PresentarProyectoServiceImpl	1
11.EvaluarProyectoStServiceImpl	1
12.AbrirPeriodoPresentacionServiceImpl	1
13.CerrarPeriodoPresentacionServiceImpl	1
14.PresentacionController	0
15.FichaProyectoController	0
16.MixtaController	0
17.NoticiaController	0

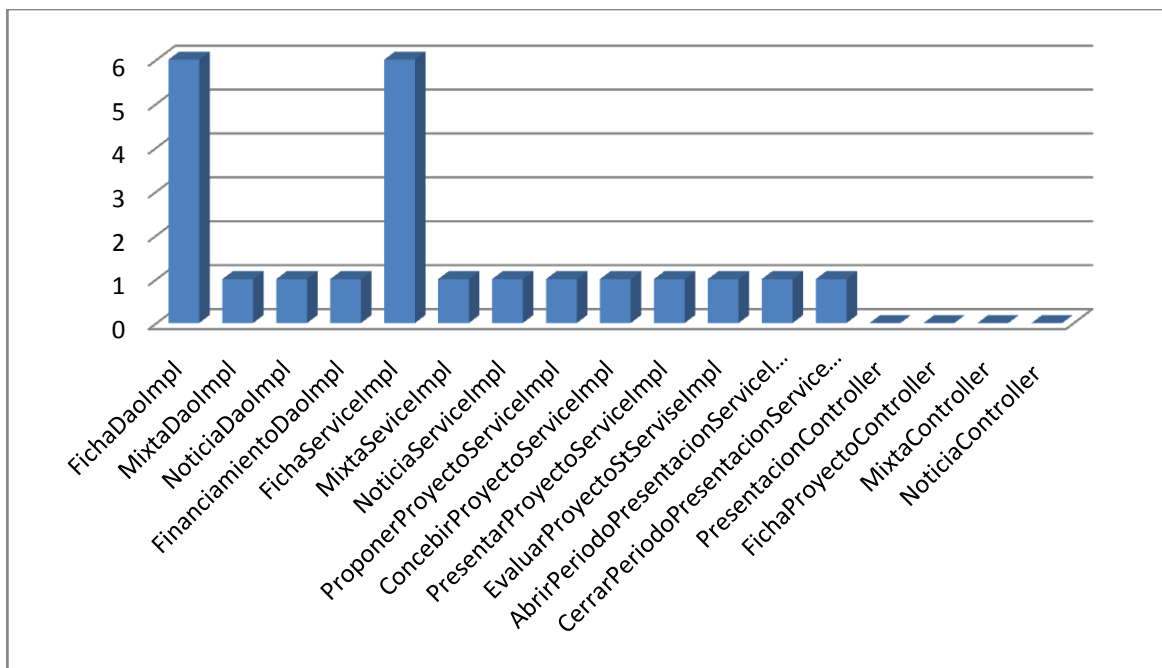


Figura 25 Relaciones de uso de las clases.

La cohesión funcional se determina con dos enfoques:

1. Determinando la cohesión funcional fuerte (CFF) y la pegajosidad: se obtienen cuando el resultado de la métrica es de 1.

Se define como:

$$\text{CFF} = \text{número de súper adhesivos (i)} / \text{número de elementos (i)}$$

2. Determinando la cohesión funcional débil(CFD):

Se define como:

$$\text{CFD} = \text{número de adhesivos (i)} / \text{número de elementos (i)}$$

Adhesivo. Se le llamará adhesivo a un elemento que aparece en dos o más rebanadas.

Súper adhesivo. Se denomina súper adhesivo a un elemento que está en todos los elementos de un módulo.

(i) Se define como la muestra.

Según los datos de las clases analizadas se tiene que:

número de elementos = 0

número de súper adhesivos(i) = 0

número de adhesivos(i) = 13

$$\text{CFD} = \text{número de adhesivos (i)} / \text{número de elementos (i)}$$

$$\text{CFD} = 13 / 17$$

$$\text{CFD} = 0.76$$

$$\text{CFF} = \text{número de súper adhesivos (i)} / \text{número de elementos (i)}$$

$$\text{CFF} = 0 / 17$$

$$\text{CFF} = 0$$

La métrica de Bieman y Ott plantea que mientras más cerca están los valores de CFF y CFD de 1 mayor será la cohesión del módulo. Los resultados demuestran que no hay una cohesión funcional fuerte, pero la relación del número de clases adhesivas con el número total de elementos de la muestra, determinados por la $\text{CFD} = 0.76$, está cercana a 1, lo que demuestra que el diseño de las clases del módulo Presentación posee una cohesión funcional con un 76% de fortaleza.

3.4 Métricas orientadas a clases. Tamaño de clase (TC)

Con el objetivo de comprobar el adecuado diseño de las clases y el nivel de reutilización de las mismas se aplicó la métrica del TC (ver Capítulo 1).

Se aplicará esta métrica para las principales clases de las tres capas definidas en el módulo.

Clases	Nº. de atributos	Nº. de operaciones
1.FichaDaolmpl	0	22
2.MixtaDaolmpl	0	7
3.NoticiaDaolmpl	0	13
4.FinanciamientoDaolmpl	0	6
5.FichaServiceImpl	1	19
6.MixtaSevicelmpl	2	10
7.NoticiaServiceImpl	3	21
8.ProponerProyectoServiceImpl	1	9
9.ConcebirProyectoServiceImpl	1	13
10.PresentarProyectoServiceImpl	1	7
11.EvaluarProyectoStServiseImpl	2	15
12.AbrirPeriodoPresentacionServiceImpl	1	9
13.CerrarPeriodoPresentacionServiceImpl	1	5
14.PresentacionController	0	1
15.FichaProyectoController	14	32
16.MixtaController	5	20
17.NoticiaController	4	13

Se presentó un **total de 17 clases** para un **promedio de atributos de 2.11** y un **promedio de operaciones de 13.06**.

De esta forma el umbral queda con los datos mostrados a continuación:

Umbral	Tamaño	Cantidad de clases
<=20	Pequeño	14

>20<=30	Medio	2
>30	Grande	1

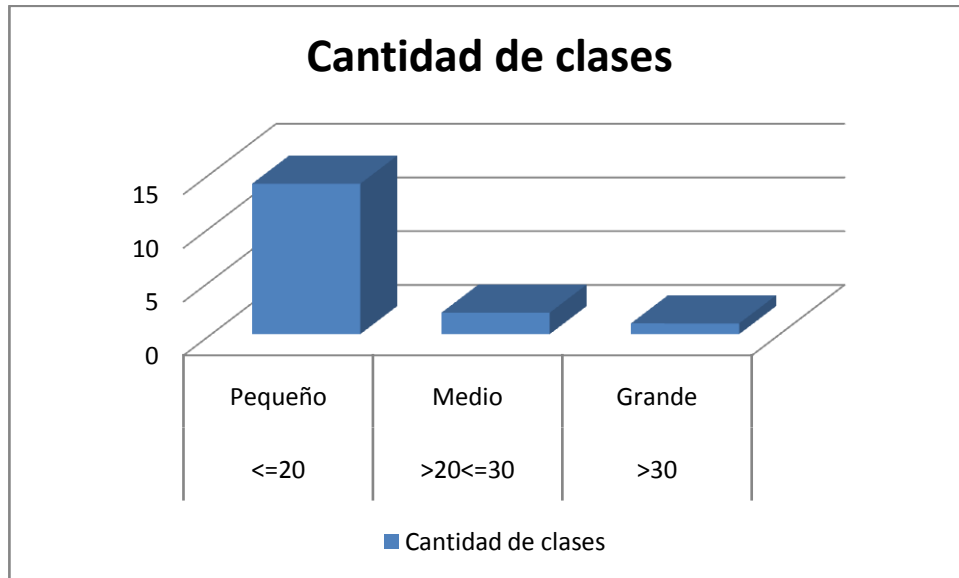


Figura 26 Número de clases por categorías.

3.5 Conclusiones

En este capítulo se analizaron los resultados obtenidos durante el desarrollo del trabajo, es decir, se evaluaron los artefactos que componen el modelo del sistema y los que componen el modelo del diseño, arribándose a las siguientes conclusiones:

- Las actas de aceptación y liberación por parte del equipo de calidad de la universidad y por los clientes del sistema demostraron que tanto los requerimientos como la especificación de los casos de uso del sistema quedaron bien definidos, especificados y sin ambigüedad, abarcando las necesidades del cliente.
- El indicador de acoplamiento del módulo Presentación determinó que existe dependencia con otros módulos para su funcionamiento, pero la misma es mínima y no repercute en el funcionamiento, puesto que depende únicamente de los módulos básicos del sistema CCV.

- El diseño de las clases del módulo Presentación posee una cohesión funcional con un 76% de fortaleza, lo que demuestra que sus elementos están altamente cohesionados.
- El 82 % de las clases diseñadas están consideradas como pequeñas, lo que facilitara el proceso de construcción del módulo Presentación.
- Las métricas aplicadas a los elementos del diseño del módulo Presentación validan la calidad del diseño elaborado.

Conclusiones Generales

Al concluir el presente trabajo se arriba a las siguientes conclusiones:

- El análisis de los procesos de negocio ya elaborados, junto a la interacción con los clientes del sistema y aplicando técnicas para la elicitación de requerimientos, propició que se obtuvieran resultados satisfactorios en la identificación de los requerimientos que debe cumplir el sistema.
- La especificación de los artefactos del modelo del sistema facilitó un mayor entendimiento y un acuerdo común entre los clientes y los desarrolladores, con respecto a las funcionalidades que el sistema debe brindar.
- La construcción de los artefactos del modelo de diseño permitió obtener los elementos que deberán ser implementados para obtener un sistema que cumpla con los requerimientos especificados por el equipo de análisis y aprobados por los clientes.
- El análisis de los artefactos obtenidos en el modelo del sistema y el de diseño permitió medir el grado de factibilidad de estos en función de la calidad, funcionalidad, ambigüedad, reutilización y consistencia.

Recomendaciones

Con vista al desarrollo del módulo Presentación y posteriores mejoras del mismo, se recomienda:

- Elaborar los artefactos restantes pertenecientes al diseño, que son necesarios para continuar con el desarrollo del módulo y que no lo realiza el diseñador.
- Desarrollar la implementación del módulo Presentación del sistema CCV.
- Aplicar ingeniería inversa de código utilizando Visual Paradigm, para mantener actualizado los diagramas de clases del diseño.
- Realizar la administración de requerimientos.

Bibliografía

- Cortizo Pérez, José Carlos, Expósito Gil, Diego y Ruiz Leyva, Miguel. 2004.** *eXtreme Programming*. España : s.n., 2004.
- Durán Toro, Amador y Bernárdez Jiménez, Beatriz. 2000.** *Metodología para la Elicitación de Requisitos de Sistemas Software*. Sevilla : s.n., 2000.
- GSIInnova. 2007.** Grupo de soluciones GSIInnova. *Grupo de soluciones GSIInnova*. [En línea] GSIInnova, 2007. <http://www.rational.com.ar/herramientas/roseenterprise.html>.
- IBM. IBM. IBM.** [En línea] IBM. [Citado el: 29 de enero de 2009.] <http://www-01.ibm.com/software/rational/>.
- IEEE Standards Association. 2004.** IEEE Standards Association. *IEEE Standards Association*. [En línea] 2004. http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 1999.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 1999.
- Larman, Craig. 1999.** *UML y Patrones, introduccion al análisis y diseño orientado a objetos*. 1999.
- Microsoft.** Microsoft. [En línea] <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/planificacion/msf.msp>.
- MITOPENSOURCEWARE Universia. 2001.** MITOPENSOURCEWARE Universia. *MITOPENSOURCEWARE Universia*. [En línea] Massachusetts institute of technology, 10 de octubre de 2001. <http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.
- Övergaard, Gunnar y Palmkvist, Karin. 2004.** *Use Cases Patterns and Blueprints*. s.l. : Addison Wesley Professional, 2004.
- Pressman, Roger S. 2005.** *Ingeniería de Software, un enfoque práctico*. 2005.
- Rational Software, Corporation. 2003.** *RUP Help*. s.l. : Rational Software Corporation, 2003.
- Ruiz, Software Architect Alvaro Ernesto Carmona. 2005.** *De los patrones de análisis y de integración a los componentes de negocio*. [Porwer Point] Bogotá, Colombia : Heinsohn Software House S.A., 2005.

- SPARX System Pty L tb. 2000-2007.** SPARX System. *SPARX System*. [En línea] SPARX System, 2000-2007. <http://sparxsystems.com.ar/products/ea.html>.
- UDELAB. 2009.** UDELAB. *Universidad de las Americas Puebla*. [En línea] 2009. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo4.pdf.
- Universidad Simon Bolivar. 1999.** LISI Laboratorio de informacion en sistemas informaticos . *LISI Laboratorio de informacion en sistemas informaticos*. [En línea] 1999. http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion_23.pdf.
- Villalón Cruzata, Martin y Montane Izaquirre, Juan Carlos. 2008.** *Modelo de casos de uso del sistema, Módulo Presentación, Sistema CCV*. La Habana : s.n., 2008.
- Visual Paradigm International. 2005.** Visual Paradigm. *Visual Paradigm*. [En línea] 2005. <http://www.visual-paradigm.com/product/vpuml/>.
- Wesley, Addison, Gamma, e., Helm, R., Johnson, R., Vlissides, J. 2003.** *Design Patterns. Elements of Reusable Object-Oriented*. 2003.
- World Wide Web Consortium. 1994.** W3C World Wide Web Consortium. [En línea] 1994. <http://www.w3c.org>.
- Zayas Heras, Yoilan y Villalón Cruzata, Martin. 2009.** *Modelo de Diseño, Módulo Presentación, Sistema CCV*. 2009.

Anexos

Anexo 1: Lista de chequeo para la especificación de requisitos.

Evaluación
Claridad
¿Los requisitos se escriben en lengua comprensible para el usuario/cliente?
¿Hay requisitos que tienen más de una interpretación?
¿Cada requisito característico del producto final se describe con una terminología única?
¿Hay un glosario en el cual los requisitos significativos y específicos están en términos definidos y claros?
¿Se entienden los requisitos para ponerlos en ejecución por un grupo independiente?
Completo
¿El requisito tiene un contenido específico?
Están especificados los cambios posibles a los requisitos.
La probabilidad de cambios está especificada para cada requisito.
¿Se definen todos los términos en cada uno de los requisitos determinados?
¿Tiene el artefacto un índice bien definido?
¿Existen áreas donde está incompleta la información debido a que el desarrollo aún no ha comenzado a especificarse?

¿La información que falta se define en el requisito?

¿Algún requisito necesita una especificación detallada?

¿Algún requisito necesita ser menos especificado?

¿Todos los requisitos se describen ellos mismos?

¿Están incluidos todos los requisitos relacionados con la funcionalidad?

¿Hay requisitos que produzcan inquietud?

¿Están incluidos todos los requisitos relacionados con el funcionamiento?

¿Están incluidos todos los requisitos relacionados con sus cualidades?

¿Están incluidos todos los requisitos relacionados con las interfaces externas?

¿Están incluidos todos los requisitos relacionados con las bases de datos?

¿Están incluidos todos los requisitos relacionados con el software?

¿Están incluidos todos los requisitos relacionados con el hardware? (No todavía)

¿Están incluidos todos los requisitos relacionados con las entradas de datos?

¿Están incluidos todos los requisitos relacionados con las salidas datos?

¿Están incluidos todos los requisitos relacionados con los mensajes de error?

¿Están incluidos todos los requisitos relacionados con la seguridad? (No todavía)
¿Están incluidos todos los requisitos relacionados con la capacidad de mantenimiento del software? (No todavía)
¿Están incluidos todos los requisitos relacionados con la instalación? (No todavía)
Consistencia (Duda)
¿Hay requisitos que describen el mismo objeto que estén en conflicto con otros requisitos con respecto a la terminología?
¿Hay requisitos que describen al mismo objeto que estén en conflicto con respecto a las características?
¿Hay requisitos que describan dos o más acciones que estén en conflicto temporalmente?
Rastreabilidad
¿Los requisitos fueron rastreados en todos los módulos?
Comprobabilidad
¿Existen requisitos que sean imposibles de cumplir?
Modificalidad
¿En el documento los requisitos se organizan de manera clara y lógica?
¿La redundancia es mínima y sólo se debe a distintos niveles de abstracción o detalle?
Contenido General
¿Cada requisito es relevante al problema y a su solución?
¿El artefacto cumple con la plantilla especificada por calidad interna?

Prototipo de Interfaz
¿Se especifican todas las interfaces utilizadas?
¿Se especifican todas las interfaces del hardware?
¿Se especifican todas las interfaces del software?
¿Se especifican todas las interfaces de comunicaciones?
¿Se especifican todos los requisitos del diseño de interfaz?
Hardware
¿Se especifica el tipo de hardware que necesita la aplicación?
Software
¿Se especifica el software requerido y el sistema operativo?
Trazabilidad
Cada requisito obedece a una necesidad específica de usuario.
Cada requisito tiene su origen en una fuente (documento o persona) específica.
Cada requisito se puede rastrear hacia delante su incorporación en determinados módulos.
Verificabilidad
Cada requisito es implementable.
Para cada requisito existe un procedimiento que, ejecutado por una persona o máquina, permite verificar si se cumple

Hay algún requisito que se va a expresar en términos verificables más adelante.

Anexo 2: Lista de chequeo para la descripción de casos de uso del sistema.

Evaluación
¿Está debidamente identificada la plantilla?
¿La plantilla cuenta con una sección de Control de Versiones ?
¿La plantilla cuenta con un Índice General , con una jerarquía de tópicos bien definida?
¿La plantilla cuenta con una sección de Introducción ?
¿Está definido el Propósito de la modelación dentro de la introducción?
¿Está definido el Alcance de la modelación dentro de la introducción?
¿Está definida una subsección de Definiciones, Acrónimos y Abreviaturas dentro de la introducción?
¿Existe una sección destinada a recoger las Referencias dentro de la introducción?
¿La plantilla cuenta con una sección donde se identifiquen y se describan los Actores del Sistema ?
¿La plantilla cuenta con una sección que recoja el diagrama de Casos de Uso del Sistema?
¿La plantilla cuenta con una sección de Especificación de Casos de Uso ?
¿El formato de letra utilizado para la plantilla en su totalidad es uniforme y solo diferencia tópicos y conceptos que verdaderamente necesiten ser diferenciados del resto del texto?
¿Es la documentación o redacción formal lo suficiente clara para ser entendida por personas fuera del equipo de proyecto?
¿Las descripciones se escriben de forma comprensible al usuario/cliente?

¿Cada descripción textual esta conformada por una terminología única?
¿Hay un glosario de términos específicos y claros que facilite el entendimiento?
¿Existen errores gramaticales?
¿Los párrafos están separados por dos espacios y justificados?
¿Cada apéndice independiente está separado uno del otro con un espacio y ordenado desde la izquierda?
¿Los apéndices están separados del texto anterior y posterior por tres y dos espacios respectivamente?
¿Existen errores ortográficos?
¿Del contraste contra diversas fuentes no surgen errores u omisiones?
¿La organización es de manera clara y lógica?
¿La organización cumple con un estándar aceptado?
¿La redundancia es mínima y sólo se debe a distintos niveles de abstracción o detalle?
¿Se etiquetan todas las figuras, tablas y diagramas?
¿Cada frase aporta a la especificación?
Actores del Sistema
¿Hay correspondencia entre los actores del sistema y los trabajadores del negocio?
¿Están descritos todos los actores de manera breve, concreta y formal?
¿Cada actor del sistema expresa un rol, y no una persona?
¿Cada actor del sistema está involucrado con al menos un caso de uso?

Diagrama de Casos de Uso del Sistema

¿EL conjunto de casos de uso del Sistema conforman todas las funcionalidades que se quiere describir con ellos?

¿Existen varios casos de uso del sistema con nombres similares?

¿El diagrama aparece bien estructurado?

¿El diagrama provee una visión general de los casos de uso del sistema fácil de comprender?

¿Existen muchas relaciones en el diagrama?

¿El diagrama es tan complejo y largo que debería ser dividido en varios diagramas más pequeños?

¿Existe algún caso de uso que no tenga relación alguna con un actor del cual se pueda prescindir?

¿Sólo son mencionados los actores que interactúan con el caso de uso?

Especificaciones de Casos de Uso

¿Cada especificación de casos de uso está debidamente numerada e identificada?

¿Los casos de uso del sistema están alineados con la estrategia del producto final?

¿Cada caso de uso del sistema sustenta al menos un objetivo de lo que se quiere automatizar?

¿Está claramente especificado el propósito del caso de uso?

¿Está definido el nivel de prioridad entre los casos de uso?

¿El caso de uso ejecuta sólo actividades dentro del sistema?

¿Existen varios casos de uso con nombres similares?

¿Los flujos centrales y alternos están totalmente descritos?

¿Cada caso de uso sustenta un objetivo del sistema?
¿Están claramente especificadas las precondiciones del caso de uso?
¿Están claramente especificadas las poscondiciones del caso de uso?
¿Cada caso de uso está involucrado con al menos un actor?
¿Los nombres de los casos de uso expresan sin ambigüedad la funcionalidad del caso de uso al que representan?
¿El nombre y la descripción son entendibles incluso para personas fuera del equipo de desarrolladores?
¿Están descritos todos los posibles flujos que pertenecen al caso de uso?
¿Está descrito claramente cuando puede variar el orden de las actividades?
¿Las descripciones de los flujos de trabajo central y alternos están bien estructuradas?
¿Se hace referencia a los requisitos implícitos en el caso de uso?
¿El inicio y fin del flujo de trabajo están claramente descritos?
¿Cada relación extendida (extended) está claramente descrita de forma tal que es obvio cómo y cuándo es insertado el caso de uso?
Paquetes de Casos de Uso del Sistema
¿Están claramente especificados el nombre y el propósito de cada paquete de casos de uso?
¿El contenido de cada paquete de casos de uso es consistente con su nombre y su propósito?
¿Es razonable la proporción entre el número de paquetes y el número de objetivos, actores y casos de uso del sistema?
¿Los paquetes de casos de uso están anidados muy profundamente?
¿La estructura del paquete añade claridad y comprensibilidad al modelo?

Anexo 3: Acta de aceptación de la documentación del módulo Presentación del sistema CCV.



ACTA DE ACEPTACIÓN

Producto: Sistema de Gestión para Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba-Venezuela.

Categoría: Aceptación CU Módulo de Presentación.

Fecha de la conciliación: 20/06/2008

Involucrados en el proceso:

Por la parte del Cliente (MENPET): Sandra Cortés

Por la parte del Suministrador (ALBET): Ing. Nahuel Massón

Observaciones del proceso:

Por acuerdo entre las partes involucradas en el proceso se Acepta el Documento CU del Módulo de Presentación de la versión v.2.0 del Sistema de Gestión para Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba-Venezuela, con fecha 20 de junio de 2008.

Para que conste la aceptación de la descripción de los CU y requerimientos del Módulo de Presentación, dando fe al acuerdo, firman la presente los principales representantes de las Partes.

Sandra Cortés
Representante MENPET

Ing. Nahuel Massón
Representante ALBET

Acta de Aceptación



Producto: Sistema de Gestión para Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba-Venezuela.

Categoría: Aceptación CU Módulo Presentación.

Fecha de la conciliación: 04/05/2008

Involucrados en el proceso:

Por la parte del Cliente (MENPET): Sandra Cortés

Por la parte del Suministrador (ALBET): Ing. Nahuel Massón

Observaciones del proceso:

Por acuerdo entre las partes involucradas en el proceso se Acepta el Documento CU del Módulo Presentación de la versión v.1.0 del Sistema de Gestión para Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba-Venezuela, con fecha 4 de mayo de 2008.

Para que conste la aceptación de la descripción de los CU y requerimientos del Módulo Presentación, dando fe al acuerdo, firman la presente los principales representantes de las Partes.


 Sandra Cortés
 (Por la parte del Cliente)


 Ing. Nahuel Massón Padilla
 (Por la parte Cubana - ALBET)

ALBET, S.A.

Centro de Negocios Miramar, Edificio Barcelona,
Oficina 322, Avenida Sta e/76 y 78, Miramar,
Playa, Ciudad Habana, Cuba

Tel/Fax: +53 (7) 837 2407

E-mail: albet@albet.cu

Anexo 4: Acta de aceptación del módulo Presentación del sistema CCV.**Acta de Aceptación**

Producto: Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela (CCV)

Categoría de las pruebas: Revisión de Módulos Presentación, Administración y Configuración del CCV versión 1.0.

Fecha de conciliación: 20 de agosto de 2008

Involucrados en el proceso:

- **Por la parte del Cliente (CCV):** Sandra Cortés
- **Por la parte desarrolladora (ALBET):** Ing. Nahuel Massón Padilla
- **Observador independiente (CALISOFT):** Ing. Dayami Rodríguez Brito
- **Observador independiente (CALISOFT):** Ing. Roig Calzadilla Díaz

Observaciones del proceso:

Durante el proceso de pruebas de aceptación se identificaron un conjunto de cambios y mejoras necesarias que quedaron registrados adecuadamente en el correspondiente documento de Respuestas a las No Conformidades detectadas, con sus respectivas observaciones. Teniendo en cuenta que las No Conformidades han sido debidamente respondidas, ejecutadas por el Equipo de Desarrollo y validada la eficacia de los mismos por los clientes, se ha tomado el acuerdo de aceptar el "Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela" para la versión 1.0, con fecha 20 de agosto de 2008, el cual se anexa a esta Acta, estableciendo de esta forma la condición necesaria y suficiente para su despliegue.

Para que conste la aceptación de los resultados de las pruebas y por tanto la aceptación del "Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela", dando fe del acuerdo firman la presente los principales representantes de las partes.

Sandra Cortés

(Por la parte del Cliente)

Ing. Nahuel Massón Padilla
(Por la parte Cubana - ALBET)

Ing. Roig Calzadilla Díaz
(Observador
Independiente CALISOFT)

Acta de Aceptación



REGISTRO DE LA RESPUESTAS A LAS NO CONFORMIDADES

Elemento	No	No conformidad	Aspecto correspondiente	Relevancia /Nivel de Falla	Estado	Solución Propuesta
Módulo Presentación	1	Extender el tiempo de completamiento de las unidades de pesos.	Módulo Presentación.	Media	RA	Extender a 900 ms el tiempo de espera para completar las unidades monetarias.
Módulo Presentación	2	Cuando una ST acepta un proyecto y la otra la rechaza no se muestra la notificación a nivel de ST.	Notificaciones.	Alta	RA	Generar la notificación una vez ocurrido las acciones planteadas.
Módulo Presentación	3	Cambiar la foto por una que tenga mayor calidad en la pantalla inicial de la Presentación.	Página de inicio Módulo Presentación.	Baja	RA	Mejorar las imágenes de modo que tuvieran mayor calidad.
Módulo Presentación	4	Aumentar el número de personas limite.	Recursos Humanos	Medio	RA	Permitir un número mayor de personas en los RRHH.
Módulo Presentación	5	Cambiar "Adquisición de Recurso Materiales" por "Asignar Recursos Materiales o Suministros".	Recursos Materiales	Baja	RA	Cambiar el texto señalado.

Anexo 5: Acta de finiquito del sistema CCV.**ACTA DE FINIQUITO**

Entre la República Bolivariana de Venezuela, actuando por órgano del Ministerio del Poder Popular para la Energía y Petróleo de la República Bolivariana de Venezuela, representado en este acto por el ciudadano **Ammar Jabour**, venezolano, mayor de edad, de este domicilio, titular de la Cédula de Identidad No. 9962729, quien actúa en su condición de Coordinador General del Convenio Integral de Cooperación Cuba-Venezuela, suficientemente facultado para este acto y debidamente designado para ejercer tal condición conforme a lo establecido en **CONTRATO E08-001-000 SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, que en lo sucesivo se denominará la "**Parte Venezolana**", por una parte; y por la otra, la sociedad mercantil **ALBET, Ingeniería y Sistemas, S.A.**, sociedad mercantil cubana constituida mediante Escritura 271 de fecha 7 de Noviembre de 2005, autorizada por la Notario Lic. Isabel Cristina Martínez Alfonso con sede en Notaría Especial del Ministerio de Justicia de Ciudad de la Habana, inscrita en el Registro Mercantil de esta ciudad con fecha 14 de Noviembre del año 2005, al Tomo XVIII, Folio 120, Hoja 11, Sección SM, con N° de inscripción 1 con domicilio social en Centro de Negocios de Miramar, Edificio Barcelona, Oficina 322, Avenida 5ta entre 76 y 78, Miramar, Municipio Playa, Ciudad de La Habana, República de Cuba, representada en este acto por el ciudadano cubano **Ibrahim Nápoles Albanés**, mayor de edad, portador de carné de identidad N° 62032504808 en su condición de Coordinador General, suficientemente facultado para este acto según lo dispuesto en **CONTRATO E08-001-000 SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, que en lo sucesivo se denominará "**Parte Cubana**", al tenor de las siguientes declaraciones y cláusulas:

CONSIDERANDO

Primero: Consta de documento privado suscrito en fecha 18 de marzo de 2008, que ambas partes celebraron un **CONTRATO DE SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, el cual fue celebrado al Amparo del Convenio Integral de Cooperación, suscrito el 30 de octubre del 2000 y su Addendum; de la Declaración Conjunta y el Acuerdo suscrito entre ambas Repúblicas, para la aplicación de la Alternativa Bolivariana para las Américas, firmados en diciembre del 2004 y de los Acuerdos y las Condiciones Generales firmados en el Acta de la Reunión de la Comisión Mixta Cuba-Venezuela, celebrada en La Habana, Cuba, cuyos contenidos se dan aquí enteramente por reproducidos.

Segundo: Consta que la **Parte Cubana** ha cumplido a entera satisfacción de la **Parte Venezolana** con el objeto, alcance y actividades previstas en el **Contrato** ya mencionado, así como con sus Anexos y Suplementos, cumpliendo por tanto con todos sus deberes y obligaciones por lo que se considera que la Solución ha sido implementada en las condiciones previstas y bajos los requisitos y especificaciones técnicas pactadas entre **Las Partes**.

Tercero: Consta que la **Parte Cubana** ha hecho entrega del **Informe Técnico Final**, de conjunto a toda la documentación que avala y soporta lo descrito en el Segundo de los **CONSIDERANDOS**, debidamente firmada y aceptada por los especialistas de la **Parte Venezolana**.

Cuarto: Consta que la **Parte Venezolana** ha pagado la totalidad de **UN MILLÓN CUATROCIENTOS DIECISIETE MIL OCHOCIENTOS VEINTIOCHO DOLARES DE LOS ESTADOS UNIDOS DE AMERICA CON VEINTIUN CENTAVOS** mediante la forma de pago prevista en el Contrato ya mencionado, a entera satisfacción de la Parte Cubana, quedando a la firma de la presente **Acta**.

LAS PARTES CONVIENEN:

Primero: Dar por terminada la relación contractual derivada del **CONTRATO DE SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA** de fecha 16 de marzo de 2008.

Segundo: **Las Partes** se otorgan el finiquito más amplio que en derecho proceda, no reservándose acción o derecho que ejercitar con posterioridad.

Leída que les fue la presente Acta, las partes se ratificaron en su contenido, para constancia firman en cuatro (4) ejemplares de igual tenor en la Ciudad de Caracas el día 17 del mes de diciembre del 2008.

Por la **PARTE VENEZOLANA**

Por la **PARTE CUBANA**



Ammar Jabour
Coordinador General



Ibrahim Nápoles Albanés
Coordinador General

Glosario de términos

Actores: Roles pertenecientes a los usuarios, agrupados según sus iteraciones con las funcionalidades del sistema.

CASE: (Computer Aided Software Engineering), Ingeniería de Software Asistida por Ordenador).

Caso de uso (CU): Representación de la agrupación de funcionalidades comunes. Representan un conjunto de iteraciones entre el sistema y sus actores.

Comisión Mixta Cuba-Venezuela: Mecanismo para la concepción, evaluación y seguimiento de las de las acciones de cooperación del Convenio Cuba-Venezuela.

CRUD: Patrón de casos de uso. Plantea que las funcionalidades de crear, obtener, actualizar y eliminar (Create, Read, Update y Delete), se pueden agrupar en un mismo CU.

DCUS: Diagrama de casos de uso del sistema. Representación de la relación de los CU con los actores del sistema.

EE: Ente Ejecutor, organismo, empresa o institución, es el responsable de un proyecto.

GRASP (General Responsibility Assignment Software Patterns): Patrones generales de software para asignación de responsabilidades.

Hibernate: Herramienta de Mapeo objeto-relacional.

IR: Ingeniería de requisitos.

M: Ministerio

Presentación de Proyectos: Proceso dentro de la Comisión Mixta Cuba-Venezuela donde se elaboran los proyectos, se trazan las metas y objetivos, se destinan los recursos que se invertirán en la ejecución de los mismos y se someten a un proceso de revisión transitando por niveles superiores. Finalmente se dejan listos para su ejecución una vez que se les asigna el financiamiento.

Paquetes de diseño: Es una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes que estén de alguna forma relacionados. No definen un comportamiento o semántica bien definida.

Persistente: Conjunto de datos y elementos que deber ser almacenados por el espacio de tiempo que se requiera, para dar soporte de información a un sistema u organización.

Plan operativo: Cronograma de todas las actividades del proyecto, donde se especifican los recursos que se emplearán en la ejecución física de cada una de ellas.

Requerimientos: Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

ST: Secretaría Técnica, oficina rectora de la administración de las actividades económicas y colaborativas de un país.

Subsistemas de diseño: Representan una separación de aspectos lógicos de diseño. Constituyen una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Puede contener clases del diseño, realizaciones de casos de uso, interfaces y otros subsistemas.