

Universidad de las Ciencias Informáticas

Facultad 8



**Análisis del Módulo de Gestión de Costos del Sistema
Integral de Gestión de Portafolios de proyectos.**

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

Autor: Kenia Rondón Nobrega.

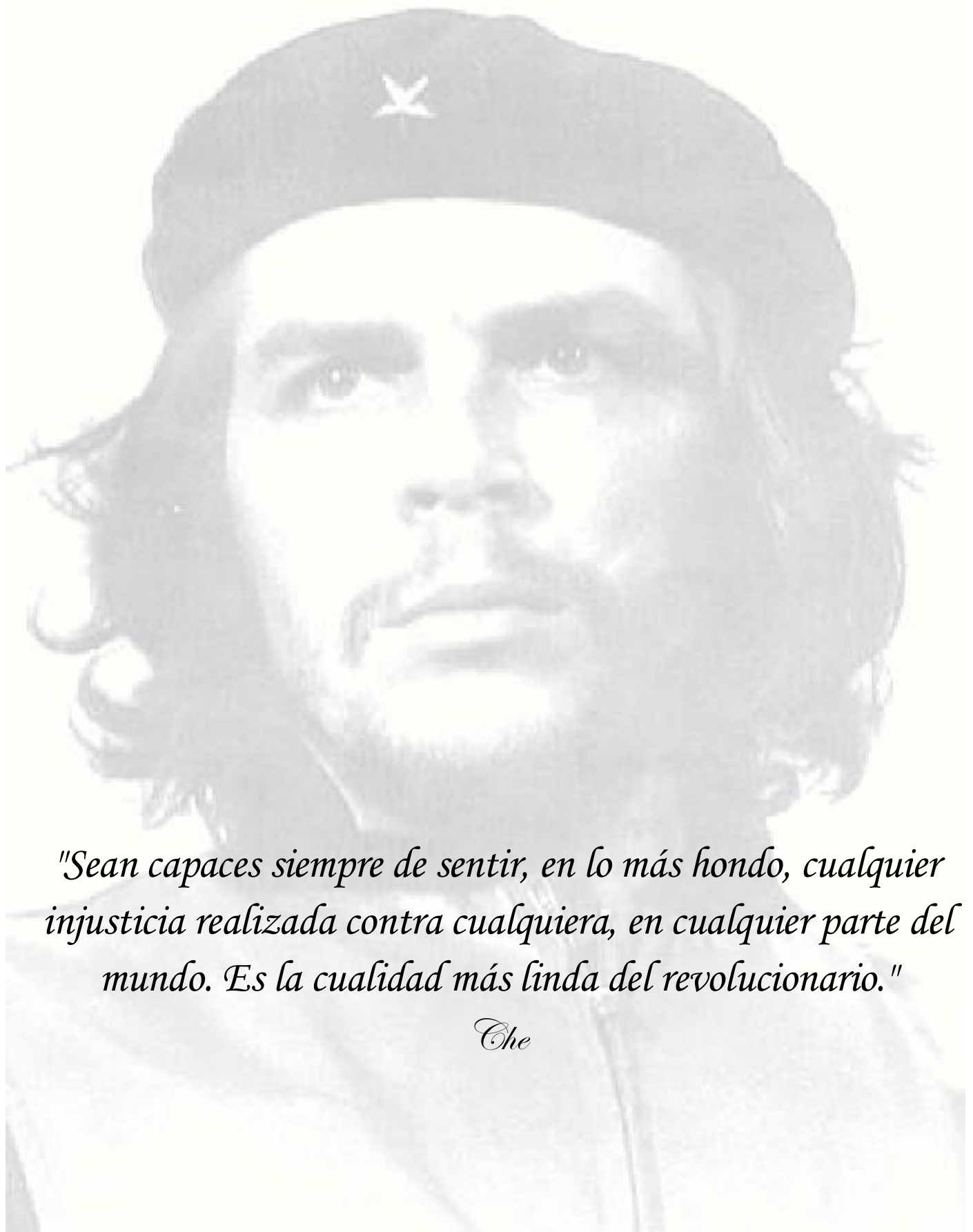
Tutor(es): Ing. Adisleydis Rodríguez Pino.

Ing. Jenny Infante Frometa.

Consultor: Ing. Manuel Vázquez Acosta.

Ciudad de La Habana, Junio del 2010

“Año 52 de la Revolución”



"Sean capaces siempre de sentir, en lo más hondo, cualquier injusticia realizada contra cualquiera, en cualquier parte del mundo. Es la cualidad más linda del revolucionario."

Che

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo “Análisis del Módulo de Gestión de Costos del Sistema Integral de Gestión de Portafolios de proyectos” y autorizo a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2010.

Kenia Rondón Nobrega
Autor

Ing. Adisleydis Rodríguez Pino
Tutor

Ing. Jenny Infante Frometa
Tutor

Dedicatoria

A mis padres, mis abuelos, mi hermana y mis sobrinas por ser lo más grande que tengo en la vida.
Gracias por existir.

Agradecimientos

A mi papá por ser mi gran ejemplo y enseñarme a ser fuerte y seguir adelante.

A mi mamá por ser la mejor mamá del mundo y siempre estar cuando la necesito.

A mi hermana por darme dos sobrinas tan bellas, que son lo que más quiero en la vida.

A mi familia y a todos mis amigos por su apoyo incondicional.

A Orestes, Yanitza y Dania por ser mis grandes amigos y por su ayuda en estos años.

A Jenny y Adisleidys por ser tutoras excepcionales.

Resumen

El presente trabajo propone un análisis del Módulo de Gestión de Costos del Sistema Integral de Gestión de Portafolios de proyectos, para dar solución a la ausencia de una herramienta que integre todos los procesos del área de costos, que permita determinar la ganancia y reducir los costos de la empresa. Para resolver el trabajo investigativo, se realizó un estudio detallado del arte de las tendencias mundiales de la Gestión de Costos y la Gestión de Portafolios de proyectos, así como de los sistemas informáticos referentes al tema. Se desarrolló una propuesta de solución al problema, donde se encuentra el modelo del dominio que expresa la relación de las entidades del área de costos, además se muestra el diagrama de paquetes y el diagrama de Casos de Uso del sistema por paquetes. También se presentan las descripciones de los Casos de Uso más significativos y se validan los resultados de la solución para verificar que posean la calidad requerida; pretendiendo que a partir del análisis realizado, posteriormente se pueda implementar el sistema, garantizando que las inversiones estratégicas se encuentren lo más alineado posible a las prioridades de la organización.

Palabras Claves: Gestión de Portafolio de proyectos, Gestión de Costos, costos, Inversiones estratégicas.

ÍNDICE DE CONTENIDO

Introducción..... 1

Fundamentación Teórica 1..... 5

 1.1 La Gestión de Portafolios de proyectos..... 5

 1.2 Tendencias actuales de la Gestión de Costos..... 7

 1.2.1 IPMA..... 7

 1.2.2 PMBOK..... 10

 1.3 Sistemas Informáticos relacionados con la Gestión de Portafolios de proyectos..... 13

 1.4 Metodología y herramientas para la modelación de software..... 18

 1.4.1 Metodología..... 18

 1.4.2 Lenguaje de Modelado..... 18

 1.4.3 Herramienta CASE..... 19

 1.5 Ingeniería de Requerimientos..... 20

 1.5.1 Características de los requerimientos..... 20

 1.5.2 Requisitos funcionales y no funcionales..... 20

 1.6 Patrones de Caso de Uso..... 23

 1.7 Conclusiones..... 25

Características del Sistema 2..... 26

 2.1 Propuesta del Sistema..... 26

 2.3 Modelo de dominio..... 29

 2.3.1 Diagrama conceptual del modelo de dominio..... 29

 2.4 Especificación de los requisitos de software..... 31

 2.4.1 Requisitos funcionales..... 31

 2.4.2 Requisitos no funcionales..... 39

2.5 Definición de los casos de uso	41
2.5.1 Definición de los actores	41
2.5.2 Diagrama de Casos de Uso del Sistema.....	42
2.6 Conclusiones.....	51
Análisis de los Resultados 3.....	52
3.1 Validación de los requisitos	52
3.2 Estudio de la validación por fases	53
3.2.1 Fase I. Validación del listado de requisitos de software	54
3.2.2 Fase II. Validación del documento de especificación de requisitos.....	56
3.2.3 Fase III. Validación del modelo de Casos de Uso del Sistema.....	60
3.3 Pruebas.....	65
3.3.1 Pruebas de sistemas	65
3.4 Conclusiones	68
Conclusiones Generales.....	69
Recomendaciones.....	70
Referencias Bibliográficas	71
Bibliografía	74
Anexos.....	79
Anexo 1 Entrevista	79
Glosario de Término	81

ÍNDICE DE FIGURAS

Figura 1 Relación de un Portafolio y de los componentes de la cartera (PMI, 2006)..... 6

Figura 2 Definición y cálculo de las capacidades de competencia de IPMA..... 8

Figura 3 Descripción General de la Gestión de Costos del Proyecto (PMI, 2004). 11

Figura 4 Gráfica de desviación del presupuesto programado, utilizando la gráfica curva "S" 28

Figura 5 Diagrama de Modelo de Dominio 30

Figura 6 Resultados de la entrevista por preguntas 32

Figura 7 Diagrama de actores 42

Figura 8 Diagrama de Paquetes..... 43

Figura 9 Diagrama de Casos de Uso del paquete Gestión de Costo de un proyecto 44

Figura 10 Diagrama de Casos de Uso del paquete Reportes ¡Error! Marcador no definido.

Figura 11 Diagrama de Casos de Uso del paquete Configuración 48

Figura 12 Diagrama de Casos de Uso del paquete Funciones Básicas ¡Error! Marcador no definido.

Figura 13 Diagrama de Casos de Uso del paquete Funcionalidades Generales ¡Error! Marcador no definido.

Figura 14 Resultados de las Revisiones de la Fase I..... 56

Figura 15 Representación de los valores aportados por cada factor de calidad aplicado. 64

ÍNDICE DE TABLAS

Tabla 1 Comparación de los procesos del área de conocimiento en sistemas de código abierto. 17

Tabla 2 Actores del Sistema 41

Tabla 3 Descripción del Caso de Uso Gestionar costo a un proyecto 47

Tabla 4 Descripción del Caso de Uso Crear Personalizar el costo del campo 51

Tabla 5 Descripción del Caso de Uso Crear Gestionar recursos de un proyecto **¡Error! Marcador no definido.**

Tabla 6 Descripción del Caso de Uso Crear Centro de costos **¡Error! Marcador no definido.**

Tabla 7 Ejemplo de elementos definidos en la lista de chequeo especificación de requisitos. 58

Tabla 8 Matriz de trazabilidad. 61

Tabla 9 Aplicación de métricas al diagrama de casos de uso del sistema. 64

Tabla 10 Caso de Prueba de la SC 1 Adicionar costo a un proyecto 67

Tabla 11 Caso de Prueba de la SC 2 Modificar costo a un proyecto 67

Tabla 12 Caso de Prueba de la SC 3 Mostrar costo a un proyecto 68

Tabla 18 Preguntas de la entrevista 80

Introducción

El mundo empresarial actual está sufriendo un sinnúmero de cambios debido a la creciente competencia global y el auge de las innovaciones tecnológicas, por lo que es una realidad para los directivos la necesidad de obtener información que les permita tomar decisiones en un conjunto de proyectos con relación al diseño de productos, así como en los procesos tecnológicos y elementos vinculados a la rentabilidad de la organización. Todo esto impone que las unidades organizativas modifiquen las gestiones de sus procesos y actividades. Una nueva cultura se va alcanzando paulatinamente en las empresas, tratando de adaptarse al nuevo entorno que les rodea, con el fin de aumentar sus posibilidades de mercado y poder emplear de manera efectiva el resto de los recursos. En este contexto uno de los procesos clave lo es, sin lugar a dudas, la Gestión de Proyectos, que desempeña un papel fundamental en el mundo debido a que es el encargado de organizar y administrar recursos de manera tal que se pueda culminar con éxito todo el trabajo requerido.

Actualmente en muchas empresas se desarrollan varios proyectos a la vez, por lo que se necesita una administración profesional y centralizada de los mismos para maximizar las probabilidades de éxito. Conocer la situación real en cada momento de los proyectos de una empresa u organización de cualquier tipo es un requisito imprescindible para la gestión eficiente de recursos. Sin embargo, debido al escaso conocimiento de las herramientas informáticas especializadas y el poco conocimiento en general de la dirección moderna de Portafolios de proyectos, pocas organizaciones tienen una metodología sistemática de gestión de los recursos dedicados a los proyectos.

La Gestión de Portafolios de proyectos es el manejo centralizado de uno o más portafolios y envuelve la identificación, priorización, autorización, gestión y control de proyectos, programas y otros trabajos relacionados, para alcanzar las metas estratégicas de negocio (Saborío, 2006). Proporciona métricas, adicionales a las financieras para la buena dirección del negocio, y brinda a la empresa la habilidad de ajustar y realinear sus presupuestos con las necesidades del negocio más de una vez al año.

Para la Gestión de Portafolios de proyectos es fundamental tener en cuenta la Gestión de Costos, que se encarga de asegurar que las tareas se lleven a cabo dentro de los rangos económicos impuestos, la misma resulta de suma importancia porque determina la viabilidad del negocio, así como el grado de productividad y eficacia en la utilización de los recursos. Además, facilita la toma de decisiones de los directivos de la organización, mejoran el modo de gestionar los recursos y actividades que se consumen y

muestran cuánto cuesta la producción de determinados artículos o la prestación de determinados servicios. También resulta de gran magnitud para la Dirección Económica porque permite determinar la ganancia de la empresa, evaluar y calcular los inventarios de producción, controlar los insumos en el proceso de producción y tomar decisiones para nuevas alternativas de producción y ventas.

En el mundo existen algunas herramientas utilizadas para la Gestión de Portafolios de proyectos, entre ellas se encuentran Rational Focal Point, Primavera Enterprise Project Portfolio Management de Oracle, Microsoft Enterprise Project Management (EPM) de Microsoft, entre otras. Contar con una de estas herramientas sería de gran ayuda para Cuba, porque le brindaría soporte a la toma de decisiones y así, podría balancear de forma continua los riesgos y beneficios de un portafolio.

En la Universidad de las Ciencias Informática (UCI) se desarrollan muchos proyectos, los cuales se hace muy complejo de controlar. Un ejemplo clave lo es la empresa ALBET; también se tiene el Centro de Tecnología de Gestión de Datos (DATEC) dividido en líneas, donde a cada una se le asignan un conjunto de proyectos para realizar. Dicho Centro, la Universidad, así como algunas organizaciones que tienen convenio con la misma, necesitan controlar las carteras de proyectos, por lo cual requiere de la creación de una nueva herramienta para la Gestión de Portafolios de proyectos; con la misma se pretende incrementar la calidad en el desarrollo de productos y automatizar los procesos de estimación y presupuesto de los costos. Debido a esta necesidad, en la UCI se ha creado una oficina de Gestión y Planificación de Proyectos que proporcionará soporte en los procesos de planificación, monitoreo y control de las unidades productivas de la Universidad, teniendo en cuenta la gestión de recursos, de riesgos y de tareas.

Por otro lado, la estimación aproximada de los costos de los recursos necesarios para completar las actividades de un conjunto de proyectos es irreal y no se llevan de forma sostenida durante todo el ciclo de vida, provocando así planificaciones irreales y desviaciones considerables del presupuesto asociado a los Portafolios de proyectos. Debido a todo lo antes expresado se necesita contar de una Plataforma que integre y gestione todos los procesos que se incluyen en la Gestión de Costos con la Gestión de Portafolios de proyectos, que permita mejorar la toma de decisiones, y la reducción de los costos de las carteras de proyectos; permitiendo garantizar que las inversiones estratégicas estén alineadas con las prioridades corporativas y se entreguen dentro del presupuesto establecido.

A partir de las necesidades detectadas, se plantea el siguiente **Problema a resolver:**

¿Cómo realizar de manera eficiente la estimación y presupuesto de los costos para un Portafolio de proyectos?

Objeto de Estudio

Gestión de Portafolios de proyectos.

Objetivo General

Realizar el análisis del Módulo de Gestión de Costos perteneciente al Sistema Integral de Gestión de Portafolios, posibilitando posteriormente su diseño e implementación.

Objetivos Específicos

- Obtener el modelo de dominio.
- Obtener los requerimientos funcionales y no funcionales.
- Modelar los casos de uso del sistema.
- Validar los requisitos del sistema.

Campo de Acción

Gestión de Costos de Portafolios de proyectos.

Idea a Defender

Si se realiza detalladamente el análisis del Módulo de Gestión de Costos, se obtendrán las bases para su posterior diseño e implementación y se logrará reducir el costo y mejorar la toma de decisiones de las carteras de proyectos.

Tareas de la investigación

1. Sistematización del estado actual de la Gestión de Portafolios de proyectos, específicamente la Gestión de Costos.
2. Estudio y evaluación de los sistemas informáticos de Gestión de Portafolios de proyectos.
3. Análisis del proceso de Ingeniería de Requisitos.
4. Realización del flujo de análisis del Módulo de Gestión de Costos.
5. Validación de los requisitos identificados a través de técnicas y métricas.

Métodos Científicos

Para el desarrollo de la investigación se utilizaron varios métodos y técnicas. Uno de ellos es el Histórico – Lógico, el cual permitió estudiar la evolución de la Gestión de Portafolios de proyectos, específicamente la Gestión de Costos y los logros que se han obtenido; el Analítico – Sintético que facilitó el resultado del trabajo con el análisis de la Gestión de Costos y posibilitó sintetizar las características esenciales, como

también permitió arribar a las conclusiones de la investigación. Además, se tiene el Hipotético – Deductivo que posibilitó obtener la Idea a Defender planteada en el trabajo; y el Análisis Bibliográfico que ayudó a escrutar gran cantidad de bibliografía referente a la Gestión de Portafolios, incluyendo la Gestión de Costos, que posibilitó el desarrollo del trabajo con mayor facilidad. Se utilizó la observación para percibir los problemas existentes en el desarrollo de productos con relación a la Gestión de Portafolios de proyectos, este método ayudó a obtener una mejor visión del problema y el objetivo de esta investigación.

Aportes esperados de esta investigación

De ser desarrollado el Módulo de Gestión de Costos se podría obtener beneficios tanto para la Universidad de las Ciencias Informáticas como para el país:

- Se contaría con una herramienta para controlar los proyectos que se desarrollan en una empresa u organización.
- Se podría comercializar, dejando grandes ganancias al país.
- Tendría una administración efectiva de recursos, proyectos y procesos.
- Se contaría con un software propio, incluyendo todas las ventajas de trabajar en el software libre.
- Una herramienta de soporte a la toma de decisiones usado por los ejecutivos para balancear de forma continua los riesgos y beneficios de un portafolio.

Estructura de la Tesis

La tesis está estructurada en tres capítulos.

Capítulo 1: Referido al marco teórico y referencial de la investigación donde se realiza un análisis crítico y valorativo del estado del arte de la Gestión de Portafolios de proyectos, en el tema de Gestión de Costos, así como técnicas, metodologías y software usados para apoyar la solución del problema.

Capítulo 2: Se plantean las características del sistema, proponiendo una estrategia que sirva de guía al proceso de Gestión de Costos en un paquete de proyectos.

Capítulo 3: Se expone el análisis de algunos resultados alcanzados.

Fundamentación Teórica **1**

En el desarrollo del presente capítulo se realizará un estudio del estado del arte de la Gestión de Portafolios de proyectos, específicamente la Gestión de Costos. Además, se analizan conceptos, tecnologías, metodologías y software que se utilizan actualmente en el mundo y que podrán ser utilizados para solucionar el problema que se plantea en este trabajo.

1.1 La Gestión de Portafolios de proyectos

Un portafolio es una colección de proyectos (temporales esfuerzos realizados para crear un producto único, servicio o resultado) o programas (un grupo de proyectos relacionados que se administran de manera coordinada para obtener beneficios) y otros trabajos que se agrupan para facilitar la gestión eficaz de los que trabajan para alcanzar los objetivos estratégicos del negocio. Los componentes de la cartera son cuantificables, es decir, que pueden ser medidos, clasificados, y priorizados (PMI, 2006).

Los proyectos o programas pueden no ser necesariamente interdependientes o directamente relacionados. El portafolio representa el conjunto de la organización de los programas activos, proyectos, subportafolios, y otros trabajos en un punto específico en el tiempo.

Es importante entender la relación de un portafolio y de los componentes que este posee (Ver **Figura 1**).

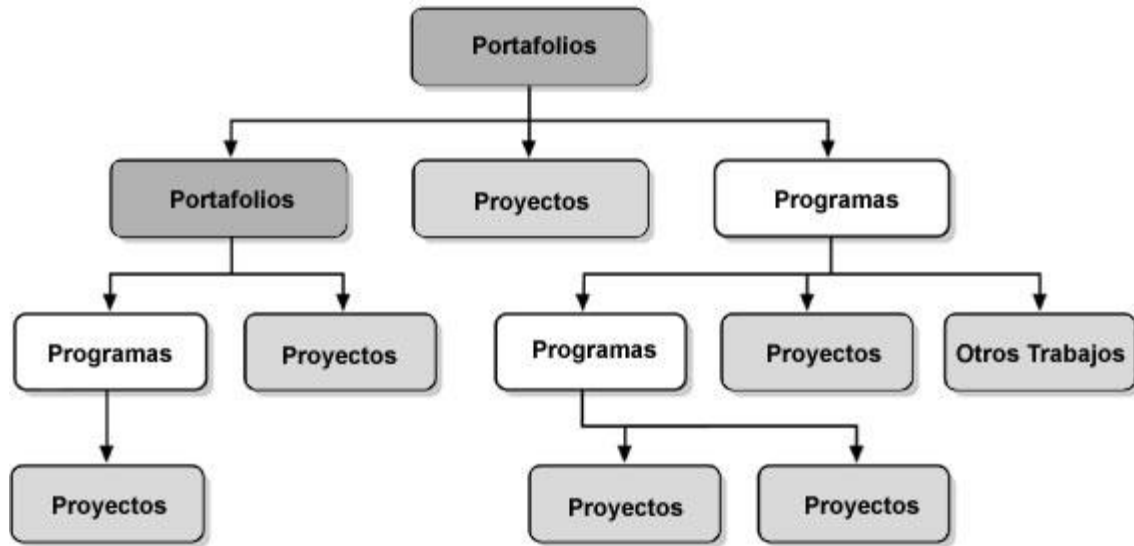


Figura 1 Relación de un Portafolio y de los componentes de la cartera (PMI, 2006).

La Gestión de Portafolios (PPM) se centraliza en gestiones de uno o más portafolios, que incluye la identificación, priorización, autorización, administración y control de los proyectos, programas y otros trabajos conexos, para lograr los objetivos estratégicos empresariales (PMI, 2006).

PPM puede contener más de un proyecto. En este contexto resulta fundamental gestionar de manera eficiente uno o varios proyectos. La Gestión de Proyectos se encarga de supervisar y controlar la ejecución de actividades planificadas y establecidas.

La Gestión de Proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requerimientos del proyecto. Este se logra mediante la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento, control y cierre (PMI, 2004).

Para lograr una efectiva Gestión de Portafolio se requiere llevar el control de los costos y presupuestos de cada proyecto para garantizar la disminución de las variaciones de los costos y que se mantengan dentro del presupuesto destinado para el mismo. La forma en que se gestionen los costos asociados a proyectos en una organización trae como consecuencia el ahorro económico por concepto de reducción de presupuesto, de ahí la importancia de elegir el método adecuado para realizar este proceso.

A continuación se realiza un estudio de las principales tendencias existentes en el mundo respecto a la Gestión de Costos, enfocado en los modelos Fundamentos de la Dirección de Proyectos (PMBOK) e International Project Management Association (IPMA).

1.2 Tendencias actuales de la Gestión de Costos

La Gestión de Costos (GC) asegura que las tareas se lleven a cabo dentro de los rangos económicos impuestos (presupuesto del proyecto o recursos asignados para la actividad correspondiente). Es importante para la eficiencia de la empresa, porque muestra cuanto le cuesta la producción de determinados artículos o la prestación de determinados servicios. En él se refleja el ahorro de los recursos materiales, laborales y financieros.

Además reviste gran importancia para la Dirección Económica porque permite determinar los gastos de la actividad y la ganancia de la empresa, establecer o demostrar los listados de precios, planificar en concordancia con el nivel de la actividad prevista en los indicadores para un período de operaciones, controlar los insumos en el proceso de producción y tomar decisiones para nuevas alternativas de producción y ventas.

A continuación se estudiarán los modelos PMBOK e IPMA, propuestos por los más prestigiosos centros de pensamiento de Gestión de Portafolios de proyectos, tomándose como base para el desarrollo del análisis del Módulo de Gestión de Costos para un Portafolio.

1.2.1 IPMA

International Project Management Association (IPMA), en español Asociación Internacional de Gestión de Proyectos, es “la organización mundial líder en la Gestión de Proyectos”. Este modelo se puede utilizar a través de la gama entera de gerentes de proyectos. Determina el nivel de conocimiento, de experiencia (habilidad), de actitud y comportamiento personal de los gerentes de proyectos para manejar la complejidad del proyecto. La gama va desde coordinar un solo proyecto hasta manejar un portafolio de proyectos de una corporación a través de multi-proyectos y programas.

Las capacidades en cada nivel se definen y calculan como se muestra en la **Figura 2**.

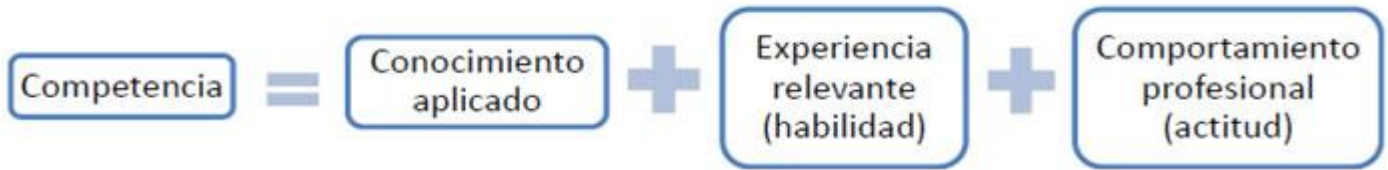


Figura 2 Definición y cálculo de las capacidades de competencia de IPMA.

Estos estándares no son más que la Línea Base de Competencia de IPMA la cual es una metodología compuesta por 46 elementos de competencia divididos en 3 capacidades respecto al: contexto de desarrollo del proyecto, comportamiento dentro del equipo y habilidades técnicas (IPMA, 2006). Precisamente dentro de las capacidades técnicas está incluido el elemento de competencia costo y financiación en el cual IPMA establece su criterio.

Costo y financiación

Los Costos del proyecto y la gestión de las finanzas es la suma de todas las medidas necesarias para la planificación, seguimiento y control de costos durante el ciclo vital del proyecto, incluyendo la evaluación de proyectos y estimaciones de costos en las primeras fases del proyecto.

La Gestión de Costos de los proyectos estima el costo de cada paquete de trabajo, los sub-sistemas y todo el proyecto y establece el presupuesto para el proyecto global. También supone la comparación prevista frente a los costos reales en varios puntos del proyecto y la estimación del costo restante, así como la actualización de la estimación del costo final. El costo de las prestaciones debe ser medible y calculable. El costo de cualquier cambio debe ser calculado, acordado y documentado.

El presupuesto del proyecto debe incluir una suma de dinero que se mantiene en reserva para financiar las incertidumbres, como las contingencias, siniestros o promedio de costos de las actualizaciones. Los fondos también pueden estar disponibles para cubrir los resultados favorables, tales como la gestión de riesgos con éxito y sin darse cuenta de las oportunidades.

El trabajo debe ser coherente con lo que el proyecto ha entregado, para las condiciones acordadas de pago. A la Gestión de Proyectos se aplican métodos para controlar, verificar y comprender lo real declarado de la entrega, el trabajo que debe ser facturado y el uso de los recursos (tales como sub-contratista de facturación y las horas de tiempo en las hojas). Esta es la base para el costo del proyecto, la presentación de informes y el control.

La gestión financiera de los proyectos asegura que en todas las fases del proyecto se sabe cuánto se requiere para cada intervalo de tiempo. Los recursos requeridos dependen del costo del proyecto, el calendario y las condiciones de pago. La gestión del proyecto también analiza los recursos financieros disponibles y gestiona toda la infrautilización o el exceso de gasto.

Debe haber fondos (un "buffer" o "flotante" o "imprevistos") disponible para las demandas inesperadas durante el ciclo vital del proyecto. Cuando se usan, los anticipos deben manejarse con especial cuidado. En términos de gastos, el efectivo en caja y el flujo de salida se calcula y se evalúan.

La financiación del proyecto abarca el proceso de recaudación de fondos de la manera más prudente y favorable. Existen diversas opciones para la financiación de proyectos, programas y portafolios, como la de los fondos internos, mediante préstamos bancarios o de construcción-operación-transferencia de consorcios.

Posibles medidas de Gestión de Costos de proceso:

1. Analizar y decidir sobre el proyecto, programa o el costo del sistema de Gestión de Portafolio.
2. Estimar y evaluar los costos de cada paquete de trabajo, incluidos los gastos generales.
3. Establecer el costo de supervisión y control de los elementos, así como la inflación y la gestión de divisas si es necesario.
4. Definir los objetivos de costos.
5. Calcular el uso de los recursos reales y los costos o gastos incurridos.
6. Tomar todos los cambios y demandas en cuenta.
7. Analizar las diferencias y las causas, comparación real frente al costo previsto.
8. Evolución del costo de Previsión y los costos finales.
9. Desarrollar y aplicar medidas correctivas.
10. Actualización de la estimación de gastos con respecto a los cambios.
11. Documentar las lecciones aprendidas y se aplican a los proyectos de futuro.

Posibles pasos del proceso de Gestión Financiera:

1. Analizar las opciones de financiamiento y modelos de proyecto, programa o portafolio.
2. Negociar con las posibles fuentes de fondos y determinar las condiciones adjuntas.
3. Seleccionar la fuente de financiación del proyecto.
4. Asignar un presupuesto, y analizar los pagos de anticipos.
5. Calcular el uso de recursos financieros y de efectivo y retiros en efectivo a los flujos del proyecto,

programa o portafolio.

6. Establecer y controlar los procesos y las autorizaciones de los pagos.
7. Reconocer o establecer y controlar la contabilidad y los sistemas de auditoría financiera.
8. Tener en cuenta la modificación del uso de los recursos financieros y los presupuestos disponibles durante el ciclo vital del proyecto.
9. Validar y manejar presupuestos, convirtiendo en costos incurridos.
10. Documentar las lecciones aprendidas y aplicar a los proyectos de futuro.

1.2.2 PMBOK

La Guía de los Fundamentos de la Dirección de Proyectos (Guía PMBOK) es un estándar reconocido a nivel internacional en la Gestión de Proyectos desarrollado por el Project Management Institute (PMI). Ésta define 9 áreas de conocimiento para la administración de proyectos. De las 5 fases que contiene para el desarrollo de un proyecto, la investigación se centrará en la fase de planificación en el proceso de Gestión de Costos.

La GC de los proyectos incluye los procesos involucrados en la planificación, estimación, preparación del presupuesto y control de costos de forma que el proyecto se pueda completar dentro del presupuesto aprobado (PMI, 2004). Los 3 procesos conocidos de la Gestión de Costos se muestran en la **Figura 3**.

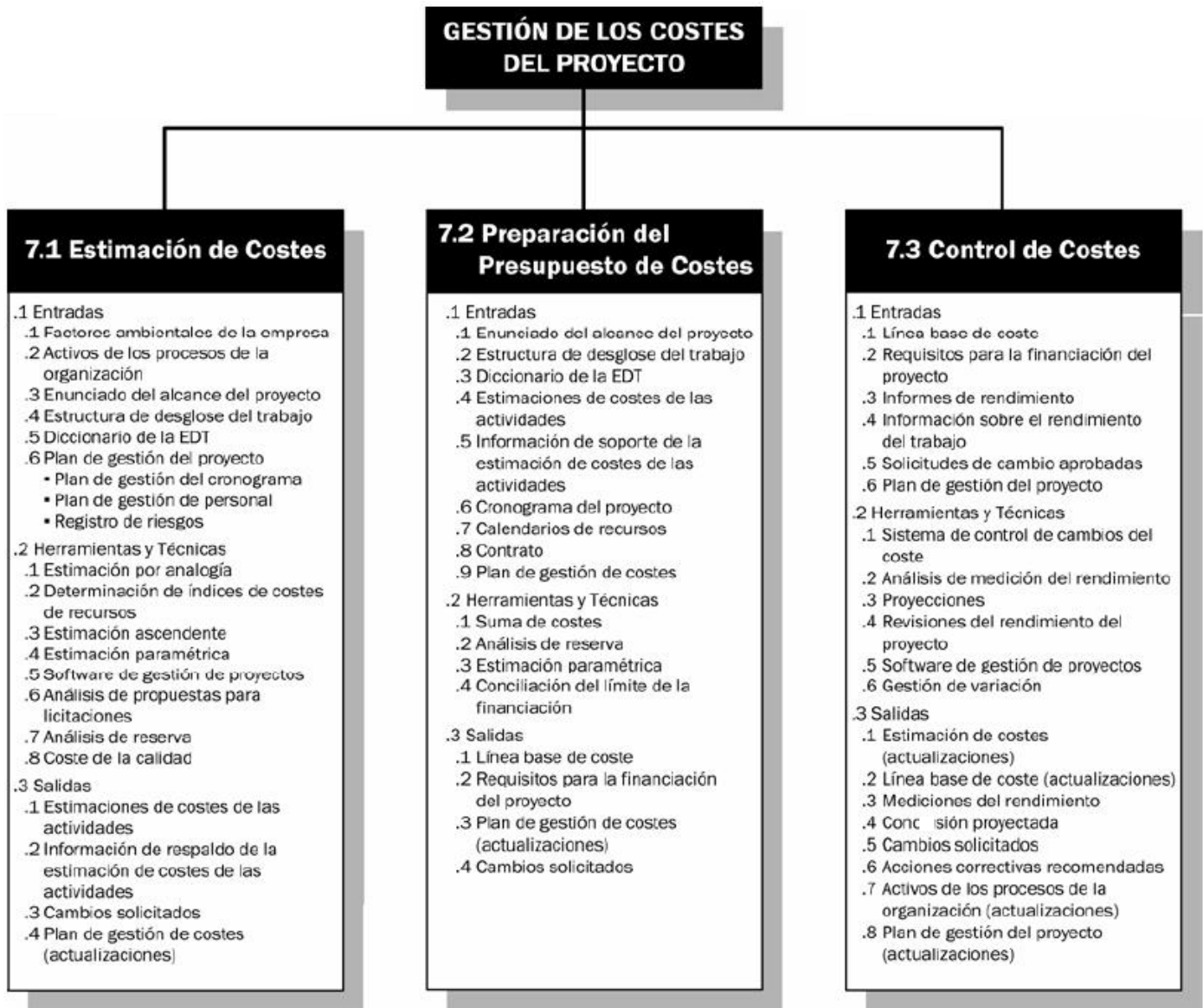


Figura 3 Descripción General de la Gestión de Costos del Proyecto (PMI, 2004).

Para el análisis del trabajo se tomarán en cuenta los procesos:

Estimación de Costos: desarrollar una aproximación de los costos de los recursos necesarios para completar las actividades del proyecto.

Preparación del Presupuesto de Costos: sumar los costos estimados de actividades individuales o paquetes de trabajo a fin de establecer una línea base de costo.

Estos procesos interactúan entre sí y también con los procesos de las demás Áreas del Conocimiento. Cada proceso puede involucrar el esfuerzo de una o más personas o grupos de personas, dependiendo de las necesidades del proyecto. Cada proceso tiene lugar por lo menos una vez en cada proyecto y se realiza en una o más fases, si el proyecto se encuentra dividido en fases.

Los procesos anteriores utilizan un conjunto de herramientas y técnicas para su ejecución. En este trabajo se analizan solamente dos siguiendo el plan de iteración de desarrollo para la primera versión del producto. A continuación se detallan los mismos.

Estimación Ascendente para Estimación de Costos: esta técnica implica estimar el costo de paquetes de trabajo individuales o actividades del cronograma individual con el nivel más bajo de detalle. Este costo detallado luego se resume o “acumula” en niveles superiores para fines de información y seguimiento. El costo y la exactitud de la estimación de costos ascendente en general están motivados por el tamaño y la complejidad de la actividad del cronograma o del paquete de trabajo individual. En general, las actividades con un esfuerzo asociado menor aumentan la exactitud de las estimaciones de costos de las actividades del cronograma (PMI, 2004).

Suma de Costos para Preparación del Presupuesto de Costos: las estimaciones de costos de las actividades del cronograma se suman por paquetes de trabajo de acuerdo con la Estructura de desglose de trabajo (EDT). Las estimaciones de costos de los paquetes de trabajo luego se suman para los niveles superiores de componentes de la EDT, tales como las cuentas de control, y finalmente para todo el proyecto.

La Gestión de los Costos del Proyecto se ocupa principalmente del costo de los recursos necesarios para completar las actividades del cronograma. Sin embargo, la Gestión de los Costos del Proyecto también debería considerar el efecto de las decisiones del proyecto sobre los costos del uso, mantenimiento y soporte del producto, servicio o resultado del proyecto. Por ejemplo, limitar el número de revisiones del diseño puede reducir el costo del proyecto a expensas de un aumento de los costos operativos del cliente. Esta visión más amplia de la Gestión de los Costos del Proyecto se denomina frecuentemente cálculo de costos del ciclo de vida.

La Gestión de los Costos del Proyecto contempla los requerimientos de información de los interesados en el proyecto. Los diferentes interesados medirán los costos del proyecto de diferentes maneras y en diferentes momentos. Por ejemplo, el costo de un elemento adquirido puede medirse cuando se toma o se compromete la decisión de la adquisición, se realiza el pedido, se entrega el elemento y se incurre o se

registra el costo real para fines de la contabilidad del proyecto. En algunos proyectos, especialmente los de menor alcance, la estimación de costos y la preparación del presupuesto de costos están tan estrechamente vinculados que se consideran como un proceso único, que puede ser realizado por una sola persona durante un período de tiempo relativamente corto. La posibilidad de influir sobre el costo es mayor en las primeras etapas del proyecto y esa es la razón por la cual la definición temprana del alcance es crítica (PMI, 2004).

Fundamentación de la selección PMBOK

El IPMA es un modelo que se construye sobre la metodología del PMI y un código deontológico. Como la metodología PMI está enfocada en los procesos y la metodología de Prince2 está basada en los productos a entregar. El modelo de IPMA-ICB se centra en las capacidades del gerente de proyecto y el PMBOK se especializa en la gestión de los procesos; por lo que tienen diferentes enfoques y lo que realmente se necesita para el análisis del Módulo de Gestión de Costos es la gestión de los procesos. Debido a lo anterior se selecciona como metodología base la propuesta por el PMBOK. Ésta brinda los principales elementos para llevar a cabo la Gestión de Costos, pero es responsabilidad de quien la utilice, la definición y aplicación de una estrategia para su uso. Además, no implica específicamente proyectos de desarrollo del software sino que está diseñado para la Gestión de Proyectos de cualquier tipo.

Para la Gestión de Portafolios de proyectos existen en el mundo actual un conjunto de sistemas propietarios y libres, de los cuales se realiza a continuación un estudio detallado.

1.3 Sistemas Informáticos relacionados con la Gestión de Portafolios de proyectos

La implantación de un sistema de Gestión de Portafolios de proyectos tiene como objetivo el seleccionar, priorizar, autorizar y gerenciar el trabajo de una organización. Estos brindan un mejor trabajo y control a las empresas, las cuales deben hacer una adecuada selección de la herramienta que utilizarán para satisfacer todas las necesidades existentes en la organización. A continuación se hace un estudio de algunos de los sistemas informáticos existentes en el mundo para este fin.

Rational Focal Point

IBM Rational Project Director gestiona y optimiza la ejecución de proyectos en todo el ciclo de vida. Es un proyecto colaborativo y tiene la solución de gestionar los recursos de manera más ágil para mejorar la

ejecución de proyectos, mientras que ayuda a los equipos a adaptarse a los cambios continuos a través de un enfoque integrado para la ejecución del ciclo de vida y la prestación de apoyo al proceso que madura con su organización.

Ventajas:

- Es guía de equipos en los procesos de mejores prácticas para aumentar la coherencia y la previsibilidad.
- Valida continuamente los costos, el alcance, los beneficios y riesgos, la redistribución de los recursos según sea necesario, en el cambio de prioridades.
- Mejora la toma de decisiones a través de la gestión de proyectos integrados.
 - Utiliza la visualización, establecimiento de prioridades, y la hoja de ruta única y las capacidades de planificación para crear planes que son valiosos y alcanzables.
- Prioriza y visualiza las necesidades del cliente para reducir los riesgos de minimización o eliminación durante la ejecución de proyectos de las capacidades más valiosas.
- Centraliza la información para escapar del caos de la gestión de correos electrónicos, documentos y hojas de cálculos, acelerando la capacidad de responder a las cambiantes condiciones del mercado y del negocio (Rational, 2010).

Desventajas:

IBM Rational Project Director a pesar de ofrecer todas las ventajas anteriores tiene limitado su acceso por ser de licencia privada, con un precio de \$686.00 USD (IBM, 2010) solo de licencia de usuario, sin contar con los impuestos no incluidos de los precios del IBM.

Primavera Enterprise Project Portfolio Management

Es el más potente, robusto y fácil de usar, es una solución para el mundo de prioridades, planificación, gestión y ejecución de proyectos, programas y portafolios. Primavera P6 Enterprise Project Portfolio Management es un sistema integrado de Gestión de Portafolio de proyectos (PPM), solución basada para satisfacer las necesidades de cada miembro del equipo, las responsabilidades y competencias. Ofrece una solución única para gestionar proyectos de cualquier tamaño, se adapta a los diferentes niveles de complejidad dentro de un proyecto, y de forma inteligente se adapta para satisfacer las necesidades de los diversos roles, funciones, niveles de habilidad o en su organización y en su equipo de proyecto.

Ventajas:

- Planifica programas y proyectos a gran escala.

- Asigna los mejores recursos y seguimiento de los progresos.
- Fomenta la colaboración en equipo.
- Gestiona eficazmente el tiempo, costo, recursos, contratos, y los cambios en una única solución.
- Proporciona visibilidad de la gestión de portafolios de proyectos a la empresa.
- Controla y visualiza los resultados de los proyectos frente al plan.
- Reduce el riesgo de incumplimiento.
- Integración con la gestión financiera y sistemas de gestión de capital humano (Oracle, 2010).

Desventajas:

Primavera Enterprise Project Portfolio Management de Oracle, este sistema resulta muy eficiente para la Gestión de Portafolios pero el ser de licencia privada dificulta su acceso. Tiene un alto valor de licencia de usuario: \$1.973 euros y como licencia de producto: \$1.794 euros (Oracle, 2010).

Microsoft Enterprise Project Management (EPM)

La solución EPM (Enterprise Project Management) de Microsoft Office es un entorno de proyecto completo de colaboración y portafolios. Es idónea para las empresas u organizaciones que necesiten un alto nivel de coordinación y estandarización entre los proyectos, administradores de proyectos y administración centralizada de los recursos.

EPM de Office incluye los siguientes productos de la familia Microsoft Office Project 2007 para proporcionar a las organizaciones una solución completa de administración de los Portafolios de los proyectos (PPM, Project Portfolio Management):

- Microsoft Office Project Professional 2007.
- Microsoft Office Project Server 2007.
- Microsoft Office Project Portfolio Server 2007.

Ventajas:

- Selecciona el portafolio más idóneo bajo las restricciones presupuestarias, empresariales y variantes que mejor se sintonicen con las estrategias empresariales de la organización.
- Evalúa y realiza el seguimiento eficazmente de los proyectos, programas y aplicaciones durante todo su ciclo de vida.
- Permite el incremento de la productividad de los empleados, duraciones de ciclos más cortas, costos reducidos y mejoras en la administración del tiempo, recuperación sustanciosa y positiva de la inversión.

- Ayuda a la organización a obtener más visibilidad, información detallada y control en todos los trabajos, a mejorar la toma de decisiones, la sintonía con la estrategia empresarial, a maximizar el uso de los recursos, así como a medir y ayudar a incrementar la eficacia operacional.
- Agiliza el desempeño de la administración de Project para ventajas competitivas. Ya sea proporcionando productos o servicios, todas las organizaciones necesitan alcanzar sus fechas de entrega, presupuestos y expectativas de sus accionistas. Para mantener alta la satisfacción y expectativas de los clientes, no hay espacio para errores o retrasos en los proyectos (Microsoft, 2010).

Desventajas:

La solución EPM (Enterprise Project Management) de Microsoft Office aparte de tener grandes ventajas, encontramos algunos inconvenientes como ser de licencia privada y tener un alto precio de: \$999.95 USD (Microsoft, 2010), además no ejecutarse en el Sistema Operativo Linux. También, un usuario tiene que gastar una cantidad razonable de tiempo en alcanzar la configuración predeterminada.

Comparación de los sistemas de código abierto

Código Abierto	Características	Desventajas
Project-Open	Herramienta web de gestión de proyectos muy completa. Características: pueden configurarse los módulos que desean y descartar los que no son necesarios: calendario, gestión de tareas, seguimiento de incidencias, control presupuestario, Wiki, foros, integración con ERPs, diagramas Gantt y una larga lista de funcionalidades.	Permite hoja de costos del sistema, el seguimiento de los gastos del proyecto y la creación de "paquetes de gasto". Estos paquetes permiten la creación de facturas, presupuestos y órdenes de compra, son actualmente de código cerrado y sólo disponible a través de la compra comercial, además no estiman los costos. En general, no se realiza el proceso de estimación de costos completamente.
Redmine	Herramienta web, desarrollada en Ruby on Rails. Características: gestión de tareas, incidencias, diagramas de Gantt, calendario de actividades, noticias, foros, wikis,	No incluye directamente ningún proceso de la Gestión de Costos. El presupuesto es un pluing para gestionar resultados de cada proyecto, es el cálculo de los

	gestión de ficheros y documentos, integración con Subversión, control de tiempos, generación de informes, entre otras, todo esto por proyecto.	indicadores claves de forma automática. Sólo se basa parcialmente en el proceso de presupuesto.
--	--	---

Tabla 1 Comparación de los procesos del área de conocimiento en sistemas de código abierto.

Resumen de los Sistemas Informáticos

Los sistemas propietarios descritos anteriormente resultan muy convenientes y completos en cuanto al tema de Gestión de Portafolios de proyectos, pero al país se le hace muy difícil su acceso debido a los precios elevados que presenta su uso. En el caso de EPM de Microsoft Office proporciona una solución completa, pero aparte de su precio tiene una dificultad para los países que trabajan con software libre y es que la herramienta no se ejecuta en sistemas GNU/Linux, es decir, no es multiplataforma. Las aplicaciones convencionales de PPM se dividen en dos grupos, los sistemas privados analizados anteriormente y los de código abierto.

Hoy en día, las herramientas de código abierto ofrecen la ventaja de una aplicación altamente personalizable, flexible y robusta, sin mirar más allá de la intención de centrarse en la ejecución del proyecto. Éstas pueden ser modificadas para satisfacer las necesidades explícitas de una organización, pero no cumplen con todos los procesos del área del conocimiento, (Ver **Tabla 1**).

Todos los sistemas analizados tanto de código abierto como privados no satisfacen las necesidades de la herramienta deseada, unos por su elevado costo y otros por la falta de funcionalidades básicas para la GC; es por ello que se realizará el análisis del Módulo de Gestión de Costos. Además, éste responde a la necesidad de alcanzar una verdadera independencia y soberanía tecnológica que garantice al país la posibilidad de contar con una herramienta de Gestión de Portafolios de proyectos, pues resulta muy beneficioso contar con una plataforma propia que puede ser comercializada e integrada a otros sistemas como el de Planificación de Recursos Empresariales (ERP, de Enterprise Resource Planning, por sus siglas en inglés) Cedrux.

1.4 Metodología y herramientas para la modelación de software

1.4.1 Metodología

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental, que ayuda a los desarrolladores a realizar un nuevo software (UBA, 2008). Sin el apoyo de una metodología, nunca se llegaría a satisfacer las necesidades de los clientes para los cuales se trabaja.

En el Centro se utiliza como pauta arquitectónica una metodología para el desarrollo de productos basada en el Modelo de Producción de Líneas de Productos de Software. Esta define las fases, actividades y artefactos que se deben generar durante el ciclo de vida del software.

Además, está basada en la adaptación de metodologías para el desarrollo de software en equipo como son SCRUM y OpenUP. Se centra en la organización propuesta para la aplicación del Modelo de Líneas de Productos de Software en CENTALAD (Lussón, 2009).

Actividades:

- Conformar Plan de Gestión de Requerimientos
- Conformar Glosario de Términos
- Conformar Modelo de Dominio
- Identificar Requisitos Funcionales y no Funcionales
- Conformar Especificación de Requisitos
- Especificación de los Casos de Uso
- Conformar los Casos de Prueba Operacionales

1.4.2 Lenguaje de Modelado

Lenguaje Unificado de Modelado (UML 2.0) tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue. Se utilizará UML 2.0 porque es el lenguaje de modelado más conocido en la actualidad, es el estándar internacional aprobado por la OMG (Object Management Group). Permite la especificación, visualización, construcción y documentación de elementos de la Ingeniería del Software.

UML es un lenguaje expresivo, claro y uniforme, definido para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos, pero sí mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios (Solenzal, 2006).

1.4.3 Herramienta CASE

Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering, CASE) es un tipo de ingeniería de software en la que se intenta aumentar la eficacia de sus procesos, al soportar la realización de las tareas con el uso de tecnologías (Pérez, 1999).

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero; permite modelar y documentar sus artefactos.

En el Centro está definido por política usar como herramientas CASE el Visual Paradigm for UML, que a continuación se analizará.

Visual Paradigm for UML

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Es una herramienta de modelado diseñada para un gran número de usuarios, incluyendo ingenieros de sistemas, analistas de sistemas, analistas de negocio y arquitectos de sistemas. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Paradigm, 2010).

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. A diferencia de muchas herramientas de modelado, pueden extenderse sus diagramas hechos desde el Visio y de Rational Rose. Además, se integra con IDEs (Entornos Integrados de Desarrollo) como Eclipse, JBuilder, NetBeans, IntelliJ IDEA, JDeveloper and WebLogic Workshop, esto ayuda a soportar la fase de implementación de desarrollo de software.

1.5 Ingeniería de Requerimientos

Según La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como (IEEE, 2010):

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 ó 2.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de construcción y producción de un software. Está basada en función de las necesidades planteadas por los clientes donde se descubre, documenta, analiza y definen los servicios o componentes de lo que se desea producir, además de las restricciones que tendrán el producto o software. Su principal tarea consiste en facilitar la comprensión de lo que el cliente requiera.

1.5.1 Características de los requerimientos

Los requerimientos de software presentan algunas particularidades, entre estas se encuentran:

- Deben ser especificados por escrito, como todo contrato o acuerdo entre dos partes.
- Tienen que ser posibles de probar o verificar, si no se puede probar el requerimiento como sabemos si cumplimos con él o no.
- Son descritos como una característica del sistema a entregar, en general es lo que el sistema debe de hacer y no como debe de hacerlo.
- Tienen que ser lo más abstracto y conciso posible, para evitar malas interpretaciones.
- Debe ser precisa, completa y clara la especificación de los requerimientos (Ingeniería, 2010).

Las características anteriores permiten tener una noción objetiva de lo correcto o no, de la especificación de los requerimientos extraídos a lo largo de la vida del software. La obtención correcta de los requerimientos puede llegar a describir con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento de un sistema.

1.5.2 Requisitos funcionales y no funcionales

Los requisitos se pueden clasificar en:

Requerimientos funcionales: son capacidades o condiciones que el sistema debe cumplir.

Requerimientos no funcionales: son propiedades o cualidades que el producto debe tener.

- Estas propiedades son las características que hacen al producto atractivo, usable, rápido o confiable.
- Están vinculados a los requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (Jacobson, 2000).

Un requerimiento no funcional especifica los criterios que se deben usar para juzgar el funcionamiento de un sistema, en lugar de un comportamiento específico. En general, los requerimientos funcionales definen lo que el sistema debería de hacer, mientras que los requerimientos no funcionales verifican cómo un sistema debería de ser. Una buena definición de requerimientos no funcionales es tan importante como los requerimientos funcionales; estos deben de ser apropiadamente definidos, analizados y trazados.

1.5.2.1 Categoría de los requerimientos no funcionales

Existen múltiples categorías para clasificar a los requerimientos no funcionales, las siguientes son las más representativas:

- **Requerimientos de Software:** debe mencionarse el software del que se debe disponer.
- **Requerimientos de Hardware:** enunciar los elementos de hardware con los que se disponen.
- **Restricciones en el diseño y la implementación:** especifica o restringe la codificación o construcción de un sistema; son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.
- **Requerimientos de apariencia o interfaz externa:** describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. Los requerimientos de apariencia se vuelven más importantes a medida que los productos de software se mueven hacia áreas más orientadas al consumidor.
- **Requerimientos de Seguridad:** es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:

- Confidencialidad: la información manejada por el sistema está protegida de acceso no autorizado y divulgación.
 - Integridad: la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
 - Disponibilidad: los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- **Requerimientos de Usabilidad:** describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello debe revisarse las especificaciones de los perfiles de usuarios y echarle un vistazo a las clasificaciones de sus niveles de experiencia.
- Los requerimientos de usabilidad se derivan de una combinación de lo que el cliente está tratando de lograr con el producto y lo que los usuarios finales esperan del mismo, estos elementos deben tenerse claro antes de escribirlos.
- **Requerimientos de Soporte:** abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de éste, así como lograr su mejoramiento progresivo y evolución en el tiempo (Ingeniería, 2010).

Con todos ellos se permitirá que el producto de software sea más aceptado por el usuario.

1.5.3 Captura de requerimientos

La captura de requerimientos es el proceso de identificar qué quiere el “cliente” del sistema propuesto. Hoy en día es una tendencia la poca comunicación entre los clientes y los desarrolladores de software, con motivo de esta situación se plantean un conjunto de técnicas que permiten mejoras en la captura de las funcionalidades del software.

Las técnicas que se usarán en la captura de requerimientos son las siguientes:

Introspección: esta técnica recomienda que el ingeniero de requerimientos se ponga en el lugar del cliente y trate de imaginar cómo desearía éste el sistema. Sobre la base de estas suposiciones comenzaría a recomendar al cliente sobre la funcionalidad que debería presentar el sistema. El problema radica en que un ingeniero no es un tipo normal de cliente, éste posee un conocimiento elevado por lo que podría recomendar más de lo que el cliente necesita.

Entrevistas: es un método común, no se entrevista a toda la gente que se relacionará con el sistema, sino a una selección de personas que represente a todos los sectores críticos de la organización, con el énfasis puesto en los sectores más afectados o que harán un uso más frecuente del nuevo sistema. Esta técnica permite la obtención de los actores que participan en el sistema, así como sus responsabilidades.

Discusiones: este tipo de entrevistas pretende que se sostenga una discusión con el cliente sobre su problemática para tratar de determinar en conjunto los requerimientos del sistema.

Talleres: los requerimientos tienen a menudo implicaciones desconocidas para las personas involucradas y a menudo no se descubren en las entrevistas o quedan incompletamente definidas durante la misma. Estas implicaciones cruzadas pueden descubrirse realizando un ambiente controlado, talleres facilitados por un analista del negocio, en donde las personas implicadas participan en discusiones para descubrir requerimientos, analizan sus detalles y las implicaciones cruzadas.

Prototipos: es una pequeña muestra de funcionalidad limitada, de cómo sería el producto final una vez terminado. Ayudan a conocer la opinión de los usuarios y rectificar algunos aspectos antes de llegar al producto terminado.

Casos de Uso: es una técnica bastante utilizada que captura cada una de las funciones del sistema y sobre la base de cada una de ellas especifica los requerimientos del mismo. Realiza pruebas tempranas de prototipos para minimizar cambios hacia el final del proyecto y reducir los costes finales (VIGO, 2010).

1.6 Patrones de Caso de Uso

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requerimientos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en la comunidad del desarrollo de software. Se presentan a modo de herramientas que permiten resolver los problemas que se les planteen a los desarrolladores de una forma ágil y sistemática. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en cómo modelar usos específicos (Ingeniería, 2009).

Los patrones de casos de uso a utilizar son los siguientes:

Concordancia (Commonality)

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

Reusabilidad: consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.

Adición: en el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

Especialización: contiene casos de uso del mismo tipo, estos son modelados como una especialización de casos de uso de tipo de uso común. Todas las acciones en estos casos de uso son heredadas por los casos de uso hijos, donde otras acciones serán adicionadas o acciones heredadas que serán especializadas. Este patrón es aplicable cuando la utilización de los casos de uso que han sido modelados son del mismo tipo, y este tipo debe hacerse visible en el modelo.

CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo: consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

Parcial: modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.

Múltiples actores

Roles diferentes: captura la concordancia entre actores manteniendo roles separados. Consiste de un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.

Roles comunes

Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del

caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

1.7 Conclusiones

Con el estudio realizado del estado del arte sobre la Gestión de Portafolios de proyectos de este capítulo, se puede concluir que se realizó un análisis general de los conceptos aplicables a la Gestión de Costos para sustentar la investigación, fundamentando una comparación entre los modelos PMBOK e IPMA para así seleccionar el más adecuado. Además, se explicó la metodología y herramienta que están definidas por el Centro. Para los sistemas informáticos que existen se realizó un estudio detallado de sus principales características, así como de sus ventajas y desventajas. Se estudió los conceptos básicos de la ingeniería de requerimientos, definiendo las técnicas de captura a utilizar.

Características del Sistema **2**

En el presente capítulo se ofrece una descripción del sistema propuesto por esta investigación, se especifican los actores, los casos de uso y el Modelo de Dominio.

Para el buen desarrollo del análisis del trabajo es necesario obtener los requerimientos funcionales y no funcionales, a partir de los cuales se identifican los casos de uso y posteriormente se describen los mismos.

2.1 Propuesta del Sistema

El Sistema Integral de Gestión de Portafolios de proyectos, pretende dar solución a la situación existente actualmente en la UCI y en el país en relación con el necesario incremento de la calidad en el proceso de Gestión de Proyectos, así sea de desarrollo de software o no. Tarea importante que depende en gran medida de la capacidad de Gestión de Proyectos de los recursos humanos, de la experiencia, de los procedimientos y estándares utilizados en la misma. Uno de los pasos esenciales para lograr dicho objetivo es desarrollar una plataforma que potencie la estandarización en el área de Gestión de Proyectos, específicamente la Gestión de Portafolios de proyectos, garantizando un proceso de toma de decisiones eficiente.

El sistema está orientado a automatizar los procesos descritos por el PMI para la Gestión de Portafolio y el PMBOK para la gestión operacional de los proyectos, combinándolos coherentemente de forma integral que posibilite a las empresas una gestión eficiente de sus portafolios.

Uno de los módulos necesarios para el desarrollo de este sistema es el Módulo de Gestión de Costos, el cual permitirá tener un control adecuado del presupuesto de la empresa, como también permitirá que los proyectos se lleven a cabo dentro del rango económico establecido.

El costo constituye un elemento indispensable en cualquier sistema de dirección económica y es fundamental para la correcta toma de decisiones en las entidades.

Para un mejor entendimiento de la Gestión de Costos relacionada con la Gestión de Proyecto se describen los siguientes términos del glosario que son conceptos básicos:

Presupuesto: identifica los diferentes costos y el monto para la iniciación de cualquier proyecto empresarial. Se refiere a recursos económicos y financieros necesarios para llevar a cabo las actividades o procesos, y para obtener los medios esenciales que deben calcularse, como el costo de realización y el costo de adquirir nuevos recursos (Mc.Graw-Hill Companies, 2007).

Gastos: expresan en términos monetarios, el monto total de los recursos tanto material, laboral como financiero, utilizados durante un período cualquiera, en el conjunto de la actividad empresarial. El concepto de gasto tiene un contenido amplio y refleja el consumo de cualquier recurso durante un período de tiempo, con independencia de su destino dentro de la empresa (Mc.Graw-Hill Companies, 2007).

Costos: es la magnitud de los recursos materiales, laborales y monetarios necesarios para alcanzar un cierto volumen de producción con una determinada calidad. El costo de la producción o los servicios constituye una parte de los gastos totales de la entidad ya que al costo se vinculan los gastos que se identifican con el proceso de producción o los servicios (Mc.Graw-Hill Companies, 2007).

Las actividades que se van a llevar a cabo por parte de un gerente financiero para darle cumplimiento a los procesos definidos en este módulo son los siguientes:

- Establecer la Estructura de Descomposición del Proyecto EDP (WBS).
- Asignarles el costo a las actividades planificadas del proyecto.
- Asignarle los costos reales al trabajo ejecutado.
- Realizar los cálculos de Valor Ganado.
- Reportar y analizar para tomar acciones.

El proceso de control es uno de los más importantes porque tiene como objetivo cerciorarse de que los hechos vayan de acuerdo con los planes establecidos. Influye sobre los factores que crean variaciones del costo y controla los cambios en el presupuesto del proyecto, además de evaluar el desempeño general frente a un plan estratégico.

Los procesos que contempla este módulo tienen explícitas algunas técnicas y herramientas, las cuales permiten un mejor entendimiento de cómo llevar a cabo el desarrollo en cada uno de los mismos.

Una de las técnicas que se utiliza en la visualización del estado del presupuesto del proyecto es la llamada curva "S" donde se expresa el valor ganado.

El valor ganado compara la cantidad de trabajo completado en un momento dado con la estimación realizada antes del comienzo del proyecto, para tener una medida del trabajo realizado, el que falta para finalizar el proyecto y a partir del esfuerzo invertido, el jefe de proyecto puede estimar los recursos que se emplearán para finalizar el proyecto. También se puede estimar en cuánto tiempo se completaría el proyecto si se mantienen las condiciones con las que se elaboró el cronograma o considerando si se mantienen las condiciones que se presentaron durante el desarrollo del proyecto.

En la **Figura 4** se representa una gráfica de curva "S" de valores acumulados contra el tiempo, donde se muestra el costo real, el programado y el valor ganado.



Figura 4 Gráfica de desviación del presupuesto programado, utilizando la gráfica curva "S".

Para el análisis del módulo también se tienen en cuenta los procesos gerenciales, los cuales mejorarán el rendimiento del proyecto. Aquí se contemplan una serie de actividades paralelas a la ejecución, que permiten a través de mediciones periódicas, conocer el estado de avance o retraso de las fases y del presupuesto de un proyecto.

- Comparación del progreso real contra el plan trazado.
- Determinar las desviaciones.
- Tomar las acciones correctivas.
- Retroalimentación.

Estos procesos muestran continuamente si se alcanzan o no los resultados definidos por el plan estratégico de la empresa. Con vistas a fortalecer el control económico de la organización se requiere una forma centralizada de gestionar los costos asociados a los proyectos, actualmente se utilizan los centros de costo con este fin.

Un centro de costo es la subdivisión mínima en el proceso del registro contable en la cual se acumulan los gastos de la actividad productiva de la empresa a los fines de facilitar la medición de los recursos utilizados y los resultados económicos obtenidos. La determinación de los centros de costo debe hacerse centrando la atención en los objetivos a lograr con la información que ellos proporcionan, como base para la toma de decisiones, por lo cual, debe tratarse siempre que sea posible que se correspondan con un área de responsabilidad claramente delimitada (Leyva, 2009). En otras palabras, se divide el proyecto en centro de costo y a partir de cada uno de ellos se tiene un mejor control de los costos del proyecto.

Todo el estudio realizado del Módulo de Gestión de Costos facilitó una mejor comprensión de lo que debería hacer el sistema en esta área.

A continuación se describen los artefactos obtenidos como resultado de la realización del flujo de Análisis. Éstos posibilitarán un mejor entendimiento de lo que debería hacer el sistema, proporcionando las bases necesarias para su posterior desarrollo.

2.3 Modelo de dominio

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción (Pressman, 2002). Captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

El Modelo de Dominio puede ser tomado como el punto de partida para el diseño del sistema, ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

2.3.1 Diagrama conceptual del modelo de dominio

El Modelo de Dominio presentado a continuación (Ver **Figura 5**) tiene como objetivo contribuir a la comprensión del análisis del Módulo de Gestión de Costos. Éste constituye una parte del modelo del Sistema Integral de Gestión de Portafolios de proyectos.

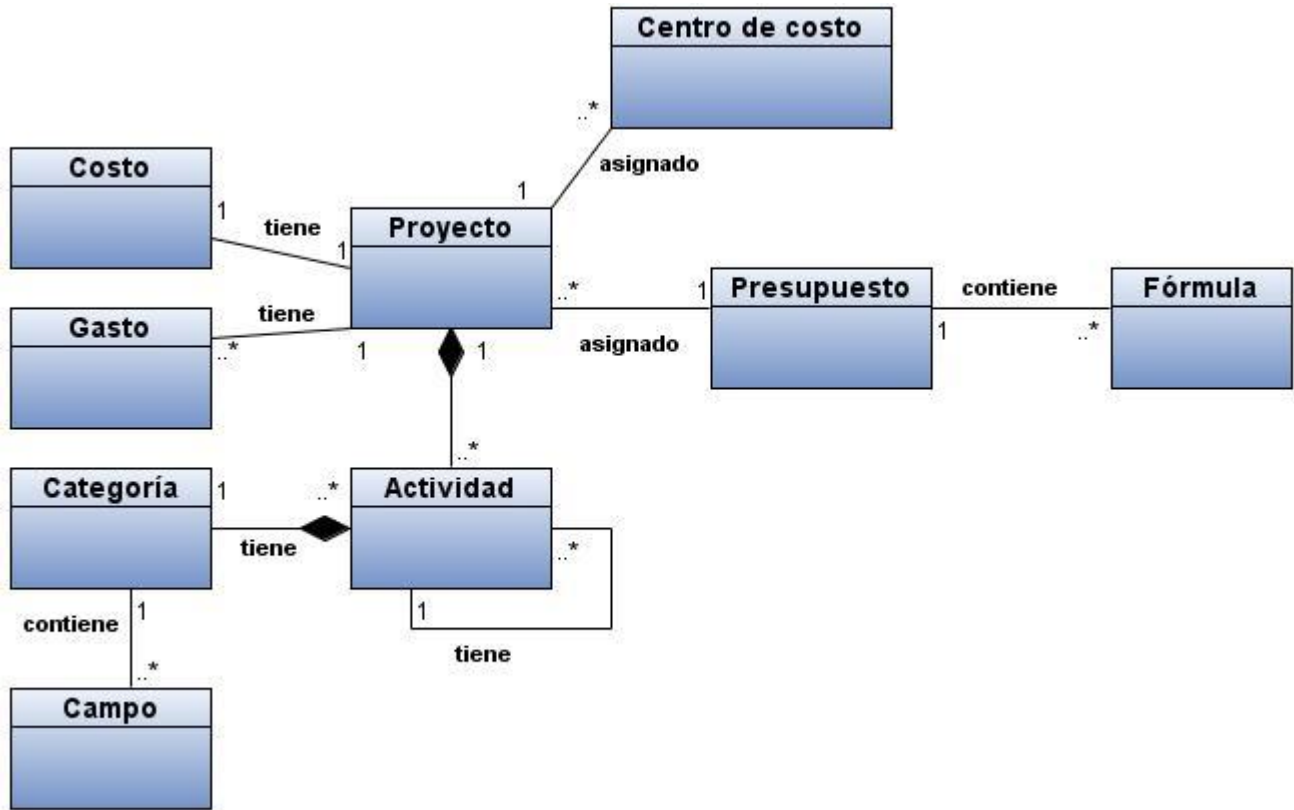


Figura 5 Diagrama de Modelo de Dominio

En el diagrama anterior se expresa la relación de las entidades del área de costo que existen en un proyecto. Un presupuesto puede ser asignado a uno o varios proyectos, siendo este el dinero con que contará para realizar el trabajo; además contiene fórmulas necesarias para los cálculos a realizar. Los Centros de costo se le asignan al proyecto para tener un mejor control del uso de su presupuesto. Un proyecto tiene un costo que se tendrá al producir el producto, además de los gastos realizado. Un proyecto está formado por un conjunto de actividades, las cuales tienen categoría que pueden ser hito, etapa, y la misma contiene campos.

Se identifican como conceptos utilizados en el diagrama a los siguientes términos del glosario:

Actividad: es una tarea simple asociada a un proyecto.

Campo: definiciones que pertenecen a una categoría de actividad, estos tienen un nombre y un valor.

Categoría: tipo de clasificación de las actividades que determinan su comportamiento, algunas clasificaciones son: hito, etapa, otras.

Centro de costo: subdivisión mínima en el proceso del registro contable en la cual se acumulan los gastos de la actividad productiva de la empresa a los fines de facilitar la medición de los recursos utilizados y los resultados económicos obtenidos.

Costo: gasto económico que representa la fabricación de un producto o la prestación de un servicio.

Fórmula: números y símbolos que muestran cómo obtener algo. Un tipo especial de ecuación que muestra la relación entre diferentes variables.

Gasto: monto total de los recursos tanto material, laboral como financieros, utilizados durante un período cualquiera, en el conjunto de la actividad empresarial.

Presupuesto: identifica los diferentes costos y el monto para la iniciación de cualquier proyecto empresarial.

Proyecto: esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

2.4 Especificación de los requisitos de software

El artefacto de especificación de requisitos software o simplemente ERS es una descripción completa del comportamiento del sistema que se va a desarrollar. Además, incluye un conjunto de casos de uso que describe todas las iteraciones que tendrán los usuarios en el software. El documento debe contener una lista detallada y completa de los requerimientos que debe cumplir el sistema (Pressman, 2002).

2.4.1 Requisitos funcionales

En la obtención de los requerimientos se utilizaron las técnicas de introspección, entrevistas, discusiones, talleres, prototipos y casos de uso con el fin de realizar una correcta captura de la información. Aplicando la técnica introspección a las herramientas estudiadas en el capítulo anterior se obtuvo un listado previo de 24 requerimientos, permitiendo obtener una suposición de lo que debería tener el sistema.

Por otro lado, en la entrevista realizada (Ver **Anexo 1**) al personal especializado en el tema de Gestión de Portafolio de proyectos y Gestión de Costos, se puede mencionar la Directora Económica Olga Leyva Monzón en la empresa ALBET, así como el personal de la oficina Económica, la oficina del Grupo de Mercadotecnia que estudia la factibilidad entre ellos la financiera, y la oficina de Gestión de Proyectos; además del económico del Centro y el jefe del departamento de Ciencias Empresariales de la

especialidad, se obtuvieron 33 requerimientos, de ellos se reafirmaron 20 del listado previo. A continuación se presentan los resultados de las preguntas realizadas (Ver **Figura 6**):

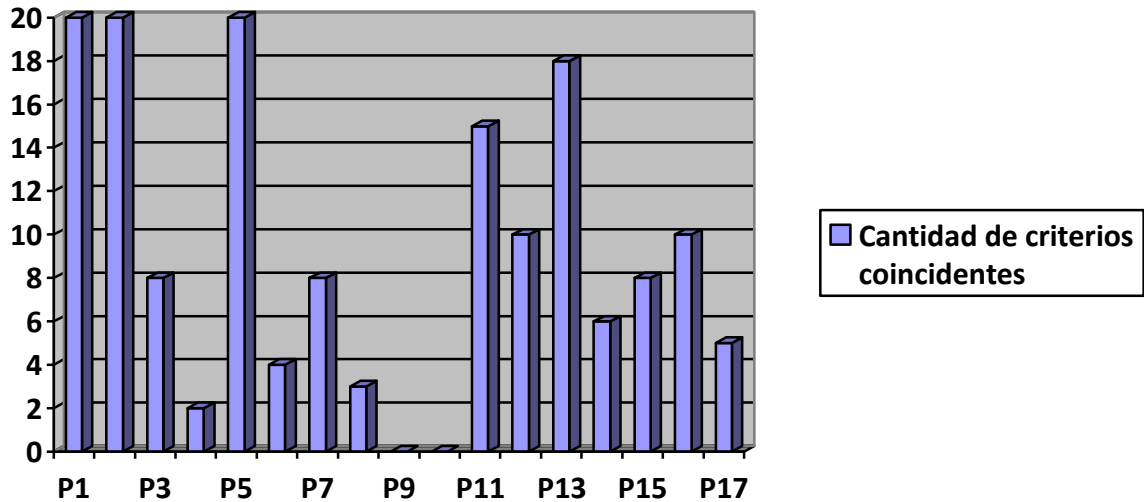


Figura 6 Resultados de la entrevista por preguntas

En esta gráfica se muestran los resultados de la entrevista, de las preguntas contra el personal entrevistado. Las barras representan cuántas personas en cada pregunta realizada tienen un mismo criterio de respuesta.

Como resultado del análisis de la información de la entrevista, se concluyó que no se tienen en cuenta ninguna técnica para realizar los procesos que contiene el área de costo. Se detectó que contemplan en cada corte del proyecto si la estimación está alineada al presupuesto, esto se hace para que no se obtengan pérdidas por lo que es necesario controlar que los costos no sobrepasen el presupuesto establecido. Las herramientas que se utilizan en la parte económica es el Excel y para Gestión de Proyecto es el Open-Project.

La realización de la entrevista facilitó un mejor entendimiento de las necesidades del control de costo de cualquier empresa, en especial ALBET, así como la obtención de nuevos requerimientos. Además, se realizaron discusiones del listado de los requerimientos obtenidos con los especialistas y el equipo de desarrollo, llegando a un acuerdo entre ambas partes de cómo realizar los procesos de estimación y presupuesto de los costos. A partir de este acuerdo se obtuvieron 8 nuevos requerimientos que se añadieron al listado actual. También se realizaron los prototipos de interfaz como vía para visualizar a los

especialistas, para mejor entendimiento de lo que se quiere obtener. Estos permitieron demostrar los conceptos, así como revelar errores y omisiones que existían en los requerimientos propuestos.

A continuación se presenta el listado de requerimientos obtenido.

Gestionar Costo a un proyecto

RF1: Asignar presupuesto a un proyecto.

El sistema debe permitir asignarle el presupuesto a un proyecto. Dato de entrada:

-Presupuesto del proyecto (Valor Monetario)

RF2: Adicionar costo a un proyecto.

El sistema debe permitir adicionar el costo actual del proyecto. El dato a introducir es:

-Costo (Valor Monetario)

RF3: Modificar costo a un proyecto.

El sistema debe permitir modificar de un proyecto el costo previamente seleccionado.

RF4: Mostrar costo a un proyecto.

El sistema debe permitir mostrar el costo asociado de un proyecto. El dato mostrado es:

-Costo (Valor Monetario)

RF5: Calcular total de los gastos de un proyecto.

El sistema debe permitir determinar el desembolso general que realiza un proyecto para desarrollar sus actividades.

RF6: Calcular el costo de las actividades padres de forma ascendente.

El sistema debe calcular de abajo arriba la suma de unidades previstas para cada tarea en un proyecto para ver si está más allá de su presupuesto estimado.

RF7: Calcular el total de gastos del proyecto agrupado por concepto.

El sistema debe permitir calcular los gastos agrupados por concepto del proyecto que pueden ser gastos administrativos, hospedajes, entre otros.

Gestionar Costo de Portafolio de proyecto

RF8: Mostrar el total de los costos de cada proyecto.

El sistema debe permitir conocer el total de los costos de cada proyecto. El dato a mostrar:

-Costo total (Valor Monetario)

RF9: Calcular el total de gastos del portafolio agrupados por concepto.

El sistema debe permitir calcular los gastos agrupados por concepto del portafolio que pueden ser gastos administrativos, hospedajes, entre otros.

RF10: Calcular el total de los costos de todos los proyectos del portafolio.

El sistema debe permitir conocer el total de los costos de todos los proyectos del portafolio.

Gestionar Reportes

RF11: Evaluar el rendimiento presupuestario hasta la fecha.

El sistema muestra la diferencia entre el costo previsto y el costo total del proyecto para ver si se sitúa por debajo o por encima del presupuesto, o si se ajusta exactamente a él, se debe mostrar en forma de gráfica de barra.

RF12: Mostrar el presupuesto planeado de los proyectos.

El sistema debe mostrar los costos planeados del proyecto. El dato a mostrar:

- Proyectos (Cadena de caracteres)
- Presupuesto planeado (Valor Monetario)

RF13: Mostrar el cumplimiento del presupuesto.

El sistema debe permitir conocer el estado actual del cumplimiento del presupuesto de cada proyecto del portafolio, debe mostrar una barra con el por ciento del presupuesto, donde se resaltará con un color diferente el por ciento cumplido hasta el momento.

RF14: Evaluar el rendimiento presupuestario hasta la fecha de una actividad.

El sistema muestra la diferencia entre el costo previsto y el costo total de los recursos para ver si se sitúa por debajo o por encima del presupuesto, o si se ajusta exactamente a él. El campo a mostrar es:

- Nombre de Actividad (Cadena de caracteres)
- Rendimiento (Cadena de caracteres)

Que puede ser Alto (cuando está por debajo), Medio (cuando está ajustado a él) o Bajo (cuando está por encima) en dependencia a donde se sitúa la diferencia de los costos anteriores.

RF15: Mostrar el trabajo total presupuestado para cada proyecto

El sistema debe mostrar los costos totales presupuestados para todos los proyectos. El dato a mostrar:

- Presupuesto total (Valor Monetario)
- Proyectos (Cadena de caracteres)

RF16: Mostrar los gastos programados restantes.

El sistema debe mostrar el costo restante que se producirán al completar el trabajo restante programado de todos los proyectos del portafolio. El campo a mostrar:

-Gasto restante (Valor Monetario)

Gestionar Reportes Financieros

RF17: Generar reporte Distribución presupuestaria de proyectos.

El sistema debe generar reportes de la distribución presupuestaria de proyectos. Los datos de a mostrar son:

-Nombre de Portafolios (Cadena de 10 a 20 caracteres)

-Número de Proyectos (Entero)

-% Sobre Total de Proyectos (Valor Numérico %)

-Presupuesto Planificado (Valor Monetario)

-% Sobre Presupuesto Total (Valor Monetario %)

RF18: Crear reportes gráficos de Flujo de caja.

El sistema debe crear una gráfica de Flujo de caja. Sus parámetros son:

-Costo (Valor Monetario)

-Costo acumulado (Valor Monetario)

Los parámetros son contra los años de las tareas.

RF19: Crear reportes gráficos de Costo presupuestado.

El sistema debe crear una gráfica de Costo presupuestado. Sus parámetros son:

-Costo presupuestado (Valor Monetario)

-Costo previsto (Valor Monetario)

-Costo (Valor Monetario)

-Costo real (Valor Monetario)

Los parámetros son contra los años de las tareas.

RF20: Crear reportes gráficos de Costo previsto.

El sistema debe crear gráfica de Costo previsto. Sus parámetros son:

-Costo previsto (Valor Monetario)

-Costo real (Valor Monetario)

Los parámetros son contra los años de las tareas.

RF21: Crear reportes gráficos de Resumen de costos de los recursos.

El sistema debe crear gráfica de Trabajo del presupuesto. Su parámetro es:

-Trabajo (Valor Numérico)

Gráfico en forma de pastel.

RF22: Crear informe del presupuesto.

El sistema debe crear un informe del presupuesto. Sus campos son:

-ID (Entero)

-Nombre de actividad (Cadena de 10 a 20 caracteres)

-Costo (Valor Monetario)

-Costo total (Valor Monetario)

-Costo previsto (Valor Monetario)

RF23: Crear informe de valor acumulado.

El sistema debe crear un informe de valor acumulado. Sus campos son:

-ID (Entero)

-Nombre de actividad (Cadena de 10 a 20 caracteres)

-Valor planeado: PV (CPTP: costos presupuestado del trabajo programado) (Valor Monetario)

-Valor acumulado: VA (CPTP) (Valor Monetario)

-AC (CRTR: costo real del trabajo realizado) (Valor Monetario)

-VP: variación de programación del valor acumulado (Valor Monetario)

- VC: variación del costo del valor acumulado (Valor Monetario)

RF24: Mostrar valor ganado de los proyectos.

El sistema debe mostrar una gráfica del valor ganado, es una técnica de gestión de proyectos que permite controlar la ejecución de un proyecto a través de su presupuesto y de su calendario de ejecución.

Se representa por una curva S en una gráfica lineal de tiempo contra valores acumulados, donde se muestran las curvas de costo real (CRTR) y el programado (CPTP), además del valor ganado (CPTR).

Sus bases de cálculo son:

-CPTP: Costo Presupuestado del Trabajo Programado

-CPTR: Costo Presupuestado del Trabajo Realizado

-CRTR: Costo Real del Trabajo Realizado

Se obtiene:

-VP: Variación del Plazo: $VP = CPTR - CPTP$

-VC: Variación de Costo: $VC = CPTR - CRTR$

Gestionar Actividad

RF25: Adicionar el costo a un recurso.

El sistema debe permitir adicionar el costo a un recurso de una actividad. Donde sus campos son:

-Unidades (Valor Numérico)

-Costo (Valor Monetario)

RF26: Modificar el costo de un recurso.

El sistema debe permitir modificar el costo de un recurso previamente seleccionado.

RF27: Eliminar el costo de un recurso.

El sistema debe permitir eliminar el costo de un recurso previamente seleccionado.

RF28: Adicionar el costo de una actividad.

El sistema debe permitir adicionar el costo de una actividad. El dato de entrada:

-Costo (Valor Monetario)

RF29: Modificar el costo de una actividad.

El sistema debe permitir modificar el costo previamente seleccionado de una actividad.

RF30: Eliminar el costo de una actividad.

El sistema debe permitir eliminar el costo previamente seleccionado de una actividad.

RF31: Calcular el costo restante de una actividad.

El sistema muestra los gastos programados restantes que se producirán al completar el trabajo programado, que se calcula:

Costo restante = (Trabajo restante * Tasa estándar) + Costo de horas extra restante.

Gestionar Campo Personalizado

RF32: Personalizar el costo del campo en la columna seleccionada.

El sistema debe permitir personalizar la columna. Donde sus campos son:

-Campo (actividad, recurso)

-Tipo (fecha, fin, costo, otros)

-Atributos personalizados (ninguno, fórmulas)

-Cálculos de las filas de resumen de grupo y tarea (ninguno, resumen, otros)

RF33: Editar fórmula de costo.

El sistema debe permitir editar fórmula al campo creado. Donde sus campos son;

-Insertar (variables, funciones)

-Área de texto para la fórmula

-Área de opciones para la fórmula

RF34: Calcular suma de todos los valores numéricos.

El sistema debe permitir calcular la suma de todos los valores numéricos.

RF35: Calcular el promedio de todos los valores numéricos.

El sistema debe permitir determinar el promedio de todos los valores numéricos.

RF36: Calcular el margen de utilidad.

El sistema debe permitir calcular la utilidad para ver si varía los costos, es decir, ver si lo real fue mayor que lo planificado. Su fórmula es: ingresos (precio) – gastos (presupuesto o lo que costará). El sistema muestra la utilidad en un campo:

-Utilidad (Valor Monetario)

RF37: Calcular el Flujo de Caja.

El sistema debe permitir calcular el flujo de caja que es la diferencia entre el flujo monetario recibido y el flujo monetario emitido. Su Formula es: costo-gastos. El sistema muestra el resultado en un campo:

-Flujo de Caja (Valor Monetario)

Importar y Exportar

RF38: Exportar la información financiera de los proyectos del portafolio.

El sistema debe exportar como PDF informes o gráficos con la información de las finanzas de los proyectos del portafolio.

RF39: Importar la información financiera de los proyectos del portafolio.

El sistema debe importar informes o gráficas de los proyectos del portafolio con la información de las finanzas.

Funcionalidades Generales

RF40: Guardar Línea Base de los costos del proyecto.

El sistema debe guardar una plantilla con el presupuesto distribuido en el tiempo que se utiliza como base respecto a la cual se puede medir, supervisar y controlar el rendimiento general del costo en el proyecto.

El presupuesto distribuido en el tiempo se obtiene sumando los costes estimados por período.

RF41: Crear Centro de costos.

El sistema debe permitir crear centros de costo que es la subdivisión mínima de un proyecto en el proceso del registro contable, en la cual se acumulan los gastos de la actividad productiva para facilitar la medición de los recursos utilizados y los resultados económicos obtenidos. Sus campos son:

- Nombre de Portafolio (Cadena de caracteres)
- Nombre de proyecto (Cadena de caracteres)
- Nombre del Centro de costo (Cadena de caracteres)
- Costo (Valor Monetario)
- Gasto (Valor Monetario)
- Precio de venta (Valor Monetario)
- Margen de utilidad (Valor Monetario)

2.4.2 Requisitos no funcionales

Usabilidad

RNF1: Debe poseer una interfaz agradable para el cliente de acuerdo con los estándares de diseño.

RNF2: Mostrar la información de forma lógica y correctamente estructurada.

RNF3: Puede ser usado por cualquier persona con conocimientos básicos sobre el uso de la computadora y un ambiente Web en sentido general.

Fiabilidad

RNF4: El sistema se debe recuperar en un tiempo prudencial de acuerdo con la anomalía ocurrida.

RNF5: El sistema debe hacer salvadas automáticas de la información.

Eficiencia

RNF6: Los tiempos de respuesta de la aplicación no deben exceder de los $300\text{ms} \pm 100\text{ms}$ para el 60% de las peticiones de consulta.

RNF7: Los tiempos de respuesta de la aplicación no deben exceder de los $900\text{ms} \pm 400\text{ms}$ para el 60% de las peticiones de escritura.

RNF8: La aplicación debe soportar la conexión concurrente de 1.000 peticiones en una ventana de 1s.

Soporte

RNF9: La aplicación contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

Hardware

RNF10: Se requiere de un servidor de 2 GB de RAM como mínimo y 4 GB de espacio libre en disco duro.

RNF11: Todas las computadoras implicadas, tanto para la administración como las de los usuarios, deben estar conectadas a una red y tener al menos 1 GB de RAM.

Software

RNF12: El sistema debe ser desarrollado en Lenguaje de script PHP 5 o superior.

RNF13: Para la instalación del servidor:

- Servidor Web Apache 2.
- Servidor de Base de Datos Postgresql 8.4.
- Sistema Operativo Linux: Ubuntu 9,10 (recomendado o superior) o Microsoft Windows.

RNF14: Para interpretación por el cliente:

- Internet Explorer v6, Internet Explorer v7 (recomendado) o superior.
- Mozilla Firefox 2.* (recomendado) o superior (recomendado).

Disponibilidad

RNF15: El sistema deberá estar disponible las 24 horas de los 7 días a la semana

Restricciones de diseño

RNF16: El software contará con las siguientes restricciones:

- Se utilizará framework de trabajos como el Zend Framework, ERP y Doctrine para el desarrollo.
- No se debe usar funciones propias de algún sistema operativo.
- Se utilizará Visual Paradigm for UML como herramienta CASE para el modelado y obtención de los diferentes artefactos que requiere el software.

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

RNF17: El sistema debe contar con una ayuda, que permita encontrar las funcionalidades fácilmente.

RNF18: El sistema debe contar con un manual de usuario, que permita conocer las características y las formas de funcionamiento del software.

Interfaz

RNF19: La aplicación contendrá distintas interfaces que servirán para el intercambio de información con la misma.

- Las interfaces de usuario implementadas serán Web.
- La interfaz de este sistema debe contar con un diseño sencillo y al mismo tiempo permitir la interpretación correcta de la información.

- El Sistema se comunicará con los gestores de bases de datos mediante protocolo TCP/IP.
- Los clientes accederán al sistema vía HTTP/HTTPS.

2.5 Definición de los casos de uso

Los diagramas de CU se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo (Braude, 2003). En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

2.5.1 Definición de los actores

A continuación, en la **Tabla 2** se muestra la descripción de los actores obtenidos del Módulo de Gestión de Costos; y en la **Figura 7** se representa el diagrama de los mismos.

Tabla 2 Actores del Sistema

Actor	Descripción
Usuario	El usuario se encargará de realizar las acciones del sistema.
Gerente general	El gerente es el encargado de gestionar los datos de las acciones centralizadas de su organismo, es decir, es el responsable de gestionar el presupuesto de portafolios, programas, proyectos y otras funciones.
Planificador	Encargado de acceder a los niveles del sistema que le permita realizar funcionalidades básicas y generales.
Económico	Es el encargado de actualizar la situación financiera del ente adscrito al que él pertenece. Participa en la elaboración del presupuesto de gastos, costos y demás funciones básicas

necesarias de las finanzas.

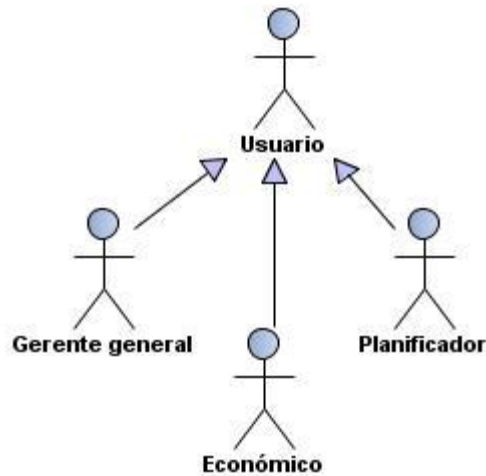


Figura 7 Diagrama de actores

2.5.2 Diagrama de Casos de Uso del Sistema

El diagrama de paquetes permite dividir un modelo en partes manejables mediante la agrupación de clases, CU u otros paquetes (Pressman, 2002).

A continuación se presenta el diagrama de paquetes obtenido del Módulo de Gestión de Costos (Ver **Figura 8**) que permitió dividir el sistema en partes, generalmente independientes para definir subsistemas; también se muestra los diagramas de Casos de Uso de cada paquete.

Diagrama de Paquetes

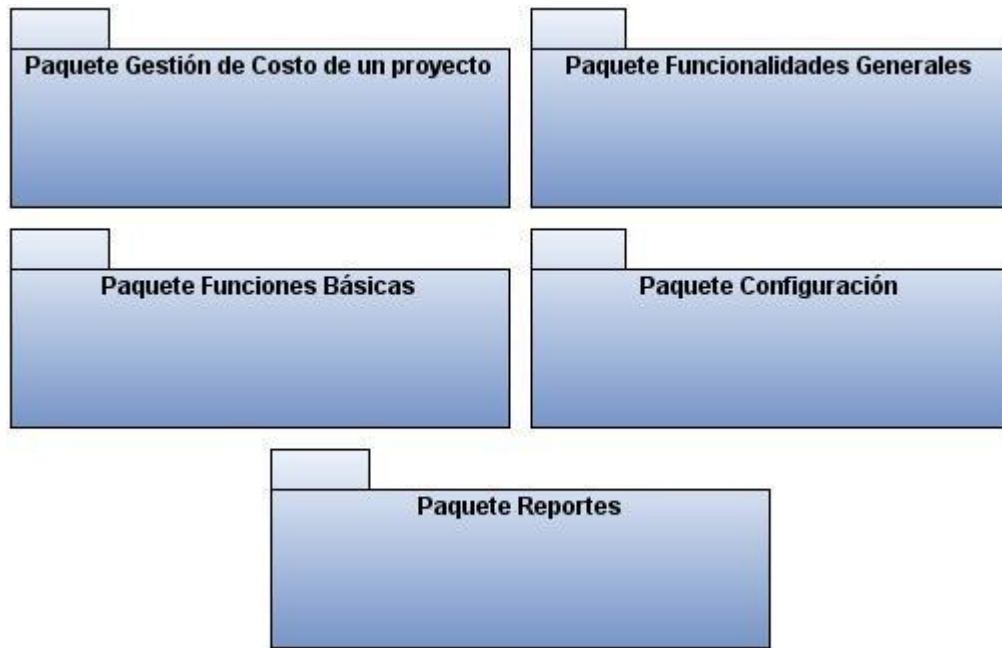


Figura 8 Diagrama de Paquetes

El diagrama anterior representa la descomposición jerárquica lógica del sistema, este cuenta con 5 paquetes que contienen un conjunto de casos de uso. Los paquetes Funcionalidades básicas y Gestión de costo de un proyecto son los más importantes porque gestionan los costos de las actividades, recursos y proyectos. También se cuenta con el de Configuración que se encarga de personalizar fórmulas y asignar fórmulas, además se tiene Funciones generales que engloban las funcionalidades adicionales que complementan la Gestión de Costos, éste contiene el importante CU Crear Centro de costos, así como el paquete Reportes que es auxiliar porque agrupa todos los reportes tanto financieros o no.

Diagramas de Casos de Uso de cada paquete

Los casos de uso describen el flujo de sucesos entre el actor y el sistema, auxiliados de prototipos de interfaz para mejor entendimiento. El proceso de selección de los CU más significativos está guiado por los riesgos que se desean mitigar. Para priorizar los casos de uso se clasificaron de acuerdo con el impacto que tienen en la arquitectura, estos pueden ser:

Críticos: son aquellos CU más importantes para los usuarios, ya que cubren las principales tareas o funciones que el sistema ha de realizar.

Secundarios: estos sirven de apoyo a los CU críticos, los mismos involucran funciones secundarias y tienen un impacto más sencillo, pero deben implementarse pronto porque responden a requisitos de interés para los usuarios.

Auxiliares: no son claves para la arquitectura y completan casos de usos críticos o secundarios.

Opcionales: estos responden a funcionalidades que pueden o no estar en la aplicación, pero no son imprescindibles en las primeras versiones (Ingeniería, 2010).

Se presentan a continuación algunos de los diagramas de CU de cada paquete, en los cuales se encuentran los casos de uso que corresponden a cada uno. Además, se muestra la descripción de los CU más significativos, correspondientes a estos diagramas. El resto de los CU se encuentran en el artefacto Modelo de CU localizado en el expediente del proyecto (Rondón, 2010).

Diagrama de Casos de Uso del paquete Gestión de Costo de un proyecto.

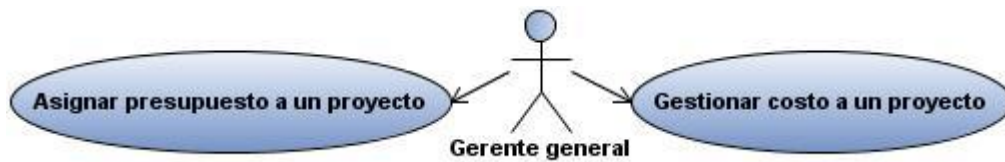


Figura 9 Diagrama de Casos de Uso del paquete Gestión de Costo de un proyecto

El diagrama anterior tiene dos casos de uso con clasificación de críticos, de los cuales solo se pondrá el Gestionar costo a un proyecto por la complejidad del gestionar.

Descripción del Caso de Uso Gestionar costo a un proyecto

Caso de Uso:	Gestionar costo a un proyecto
Actores:	Gerente general
Resumen:	<p>El caso de uso se inicia cuando el actor selecciona la opción “Información del Proyecto”, el cual le permite Adicionar costo a un proyecto, Modificar costo a un proyecto, Mostrar costo a un proyecto.</p> <p>El caso de uso termina una vez finalizada alguna de las opciones que permite.</p>
Precondiciones:	<p>El sistema debe estar ejecutándose correctamente.</p> <p>El sistema debe tener creado al menos un proyecto.</p>

Referencias	RF2, RF3, RF4	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El actor selecciona en el menú Proyecto la opción "Información del Proyecto".		2- El sistema muestra los datos de la interfaz Información del Proyecto, visualizando el campo costo que permite: -Adicionar costo a un proyecto -Modificar costo a un proyecto -Mostrar costo a un proyecto
Sección "Adicionar costo a un proyecto"		
1- El actor suministra el costo actual del proyecto. 2- Selecciona Aceptar.		3- El sistema verifica que no falten los datos obligatorios.
		4- El sistema valida los datos entrados. 5- El sistema persiste la información. 6-El caso de uso finaliza.
Prototipo de Interfaz		

Información del Proyecto

Nombre del proyecto:

Fecha inicio: Fecha fin:

Estado del proyecto: Tipo de Proyecto:

Enunciado del Proyecto:

Costo del proyecto:

Costo actual:

Flujo Alternativo al paso 2 "Operación Cancelada"

2. a El actor cancela el flujo antes de culminar el caso de uso.	2. b Se retorna a la pantalla anterior sin guardar ningún cambio.
--	---

Flujo Alternativo al paso 3 "Datos obligatorios"

	3. a El sistema verifica que faltan los datos en los campos obligatorios.
	3. b El sistema muestra símbolos al lado de los campos que son obligatorios.

Flujo Alternativo al paso 4 "Datos incorrectos"

	<p>4. a El sistema verifica que los datos introducidos no son correctos.</p> <p>4. b El sistema muestra símbolos de error al lado de los campos en los cuales se introdujeron datos no correctos.</p>
Sección “Modificar costo”	
	1- El sistema le muestra en la interfaz los datos del costo del proyecto.
2- Selecciona el costo actual y lo modifica.	3- El sistema almacena los cambios y muestra el dato del costo actual del proyecto.
Sección “Mostrar costo”	
1- El actor selecciona en el menú Proyecto la opción “Información del Proyecto”.	1- El sistema muestra los datos de la interfaz Información del Proyecto, visualizando el costo del proyecto.
Poscondiciones	<p>Se adiciona el costo a un proyecto.</p> <p>Se modifica el costo a un proyecto.</p> <p>Se muestra el costo a un proyecto.</p>

Tabla 3 Descripción del Caso de Uso Gestionar costo a un proyecto

Diagrama de Casos de Uso del paquete Configuración.

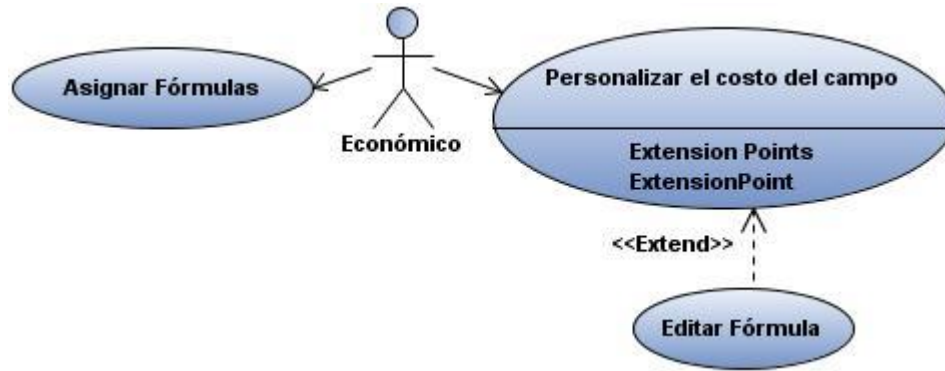


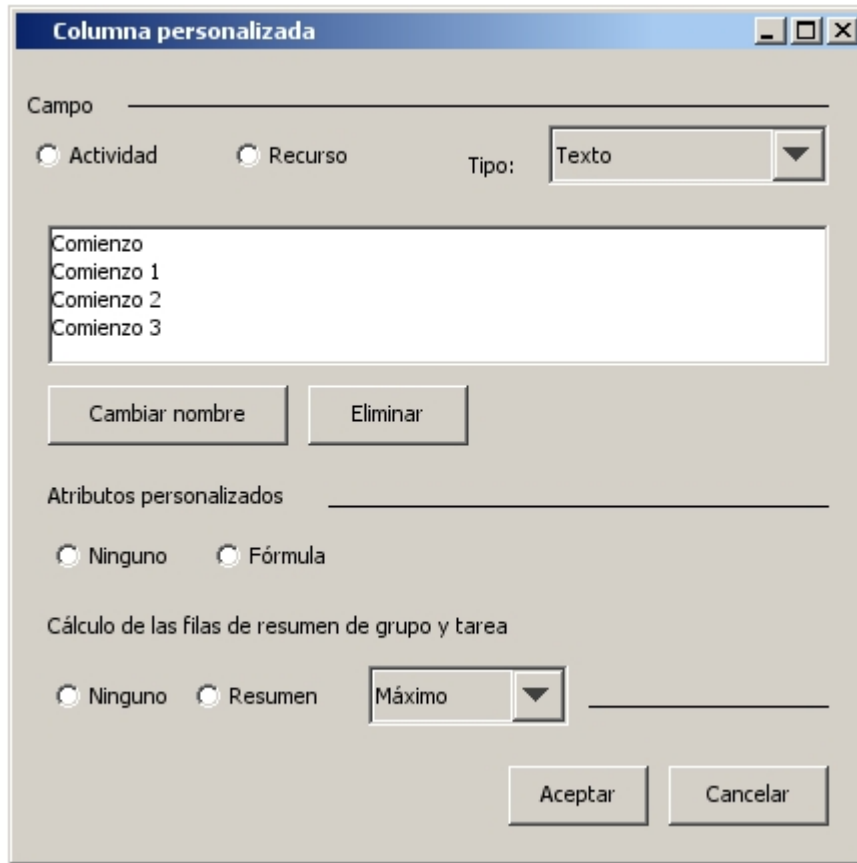
Figura 10 Diagrama de Casos de Uso del paquete Configuración

El paquete Configuración posee dos CU que obtienen clasificación de críticos, a continuación se presenta uno de ellos.

Descripción del Caso de Uso Personalizar el costo del campo

Caso de Uso:	Personalizar el costo del campo
Actores:	Económico
Resumen:	El caso de uso se inicia cuando el actor solicita “Personalizar el costo del campo”. El caso de uso finaliza una vez que se personalice el costo del campo.
Precondiciones:	El sistema debe estar ejecutándose correctamente. Debe existir al menos un proyecto. Deben existir al menos una actividad.
Referencias	RF33
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El actor selecciona en el menú Herramientas la opción Personalizar.	2- El sistema muestra un menú con las opciones: -Barra de herramientas -Campos

	-Formas
3-El actor selecciona "Campos"	4- El sistema muestra una interfaz con los siguientes campos: -Campo (actividad, recurso) -Tipo (fecha, fin, costo, otros) -Atributos personalizados (ninguno, fórmulas) -Cálculos de las filas de resumen de grupo y tarea (ninguno, resumen, otros)
5- El actor suministra los datos. 6- El actor selecciona "Aceptar".	7- El sistema verifica que no falten los datos obligatorios. 8- El sistema valida los datos entrados. 9- El sistema persiste la información. 10- El caso de uso finaliza.
Prototipo de Interfaz	



Flujo Alternativo al paso 6 "Operación Cancelada"	
6. a El actor cancela el flujo antes de culminar el caso de uso.	6. b Se retorna a la pantalla anterior sin guardar ningún cambio.
Flujo Alternativo al paso 7 "Datos obligatorios"	
	7. a El sistema verifica que faltan los datos en los campos obligatorios. 7. b El sistema muestra símbolos al lado de los campos que son obligatorios.
Flujo Alternativo al paso 8 "Datos incorrectos"	
	8. a El sistema verifica que los datos introducidos no son correctos. 8. b El sistema muestra símbolos de error al lado de los campos en los cuales se

	introdujeron datos no correctos.
Poscondiciones	Se guarda la personalización del costo del campo.

Tabla 4 Descripción del Caso de Uso Crear Personalizar el costo del campo

2.6 Conclusiones

Se abordaron las características principales del análisis del Módulo de Gestión de Costos en la descripción de la propuesta de solución. Además, a partir del análisis se obtuvo la realización del modelo de dominio, se especificaron los requisitos del software tanto funcionales como no funcionales y se definió los casos de uso del sistema, los cuales fueron representados en el diagrama de paquetes. También se realizó la descripción de los casos de uso del sistema por los paquetes obtenidos.

Análisis de los Resultados **3**

En este capítulo se realizará el análisis de los resultados alcanzados en el capítulo anterior, con el fin de evaluar los artefactos generados. Se utilizarán para la validación técnicas y algunas métricas de la calidad del software. Además, se plantean los resultados obtenidos y se analizan teniendo en cuenta las diferentes métricas o técnicas aplicadas a cada artefacto generado, con el objetivo de verificar la calidad con que fueron realizados.

3.1 Validación de los requisitos

Los requerimientos una vez definidos necesitan ser validados. La validación de requerimientos es muy importante, pues un levantamiento de requerimientos con errores que no se detecten a tiempo conduce a resultados inesperados provocando costos excesivos y gran pérdida de tiempo (Ingeniería, 2010).

Existen algunas propuestas de técnicas para la realización de la validación, la mayoría se destinan a revisar los modelos obtenidos en la definición de requerimientos. A continuación se muestran las que se usarán para verificar la calidad de los artefactos del análisis.

- **Revisiones (Reviews o Walk-throughs):** esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requerimientos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.
- **Listas de chequeo:** son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados.
- **Auditorías:** revisión de la documentación, esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.

- **Matrices de trazabilidad:** esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (Durán A., 1999). Es necesario ver qué objetivos cubre cada requerimiento, de tal forma se puede detectar inconsistencias u objetivos no cubiertos.
- **Prototipos:** algunas propuestas se basan en obtener de la definición de requerimientos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al cliente hacerse una idea de la estructura de la interfaz del sistema con el usuario (Ingeniería, 2010).
Puede revelar errores u omisiones en los requerimientos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto. La realización de prototipos es una de las propuestas más usadas en la validación de los requerimientos actualmente, por sus beneficios que reportan a los usuarios y desarrolladores.
- **Casos de Prueba:** un caso de prueba se deriva de un caso de uso en el modelo del sistema. Con estos casos de prueba se validan los requerimientos funcionales del sistema. Se debería desarrollar un caso de prueba para cada escenario del caso de uso. Los escenarios de un caso de uso se identifican describiendo los distintos caminos del flujo básico y flujo alternativo del caso de uso. Incluye la verificación del resultado de la interacción entre los actores y el sistema, que satisfacen las precondiciones y poscondiciones especificadas.

A continuación se presenta una combinación de las diferentes técnicas aquí explicadas, aplicando las mismas a los artefactos del análisis para validar la calidad de los resultados obtenidos.

3.2 Estudio de la validación por fases

La validación se refiere a un conjunto de diferentes actividades, que aseguran que el software construido se ajusta a los requisitos del cliente. El cual, en la Ingeniería de Requisitos permite demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente (Pressman, 2002).

Para garantizar la calidad de los artefactos del análisis del módulo de Gestión de Costos, se aplicará el procedimiento para la validación de requisitos de software (PVRS), realizado en un trabajo de diploma que tiene como título: “Procedimiento para la validación de requisitos de software”.

Éste tiene como propósito aplicarse a todos los proyectos de la Universidad de las Ciencias Informáticas, que contribuirá a mejorar la calidad de los requerimientos que son entregados al cliente (Padrón Blanco, 2009).

Se pretende validar los documentos que aportan gran flujo de información referente a los requisitos de software, los cuales son:

- Documento de especificación de requisitos de software.
- Modelo de casos de uso del sistema.

El PVRS presentan un conjunto de actividades que servirán de guía para el proceso de validación, este se ha dividido en tres fases para mejor organización:

I: Validar los requisitos de software.

II: Validar el documento de especificación de requisitos de software.

III: Validar el modelo de casos de uso del sistema.

A continuación se explica cada una de las actividades realizadas en cada fase del procedimiento, la cuales se realizaron en 3 revisiones dando a conocer los resultados obtenidos de cada validación.

3.2.1 Fase I. Validación del listado de requisitos de software

En esta fase se validó el listado de requerimientos definidos en el sistema, se aplicó la técnica revisiones para corregir la información del listado de los mismos. Se utilizó para revisar que los requerimientos se encuentren contenidos en un documento tangible tanto para los clientes y desarrolladores. Además de verificar que los requerimientos no sean ambiguos, siendo estos posibles de probar, así como que describa de forma concisa qué es lo que debe hacer el sistema. También su descripción debe ser lo más detallado posible, siendo conciso en su información.

Para esta primera fase se aplicó además, las listas de chequeo del Centro de Calidad de Software (Calisoft) como:

- Criterios para validar requisitos del cliente.
- Criterios para validar requisitos del producto.
- Lista de Verificación de Criterios para validar requisitos del producto.

Con estas listas se aplicarán un conjunto de preguntas a los requerimientos, que permitirá verificar la ambigüedad, factibilidad y si estos pueden ser probados. Además de identificar los elementos de entrada y salida y evaluar si su impacto es positivo. En esta fase se desarrollaron 3 revisiones para validar el listado de requerimientos, a continuación se presentan los resultados de cada una de ellas.

Revisión 1:

En esta primera revisión se detectó que 9 de los requerimientos eran ambiguos, y 5 no eran posibles de probar. Además, faltaban 6 requerimientos de reportes financieros de la Gestión de Costos que no

estaban contenidos en el documento. Se identificó que 2 de los requerimientos no describían las necesidades que debía cumplir el software. También se observó que según las nuevas plantillas de Calisoft les faltaban las entradas y salidas de cada requerimiento.

Revisión 2:

Para esta revisión se corrigieron los problemas encontrados anteriormente, pero aún con los arreglos previamente hechos se encontró errores en el listado de los requerimientos. Se observaron 2 requerimientos redundantes que fueron redactados en uno solo, además de los 6 requisitos descritos de reportes financieros se encontró que 2 eran tan complejos que se separaron en 4 requerimientos. Además, se encontraron 3 requerimientos que eran ambiguos. A pesar de haber realizado los arreglos anteriores, se detectó que las entradas y salidas de cada requerimiento no estaban bien especificadas, y no describían todas las entradas que en realidad contenían cada uno.

Revisión 3:

Al realizar la tercera revisión se habían solucionado los errores que se detectaron en las anteriores. Se observó que se eliminaron las ambigüedades e inconsistencias, además de las redundancias. Los requerimientos eran posibles de verificar y no faltaba ninguno principal; aunque se encontró 2 requerimientos que les hacía falta abundar en su especificación para dejar claro conceptos que podrían confundir entre costos y gastos.

Como resultado de las 3 revisiones, se puede afirmar que los requerimientos obtenidos en el análisis realizado y según los resultados obtenidos en esta etapa de definición de los mismos son correctos. Se puede asegurar que en cada revisión se fue perfeccionando cada requerimiento, y se obtuvo un listado con un 97 por ciento de la calidad esperada, que se puede evidenciar en la siguiente figura (Ver **Figura 11**).

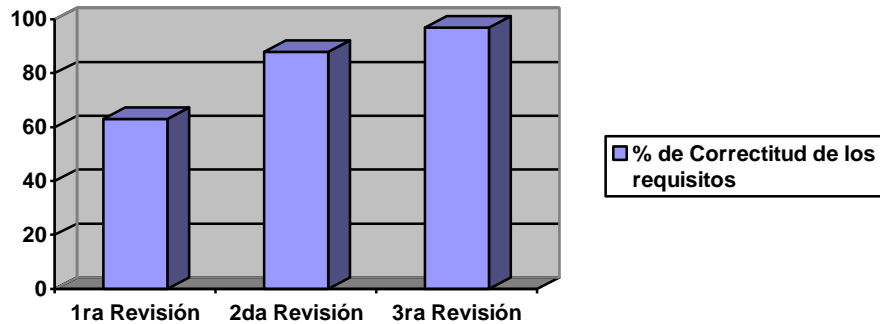


Figura 11 Resultados de las Revisiones de la Fase I.

En la figura, cada barra representa una revisión contra el total de corrección de la especificación de los requisitos. Se observa el avance de los arreglos de cada error encontrado y el desarrollo más exhaustivo de la completitud de los requerimientos.

3.2.2 Fase II. Validación del documento de especificación de requisitos

Para esta segunda fase se validó el documento de especificación de requisitos de software (ERS), para lograr una mejor calidad en del mismo. Se aplicó la técnica de auditoría con la lista de chequeo especificación de requisitos como se muestra en la siguiente tabla (Ver **Tabla 5**), para verificar una parte del documento.

Estructura del documento			
Indicadores a Evaluar	Eval	Cantidad de elementos afectados	Comentarios
1. ¿Está el documento acorde con la plantilla estándar del proyecto o del expediente de proyecto?	B	Ninguno	
2. ¿Contiene las secciones obligatorias definidas en el	B	Ninguno	

expediente? (Ver Expediente de Proyecto)			
Elementos definidos por la metodología			
Indicadores a Evaluar	Eval	Cantidad de elementos afectados	Comentarios
1. ¿Debería especificarse algún requisito con más detalle?	B	Ninguno	
2. ¿Han sido abordadas e identificadas los valores de entradas y salidas?	B	Ninguno	
3. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?	B	Ninguno	
4. ¿No aparece un mismo requisito en más de un lugar del documento de especificación?	B	Ninguno	
5. ¿Existe correspondencia entre el modelo de caso de uso, las Especificaciones Suplementarias y las especificaciones de requerimientos?	B	Ninguno	
6. ¿Se han identificado los requerimientos de software y de hardware?	B	Ninguno	
7. ¿Se puede verificar cada requisito? (Un requisito se dice que es verificable si	B	Ninguno	

existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento, ejemplo la especificación del caso de uso).			
8. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?	B	Ninguno	
Semántica del documento			
Indicadores a Evaluar	Eval	Cantidad de elementos afectados	Comentarios
1. ¿Ha identificado errores ortográficos?	B	Ninguno	
2. ¿Se entiende claramente lo que se ha especificado en el documento?	R	2	Hay especificaciones que tienen varias interpretaciones
3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?	B	Ninguno	
4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?	B	Ninguno	

Tabla 5 Ejemplo de elementos definidos en la lista de chequeo especificación de requisitos.

Como resultado de esta lista de chequeo se detectó que hay 2 especificaciones que tienen varias interpretaciones. Después de aplicar al documento de ERS la lista de chequeo especificación de requisitos, se procede al refinamiento del mismo para que todos los elementos tengan una evaluación de B y así garantizar que no presente ningún error.

Para que el documento de ERS cumpliera con las características solicitadas, se hizo necesario iterar 2 veces el procedimiento. En la primera iteración se detectó el problema explicado anteriormente que fue solucionado, dando paso a la próxima iteración. En esta última iteración se evidenció el avance de corrección del documento en cuanto al error anterior, quedando con evaluación de B todos los elementos de la lista y garantizando la calidad del documento.

En el documento ERS también se aplicó la métrica para la Calidad de la Especificación de los Requisitos de Software, realizado con cara al cliente, donde ellos expresan su aceptación por los requisitos existentes, a continuación se explica el procedimiento de la métrica.

Métrica para la Calidad de la Especificación de los Requisitos de Software

Existe una lista de características para poder valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: especificidad, corrección, complejidad, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. Aunque muchas de las características anteriores pueden ser de naturaleza cuantitativa (Pressman, 2002).

Para medir las características de la especificación, es necesario profundizar cuantitativamente en la especificidad y en la completitud.

La métrica es el resultado de dividir el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas (n_{ui}) entre la cantidad de requisitos de software (n_r).

$$Q_1 = n_{ui} / n_r$$

$$n_r = n_f + n_{nf}$$

Donde:

n_f : cantidad de requisitos funcionales.

n_{nf} : cantidad de requisitos no funcionales.

Cuanto más cerca de 1 esté el valor de Q_1 menor será la ambigüedad de la especificación.

Se realizaron dos revisiones para llegar a obtener un buen análisis, sin ambigüedad en los requisitos, en las cuales se obtuvieron los siguientes resultados:

Revisión 1:

En esta primera revisión, los revisores tuvieron una interpretación bastante parecida de los requerimientos. A continuación se aplica la ecuación anterior.

$$n_f = 41, n_{nf} = 19$$

$$nr = 41 + 19, nr = 60$$

$$Q1 = 59/60$$

$$Q1 = 0,983$$

Revision 2:

En los requisitos presentados en el documento ERS, todos los revisores tuvieron la misma interpretación, quedando de la siguiente manera la ecuación planteada anteriormente:

$$nf = 41, nnf = 19$$

$$nr = 41 + 19, nr = 60$$

$$Q1 = 60/60$$

$$Q1 = 1$$

Después de aplicar la ecuación se observó que los requisitos en la primera revisión tenían un nivel bajo de ambigüedad, porque el valor de Q se aproxima bastante a 1. Mientras que en la segunda revisión no presentan ningún nivel de ambigüedad, porque el valor de Q es 1. Con este resultado se puede decir que se tiene un documento de especificación de requisitos que presenta las características exigidas en su validación.

3.2.3 Fase III. Validación del modelo de Casos de Uso del Sistema

En la última fase se validó el documento de casos de uso del sistema para corregir errores que pueda tener. Se aplicó la técnica de matriz de trazabilidad, prototipos de interfaz y métricas de CU con el objetivo de verificar la calidad requerida en el proceso de análisis realizado. Se verificó que cada requisito se vean reflejado en al menos un caso de uso, además de observar la completitud, consistencia y complejidad de los CU.

Aplicando la técnica matriz de trazabilidad se verificó cada requerimiento contra los casos de uso existentes, para lograr que todos los requisitos se encontraran en al menos un CU. En esta técnica se realizó 2 iteraciones, obteniendo en la primera que 3 requerimientos no se reflejaban en ningún caso de uso, ya en la segunda iteración fueron otros los resultados. Se detectó que todos los requerimientos identificados responden al menos a un CU del sistema y viceversa. A continuación se presenta la matriz resultante del análisis realizado (Ver **Tabla 6**):

Requisitos Funcionales	Casos de Uso												
	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10	CU11	CU12	CU13
RF1	X												
RF2		X											
RF3		X											
RF4		X											
RF5									X				
RF6									X				
RF7									X				
RF8									X				
RF9									X				
RF10									X				
RF11			X										
RF12			X										
RF13			X										
RF14			X										
RF15			X										
RF16			X										
RF17				X									
RF18				X									
RF19				X									
RF20				X									
RF21				X									
RF22				X									
RF23				X									
RF24				X									
RF25					X								
RF26					X								
RF27					X								
RF28						X							
RF29						X							
RF30						X							
RF31									X				
RF32							X						
RF33								X					
RF34									X				
RF35									X				
RF36									X				
RF37									X				
RF38										X			
RF39											X		
RF40												X	
RF41													X

Tabla 6 Matriz de trazabilidad.

Una vez realizada la técnica anterior se procedió a la aplicación de la métrica de CU. Éste consta de cuatro atributos que se tuvieron en cuenta, los cuales son: Consistencia, Correctitud, Completitud y Complejidad.

Donde:

- Completitud: grado en que se han detallado los casos de uso más relevantes.
- Correctitud: grado en que las interacciones entre actor y sistema soportan el modelo de negocio.
- Complejidad: grado de presentación de los elementos que describe el contexto y la claridad del sistema.
- Consistencia: grado en que los casos de uso del sistema describen las interacciones entre el usuario y el sistema.

A continuación se muestra una tabla con los resultados de las métricas que deben cumplir los requisitos del software (Ver **Tabla 7**).

Factor Completitud	Métrica Asociada	Sistema
Factor 7. ¿Están definidos todos los requerimientos que justifican la funcionalidad del Caso de Uso?	Métrica 7: Número de requerimientos omitidos por Caso de Uso. Umbral < 10%	Total de casos de uso: 13 Número de Caso de Uso con requerimientos sin definir: 0 Representa: 0%
Factor 8. ¿Existen requerimientos que no han sido considerados en algún Caso de Uso?	Métrica 9: Número de requerimientos que no son considerados en ningún Caso de Uso.	Total de RF: 42 Total de RF sin incluir en algún Casos de Uso: 0 Representa: 0%
Factor 13. ¿Todos los Casos	Métrica 15: Número de Casos	Total de Casos de Uso: 13

de Uso han sido clasificados de acuerdo con su relevancia?	de Uso que no han sido clasificados Umbral: < 10%	Número de Casos de Uso sin clasificar: 0 Representa: 0%
Factor Consistencia	Métricas Asociadas	Sistema
Factor 14. ¿El nombre dado a los Casos de Uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	Métrica 16: Número de Casos de Uso que tienen un nombre incorrecto Umbral < 20%	Total de Casos de Uso: 13 Número de Casos de Uso que tienen un nombre incorrecto: 0 Representa: 0%
Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 25: Número de Casos de Uso complejos que no tienen separación del flujo básico y de flujos alternos. Umbral: < 20%	Total de Casos de Uso: 13 Número de Casos de Uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Representa: 0%
Factor Correctitud	Métrica Asociada	Sistema
Factor 23: ¿Representa el Caso de Uso requerimientos comprensibles por el usuario?	Métrica 28: Número de Casos de Uso en que los requerimientos representados no son comprensibles por el	Total de Casos de Uso: 13 Número de Casos de Uso en que los requerimientos

	usuario. Umbral: < 5%	representados no son comprensibles por el usuario: 0 Representa: 0%
Factor Complejidad	Métrica Asociada	Sistema
Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	36. Número de elementos del diagrama que requieren reubicación. Umbral: < 30%	Total de Casos de Uso: 13 Número de elementos del diagrama que requieren reubicación: 0 Representa: 0%

Tabla 7 Aplicación de métricas al diagrama de casos de uso del sistema.

Para evidenciar los valores numéricos de la medición de los factores de calidad que han sido evaluados, se muestra el siguiente gráfico (Ver **Figura 12**).

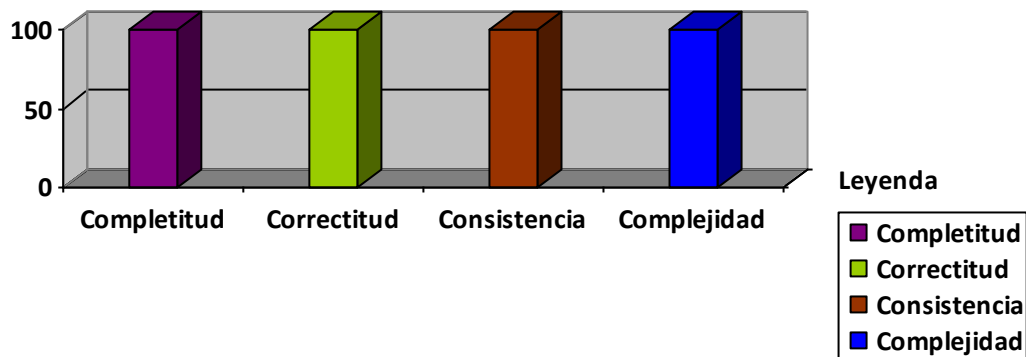


Figura 12 Representación de los valores aportados por cada factor de calidad aplicado.

Al aplicar estas métricas se evaluaron los factores completitud, consistencia, correctitud y complejidad de los CU del sistema. Se demostró que los casos de uso fueron descritos detalladamente, en ellos se evidencia el flujo normal y alterno que proporciona una mayor legibilidad. Todos los CU tenían un nombre correcto, además los requisitos son implementados en al menos en un caso de uso. Estos resultados permitió demostrar que el artefacto caso de uso del sistema posee la calidad deseada en la validación.

También se realizaron los prototipos de los CU, el cual facilitó visualizar mejor estas descripciones para que los especialistas tuvieran una idea de la estructura de la interfaz del sistema. Resultó una herramienta primordial para comprobar los requerimientos del sistema, siendo un modelo que integró las expectativas y necesidades de los clientes. Con los prototipos se validó los requisitos funcionales que fueron capturados durante la etapa de Requerimientos, además de representar en un entorno fácil las necesidades de los usuarios.

Aparte de las técnicas realizadas para validar que los casos de uso engloban correctamente los requisitos, existen los casos de pruebas, que son una forma más de verificar que los artefactos obtenidos en el análisis son correctos. A continuación se presentan las pruebas elaboradas.

3.3 Pruebas

La etapa de pruebas es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la proyección del sistema. Esta consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema y tiene como objetivo descubrir errores.

3.3.1 Pruebas de sistemas

Se realizaron un caso de prueba (CP) por cada CU, en vista a obtener un nivel correcto del análisis efectuado a cada caso de uso descrito en el capítulo anterior. Ésta técnica permitió controlar el desempeño de las descripciones realizadas a los CU, comprobándose que poseen la calidad pretendida.

A continuación se presenta algunos de los 13 CP realizados que cuentan con 41 secciones, los otros se encuentran en el artefacto Diseño de Casos de Prueba que se encuentran en el expediente de proyectos (Rondón, 2010).

Casos de Prueba para el Caso de Uso Gestionar costo a un proyecto

ID del escenario	Escenario	Nombre del proyecto	Fecha inicio	Fecha Fin	Estado del proyecto	Tipo de proyecto	Enunciado del proyecto	Costo actual	Respuesta del sistema	Resultado de la prueba
EC 1.1: Adicionar costo a un proyecto	Adicionar costo a un proyecto	V	V	V	V	V	V	V	El sistema verifica que los datos estén correctos y persiste la información.	
EC 1.2: Operación Cancelada	Operación Cancelada	V	V	V	V	V	V	V	El sistema cancela la operación y se retorna a la pantalla anterior sin guardar ningún cambio.	
EC 1.3: Datos obligatorios	Datos obligatorios	I	V	V	V	V	V	V	El sistema visualiza símbolos al lado de los campos que son obligatorios.	
		V	I	V	V	V	V	V		
		V	V	I	V	V	V	V		
		V	V	V	I	V	V	V		
		V	V	V	V	I	V	V		
		V	V	V	V	V	I	V		
		V	V	V	V	V	V	I		

EC 1.4: Datos incorrectos	Datos incorrectos	I	V	V	V	V	V	V	El sistema muestra símbolos de error al lado de los campos en los cuales se introdujeron datos incorrectos	
		V	I	V	V	V	V	V		
		V	V	I	V	V	V	V		
		V	V	V	I	V	V	V		
		V	V	V	V	I	V	V		
		V	V	V	V	V	I	V		
		V	V	V	V	V	V	I		

Tabla 8 Caso de Prueba de la SC 1 Adicionar costo a un proyecto

ID del escenario	Escenario	Nombre del proyecto	Fecha inicio	Fecha Fin	Estado del proyecto	Tipo de proyecto	Enunciado del proyecto	Costo actual	Respuesta del sistema	Resultado de la prueba
EC 2.1: Modificar costo a un proyecto	Modificar costo a un proyecto	V	V	V	V	V	V	V	El sistema modifica el costo actual y guarda la información.	

Tabla 9 Caso de Prueba de la SC 2 Modificar costo a un proyecto

ID del escenario	Escenario	Nombre del proyecto	Fecha inicio	Fecha Fin	Estado del proyecto	Tipo de proyecto	Enunciado del proyecto	Costo actual	Respuesta del sistema	Resultado de la prueba
EC 3: Mostrar costo a un	Mostrar costo a un	V	V	V	V	V	V	V	El sistema visualiza el costo actual del	

proyecto	proyecto								proyecto.	
----------	----------	--	--	--	--	--	--	--	-----------	--

Tabla 10 Caso de Prueba de la SC 3 Mostrar costo a un proyecto

3.4 Conclusiones

En este capítulo se realizó el proceso de análisis de los resultados, se validaron los requerimientos funcionales y los casos de uso mediante la aplicación de técnicas como: matriz de trazabilidad, casos de pruebas, listas de chequeo, revisiones, auditorías y prototipos; así como métricas para evaluar la calidad. Con estas técnicas y métricas se logró validar todos los artefactos obtenidos en el análisis, demostrando que tienen la calidad requerida.

Conclusiones Generales

Con la culminación del presente trabajo se puede constatar que:

- Con el estudio detallado del estado del arte de la Gestión de Costos de Portafolios de proyectos se identificaron las entidades del dominio relevantes para el desarrollo de la solución.
- La aplicación sistemática de las técnicas de Ingeniería de Requisitos particularmente la Introspección de las herramientas existentes, permitió la elaboración de los artefactos Especificación de Requisitos y Modelo del Sistema para las posteriores actividades de Diseño e Implementación.
- Se le aplicó a los artefactos generados las métricas y técnicas definidas en el estudio, garantizando la calidad de los resultados.
- Todo lo anterior nos permite reafirmar que el análisis detallado del Módulo de Gestión de Costos obtenido constituye la base para el diseño e implementación del sistema, que redundará en una reducción de costos y en una mejora en la toma de decisiones sobre las carteras de proyectos.

Recomendaciones

Se proponen las siguientes recomendaciones:

1. Realizar el diseño del Módulo de Gestión de Costos para un sistema integral de Gestión de Portafolios de proyectos.
2. Implementar la Plataforma de Gestión de Portafolios de proyectos, en la cual se integre el Módulo de Gestión de Costos.
3. Unir el sistema informático de Gestión de Portafolios de proyectos con el ERP.
4. Aplicar el sistema integral al Centro DATEC y a la empresa ALBET para verificar sus resultados.
5. Extender la herramienta en otras organizaciones y empresas de Cuba, además de comercializarla a otros países.

Referencias Bibliográficas

1. **Braude, Eric J. 2003.** *Ingeniería de Software: Una perspectiva orientada a objetos.* s.l. : Editorial Ra-Ma, 2003.
2. **Durán A., Bernárdez, B., Ruiz, A., Toro M. 1999.** *A Requirements Elicitation Approach Based in Templates and Patterns. Workshop de Engenharia de Requisitos.* . Buenos Aires,Argentina : s.n., 1999.
3. **D-PAQ.** Diagrama de paquetes según Métricas V3. [En línea] http://www.di.uniovi.es/~dediego/is/recursos/d_paq.pdf.
4. **Facultad de Ingeniería UBA. 2008.** INGENIERIA DE SOFTWARE EDUCATIVO. [En línea] 2008. <http://www.itba.edu.ar/archivos/secciones/c19-icie99-ingenieriasoftwareeducativo.pdf>.
5. **García, Lourdes y Fernández, Leidy. 2009.** Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software(PROCIR) . Instituto Tecnológico de Morelia : Departamento de Sistemas y Computacion, 2009.
6. **IBM. 2010.** [En línea] 2010. <http://www.ibm.com/software/awdtools/focalpoint/features/>.
7. —. **2010.** [En línea] 2010. <http://www-01.ibm.com/software/awdtools/focalpoint/>.
8. **IEEE. 2010.** [En línea] IEEE Standard Glossary of Software Engineering Terminology, 2010. <http://standards.ieee.org>.
9. **Ingeniería, Software 1. 2010.** *Fase de Inicio.Flujo de trabajo de requerimientos.* 2007. Conferencia 6.
10. —. **2009.** *Profundización del flujo de trabajo de requerimientos.* 2009.
11. **IPMA. 2006.** *International Project Management Association (IPMA).* 2006.
12. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James. 2000.** El Proceso Unificado de Desarrollo de Software. *Capítulos 6, 7, 8.* s.l. : Addison Wesley, 2000.
13. **Leyva, Olga Monzón y ALBET. 2009.** ALBET. *Gestión Económica Contable y Financiera II. Gestión de Costo.* 2009.
14. **Lowe, D., Hall, W. 1999.** *Hypermedia and the Web. An Engineering approach.* s.l. : John Wiley & Son, 1999.
15. **Lussón, Quevedo. 2009.** CENTALAD. [En línea] 2009. [Citado el: 30 de 1 de 2010.] <http://portal.centalad.prod.uci.cu>.
16. **Mc.Graw-Hill Companies. 2007.** *Contabilidad de Costos. Tomo 1.* s.l. : Félix Varela, 2007.

17. **Microsoft. 2010.** Microsoft Office Online. [En línea] 2010. <http://office.mocrosoft.com/es-es/epmsolution/HA101656443082.aspx>.
18. —. **2010.** [En línea] 2010. <http://www.microsoft.com/project/en/us/trainlearn>.
19. —. **2010.** Microsoft Office Online. [En línea] 2010. <http://office.mocrosoft.com/es-es/epmsolution/HA101656443082.aspx>.
20. **Oracle. 2010.** ORACLE. [En línea] 2010. <http://www.oracle.com/global/es/products/applications/project-management.html>.
21. **ORALLO.** http://eva.uci.cu/file.php/102/Curso_2009-2010/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf. [En línea]
22. **Padrón Blanco, Geidys y Rodríguez Iglesias, Ailec. 2009.** Procedimiento para la validación de requisitos de software. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas*. 2009.
23. **Paradigm, Visual. 2010.** *Paradigm Visual para UML*. [En línea] 2010. [Citado el: 4 de 2 de 2010.] http://freedownloadmanager.org/es/downloads/Paradigm_Visual_para_UML_Bcuenta_de_Plataforma_de_Java_14715_p/.
24. **Pérez. 1999.** *Arquitectura para Ambientes CASE Integrados*. Universidad de las Villas : s.n., 1999.
25. **PMI. 2004.** *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®)*. EE.UU : Tercera Edición, 2004.
26. —. **2006.** *Project Management Institute(PMI)*. USA : s.n., 2006. PA 19073-3299.
27. **Pressman. 2002.** *“Ingeniería del Software. Un enfoque práctico”*. 2002.
28. **Rational. 2010.** [En línea] 2010. <http://www-01.ibm.com/software/rational/products/rpc/index.html>.
29. **ROI. 2009.** CN Crece Negocios.com. *Retorno sobre la inversión*. [En línea] 30 de 9 de 2009. [http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.crecenegocios.com%2Fretorno-sobre-la-sobre-inversion-roi%2F&rct=j&q=Retorno+sobre+la+sobre+inversi%C3%B3n+\(ROI\)&ei=P9CqS7nFI8P_lgeJh4C4BA&usg=AFQjCNHRvQlcQ3baG3sBz](http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.crecenegocios.com%2Fretorno-sobre-la-sobre-inversion-roi%2F&rct=j&q=Retorno+sobre+la+sobre+inversi%C3%B3n+(ROI)&ei=P9CqS7nFI8P_lgeJh4C4BA&usg=AFQjCNHRvQlcQ3baG3sBz).
30. **Rondón, Kenia. 2010.** *Artefacto: "Diseño de Casos de Prueba"*. 2010.
31. —. **2010.** *Artefacto: "Modelo del Sistema"*. 2010.
32. **Saborío, Ronald y Quesada, Omar. 2006.** *Gestión de Portafolio*. 2006.

33. **Solenzal, Guillermo y Díaz, Sergio. 2006.** MULTIMEDIA AUTO-APRENDE. [En línea] 2006. <http://www.ilustrados.com/documentos/tesis-multimedia-auto-aprende-130907.pdf>.
34. **Synergix. 2008.** [En línea] 2008. <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
35. **Telemática. 2009.** *Publicación quincenal editada por el Departamento de Telecomunicaciones y Telemática del Instituto Superior Politécnico José Antonio Echevarría Auspiciada por el Ministerio de la Informática y las Comunicaciones.* [En línea] 30 de 1 de 2009. http://www.fiec.espol.edu.ec/resources/download/revista/Telematica_A%C3%B1oVII_No21.pdf.
36. **VIGO, Universidad de. 2010.** Captura de Requisitos. [En línea] 2010. http://jcgm.ei.uvigo.es/jcmoreno/?page_id=2.

Bibliografía

1. **Andriole, Stephen J. 1996.** *Managing Systems Requirements: Methods, Tools, and Cases.* s.l. : McGraw Hill, 1996.
2. **Braude, Eric J. 2003.** *Ingeniería de Software: Una perspectiva orientada a objetos.* s.l. : Editorial Ra-Ma, 2003.
3. **Briand L.C., Daly J.W. & Wüst J. 1996.** "A Unified Framework for Cohesion Measurement in Object-Oriented Systems". *Empirical Software Engineering*, 3, 65-117 1998. 1996.
4. **Cockburn, Alistair. 2001.** *Writing Effective Use Cases.* s.l. : Addison Wesley Longman, 2001.
5. **Cooper, Alan. 1999.** *The Inmates are Running the Asylum.* Indianapolis : SAMS, 1999.
6. **Davis, Alan. 1993.** *Software Requirements-Objects, Functions and States.* Englewood Cliffs : Prentice Hall, 1993.
7. **Diccionario. 2010.** Diccionario Informático. [En línea] 2010. <http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=C&pagina=3>.
8. **Dorfman, Richard H. Thayer and Merlin. 1997.** *Software Requirements Engineering.* IEEE Computer Society Press : 2nd Edition, 1997.
9. **Durán A., Bernárdez, B., Ruiz, A., Toro M. 1999.** *A Requirements Elicitation Approach Based in Templates and Patterns. Workshop de Engenharia de Requisitos.* . Buenos Aires, Argentina : s.n., 1999.
10. **D-PAQ.** Diagrama de paquetes según Métricas V3. [En línea] http://www.di.uniovi.es/~dediego/is/recursos/d_paq.pdf.
11. **Facultad de Ingeniería UBA. 2008.** INGENIERIA DE SOFTWARE EDUCATIVO. [En línea] 2008. <http://www.itba.edu.ar/archivos/secciones/c19-icie99-ingenieriasoftwareeducativo.pdf>.
12. **Fenton E., Pfleeger, S.L. 1997 .** "Software Metrics. A Rigorous and Practical Approach. Second edition". . s.l. : International Thompson Computer Press., 1997 .
13. **García, Lourdes y Fernández, Leidy. 2009.** Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software(PROCIR) . Instituto Tecnológico de Morelia : Departamento de Sistemas y Computacion, 2009.
14. **Gould, John D. 1988 .** "How to Design Usable Systems". in Helander, Martin, ed. Handbook of Computer Interaction, North-Holland, Amsterdam, The Netherlands : s.n., 1988 . 757-789.

15. **Grady, Robert. 1992.** *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall : s.n., 1992.
16. **Hall, Larman-Prentice.** UML y Patrones. *Introducción al análisis y diseño orientado a objetos*.
17. **Hall, Prentice. 2000.** *UML y Patrones*. LARMAN : Craig, 2000.
18. **Holtzblatt, Hugh Beyer and Karen. 1998.** *Contextual Design*. San Francisco : s.n., 1998. Morgan Kaufmann Publishers.
19. **Holtzblatt, K., and H. Beyer. 1996.** "Contextual Design: Principles and Practice," *Field Methods for Software and Systems Design*. Wixon and J. Ramey (Eds.), NY. : John Wiley & Sons, 1996.
20. **IBM. 2010.** [En línea] 2010. <http://www.ibm.com/software/awdtools/focalpoint/features/>.
21. —. **2010.** [En línea] 2010. <http://www-01.ibm.com/software/awdtools/focalpoint/>.
22. **IEEE. 2010.** [En línea] IEEE Standard Glossary of Software Engineering Terminology, 2010. <http://standards.ieee.org>.
23. **IEEE Std. 1993.** *Recommended Practice for Software Requirements Specifications*. Software Engineering Standards Committee of the IEEE Computer Society. New York : NY, 1993.
24. **Ingeniería, Software 1. 2010.** *Fase de Inicio. Flujo de trabajo de requerimientos*. 2007. Conferencia 6.
25. —. **2009.** *Profundización del flujo de trabajo de requerimientos*. 2009.
26. **IPMA. 2006.** *International Project Management Association (IPMA)*. 2006.
27. **ISO/TC159. 1999.** Human-centred design processes for interactive systems. [book auth.] Report ISO 13407:1999. *International Organization for Standardization*. Geneva, Switzerland : s.n., 1999.
28. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software. Capítulos 6, 7, 8*. s.l. : Addison Wesley, 2000.
29. **John D. Gould, Stephen J. Boies, Stephen Levy, John T. Richards and Jim Schoonard. 1987.** "The 1984 Olympic Message System: a test of behavioral principles of system design". *Communications of the ACM* : s.n., 1987, Vols. 30, nº 9. 758-769.
30. **2003.** *La Dirección Integrada de Proyectos haciendo uso de las Nuevas Tecnologías de la Informática y las Comunicaciones*. Cuba : CETA. ISPJAE, 2003.
31. **Kendall, Kendall &. 1997.** "Análisis y Diseño de Sistemas". 1997.
32. **Kovitz, Benjamin L. 1999.** *Practical Software Requirements-A Manual of Content & Style*. s.l. : Manning Publications, 1999.

33. **Larman, Craig.** *UML y Patrones. Capítulo 6 GRASP: Diseño de objetos con responsabilidades.*
34. **Leyva, Olga Monzón y ALBET. 2009.** ALBET. *Gestión Económica Contable y Financiera II. Gestión de Costo.* 2009.
35. **Lockwood, Larry Constantine and Lucy A.D. 1999.** *Software for Use.* . s.l. : MA: Addison Wesley Longman, 1999.
36. **Lowe, D., Hall, W. 1999.** *Hypermedia and the Web. An Engineering approach.* s.l. : John Wiley & Son, 1999.
37. **Lussón, Quevedo. 2009.** CENTALAD. [En línea] 2009. [Citado el: 30 de 1 de 2010.] <http://portal.centalad.prod.uci.cu>.
38. **Mayhew, Deborah J. 1999 .** *The Usability Engineering Lifecycle.* s.l. : Morgan Kaufmann Publishers, 1999 .
39. **Mc.Graw-Hill Companies. 2007.** *Contabilidad de Costos. Tomo 1.* s.l. : Félix Varela, 2007.
40. **Microsoft. 2010.** [En línea] 2010. <http://www.microsoft.com/latam/office/project/prodinfo/epm/overview.msp>.
41. —. **2010.** [En línea] 2010. <http://www.microsoft.com/project/en/us/trainlearn>.
42. —. **2010.** Microsoft Office Online. [En línea] 2010. <http://office.mocrosoft.com/es-es/epmsolution/HA101656443082.aspx>.
43. **Oracle. 2010.** ORACLE. [En línea] 2010. <http://www.oracle.com/global/es/products/applications/project-management.html>.
44. **ORALLO.** http://eva.uci.cu/file.php/102/Curso_2009-2010/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf. [En línea]
45. **OrBit. 2010.** The rAils OrBit. [En línea] 22 de 1 de 2010. <http://jonathanvalencia.wordpress.com/>.
46. **Padrón Blanco, Geidys y Rodríguez Iglesias, Ailec. 2009.** Procedimiento para la validación de requisitos de software. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.* 2009.
47. **Paradigm, Visual. 2010.** *Paradigm Visual para UML.* [En línea] 2010. [Citado el: 4 de 2 de 2010.] http://freedownloadmanager.org/es/downloads/Paradigm_Visual_para_UML_Bcuenta_de_Plataforma_de_Java_14715_p/.
48. **Pérez. 1999.** *Arquitectura para Ambientes CASE Integrados.* Universidad de las Villas : s.n., 1999.

49. **Peter Eeles. 2004.** *Capturing Architectural Requirements*. [Online] The Rational Edge, 2004. <http://www.ibm.com/developerworks/rational/library/4706.html>.
50. **PMI. 2004.** *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®)*. EE.UU : Tercera Edición, 2004.
51. —. **2006.** *Project Management Institute(PMI)*. USA : s.n., 2006. PA 19073-3299.
52. **Politécnico_Nacional. 2008.** Comisión de Operación y Fomento de Actividades Académicas. *Propuesta de "Lineamientos para Sistemas Informáticos"*. [En línea] Instituto Politécnico Nacional, 2008. <http://cofaa.ipn.mx/conocenos/normateca/lineamientos/documentos/especificaciones.pdf>.
53. **Pressman, Roger S. 2002. Páginas 184-186 .** "Ingeniería del Software. Un enfoque práctico". [aut. libro] McGraw-Hill. *Técnicas para facilitar las especificaciones de la aplicación*. Interamericana de España : s.n., 2002. Páginas 184-186 .
54. —. **2002.** "Ingeniería del Software. Un enfoque práctico" . 2002.
55. **Rational. 2010.** [En línea] 2010. <http://www-01.ibm.com/software/rational/products/rpc/index.html>.
56. **ROI. 2009.** CN Crece Negocios.com. *Retorno sobre la inversión*. [En línea] 30 de 9 de 2009. [http://www.google.com/cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.crecenegocios.com%2Fretorno-sobre-la-sobre-inversion-roi%2F&rct=j&q=Retorno+sobre+la+sobre+inversi%C3%B3n+\(ROI\)&ei=P9CqS7nFI8P_lgeJh4C4BA&usg=AFQjCNHRvQlcQ3baG3sBz](http://www.google.com/cu/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Fwww.crecenegocios.com%2Fretorno-sobre-la-sobre-inversion-roi%2F&rct=j&q=Retorno+sobre+la+sobre+inversi%C3%B3n+(ROI)&ei=P9CqS7nFI8P_lgeJh4C4BA&usg=AFQjCNHRvQlcQ3baG3sBz).
57. **Rondón, Kenia. 2010.** *Artefacto: "Diseño de Casos de Prueba"*. 2010.
58. —. **2010.** *Artefacto: "Modelo del Sistema"*. 2010.
59. **Ruth Malan and Dana Bredemeyer. 2001.** *Defining Non-Functional Requirements. Documento técnico*. [Online] 2001. www.bredemeyer.com.
60. **Saavedra, Jorge.** [En línea] [Citado el: 15 de 2 de 2010.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
61. **Saborío, Ronald y Quesada, Omar. 2006.** *Gestión de Portafolio*. 2006.
62. **Sawyer, Ian Sommerville and Pete. 1997.** *Requirements Engineering-A Good Practice Guide*. New York : John Wiley & Sons, 1997.
63. **Schach, Stephen R. 2006.** *Ingeniería de Software Clásica y Orientada a Objetos*. s.l. : McGraw-Hill, 2006.
64. **SlideShare. 2009.** [En línea] 2009. <http://www.slideshare.net/Caryl/05-casos-uso-bis-presentation>.

65. **Solenzal, Guillermo y Díaz, Sergio. 2006.** MULTIMEDIA AUTO-APRENDE. [En línea] 2006. <http://www.ilustrados.com/documentos/tesis-multimedia-auto-aprende-130907.pdf>.
66. **Spence, Kurt Bittner and Ian. 2003 .** *Use Case Modeling*. s.l. : Addison Wesley Longman, 2003 .
67. **Synergix. 2008.** [En línea] 2008. <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
68. **Telemática. 2009.** *Publicación quincenal editada por el Departamento de Telecomunicaciones y Telemática del Instituto Superior Politécnico José Antonio Echevarría Auspiciada por el Ministerio de la Informática y las Comunicaciones*. [En línea] 30 de 1 de 2009. http://www.fiec.espol.edu.ec/resources/download/revista/Telematica_A%C3%B1oVII_No21.pdf.
69. **Ury, Roger Fisher and William. 1991.** *Getting to Yes-Negotiating Agreement Without Giving In, 2nd Edition*. . s.l. : Penguin Books USA, 1991.
70. **VIGO, Universidad de. 2010.** Captura de Requisitos. [En línea] 2010. http://jcgm.ei.uvigo.es/jcmoreno/?page_id=2.
71. **Weinberg, Donald Gause and Gerald. 1989.** *Exploring Requirements-Quality Before Design*. New York : Dorset House, 1989.
72. **Weinberg, Gerald. 1995.** *"Just Say No! Improving the Requirements Process"*. s.l. : American Programmer, 1995.
73. **Wesly, Addison. 2004.** *Use Case: Patterns and Blueprints*. OVERGAARD, Gunnar : PALKVIST, 2004.
74. **Widrig, Dean Leffingwell and Don. 2000.** *Managing Software Requirements – a Unified Approach*. s.l. : Addison Wesley Longman, 2000.
75. —. **1999.** . *Effective Requirements Management*. s.l. : Addison Wesley Longman, 1999. .
76. **Winters, Geri Schneider and Jason P. 1998.** *Applying Use Cases-A Practical Guide*. . s.l. : Addison Wesley Longman, 1998.

Anexos

Anexo 1 Entrevista

Preguntas en la entrevista.	
No.	Preguntas
1	¿Cuál es la diferencia entre presupuesto, costo y gasto?
2	¿Qué asignan al proyecto, costo o presupuesto?
3	¿Cómo realizan la estimación de los proyectos?, ¿A qué estiman?
4	¿Qué tienen en cuenta para la estimación?, ¿Qué técnica utilizan? Ejemplo las contenidas en el PMBOK
5	¿Qué toman en cuenta para el presupuesto?, ¿Cómo lo usan, y a la hora de asignarlo qué tienen en cuenta?
6	¿Cómo guardan la línea base?
7	¿Cómo ven la variación de los costos?
8	¿Cómo controlan los costos?, ¿Qué técnica utilizan?
9	Explique si usan indicadores del rendimiento, ¿Cómo lo hacen?
10	¿Cómo realizan el análisis de la inversión?
11	¿Cómo y para qué realizan el Flujo de Caja?
12	¿Cómo ven la factibilidad?
13	¿Qué es para ustedes los Centros de costo?, ¿Cómo lo crean?

14	¿De qué manera ven el valor ganado y cómo lo visualizan en la curva "S"?
15	¿Qué tipo de reportes tienen y cómo lo llevan a cabo?
16	¿Cómo registran el rendimiento del proyecto, así como el cumplimiento?
17	¿Qué herramientas utilizan?

Tabla 11 Preguntas de la entrevista

Glosario de Término

Actividad: Es una tarea simple asociada a un proyecto.

Análisis: Método empleado para traducir los objetivos del proyecto en productos entregables y requisitos tangibles.

Campo: Definiciones que pertenecen a una categoría de actividad, estos tienen un nombre y un valor.

Categoría: Tipo de clasificación de las actividades que determinan su comportamiento, algunas clasificaciones son: hito, etapa, otras.

Centro de costo: Es la subdivisión mínima en el proceso del registro contable en la cual se acumulan los gastos de la actividad productiva de la empresa a los fines de facilitar la medición de los recursos utilizados y los resultados económicos obtenidos.

Control de costos: Influir sobre los factores que crean variaciones del costo y controlar los cambios en el presupuesto del proyecto.

Costos: Gasto económico que representa la fabricación de un producto o la prestación de un servicio.

DATEC: Centro de Gestión de Datos.

Estimación ascendente: Técnica que estimar el costo de paquetes de trabajo individuales o actividades del cronograma individuales con el nivel más bajo de detalle.

Estimación de costos: Proceso para desarrollar una aproximación de los costos de los recursos necesarios para completa la actividad del proyecto.

Estructura de Desglose del Trabajo (EDT): Es una descomposición jerárquica, orientada al producto entregable, del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requeridos.

Fórmula: Números y símbolos que muestran cómo obtener algo. Un tipo especial de ecuación que muestra la relación entre diferentes variables.

Gastos: Expresan, en términos monetarios, el monto total de los recursos tanto material, laboral como financieros, utilizados durante un período cualquiera, en el conjunto de la actividad empresarial.

Gestión de Costos: Describe los procesos involucrados en la planificación, preparación del presupuesto y control de costos de forma que el proyecto se pueda completar dentro el presupuesto aprobado.

Gestión de Portafolio: Es la identificación, priorización, autorización, administración y control de los proyectos, programas y otros trabajos conexos, para lograr los objetivos estratégicos empresariales.

Ingeniería de requisitos: Elicitación, análisis y negociación y especificación.

PMI (Project Management Institute): Constituyen la suma de conocimientos en la profesión de dirección de proyectos.

Portafolio: Colección de proyectos o programas y otros trabajos que se agrupan para facilitar la gestión eficaz de los que trabajan para alcanzar los objetivos estratégicos del negocio.

Presupuesto: Identifica los diferentes costos y el monto para la iniciación de cualquier proyecto empresarial.

Procedimiento: Es el modo de ejecutar determinadas acciones que suelen realizarse de la misma forma, con una serie común de pasos claramente definidos, que permiten realizar una ocupación o trabajo correctamente.

Proceso de Ingeniería de Requisitos: Conjunto de actividades que son seguidas con el objetivo de descubrir, modelar, validar y mantener un documento de requisitos. Este proceso debe lidiar con diferentes puntos de vista, usar una combinación de técnicas, herramientas y personas. Todo este proceso acontece en un universo de discurso con actores reales, por lo que se puede considerar un proceso centrado en las personas.

Producto: Artefactos que se crean durante el ciclo de vida de un proyecto.

Proyectos: Esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

Requisitos: Condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal.

Requisitos funcionales: Refieren a las funciones específicas del software y definen qué es lo que se espera que realice el producto software que se desarrollará.

Requisitos no funcionales: Propiedades o cualidades que el producto debe tener. Características que hacen el producto atractivo, usable, rápido o confiable.

Suma de costos: Suman las actividades del cronograma por paquetes de trabajo de acuerdo con la EDT.