

*Universidad de las Ciencias Informáticas
Facultad 8*

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

Análisis del Módulo de Gestión de Tiempo
del Sistema Integral para la Gestión de
Portafolios de Proyectos

Autor(es): Elizabeth Reina Meneses González

*Tutor(es): Ing. Jenny Infante Frometa
Ing. Adisleydis Rodríguez Pino*

Ciudad de La Habana, junio de 2010

Año 52 de la Revolución



Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del trabajo **“Análisis del Módulo de Gestión de Tiempo del Sistema Integral para la 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 _____.

Nombre y Apellidos del Autor

Nombre y Apellidos del Tutor

Nombre y Apellidos del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “**Análisis del Módulo de Gestión de Tiempo del Sistema Integral para la Gestión de Portafolios de Proyectos**”, fue realizado en <nombre de lugar>. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a <valor en MN o USD del efecto económico>

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

AGRADECIMIENTOS

Quiero agradecer a todas las personas que han sido mis sustentos durante todo este tiempo.

A mi mamá, por ser una persona especial, por apoyarme sin cuestionar mis decisiones y por desear lo mejor para mí. En compensación a todo el cariño y el amor que siempre me ha brindado, este es mi regalo para ella después de tantos años de sacrificio.

A mi papi, porque es una persona que no se detiene ante nada y ha sabido llenarme de toda esa fuerza para lograr mis propósitos. Le doy las gracias, porque a él, le debo lo que soy.

A mi hermano, porque ha sabido alcanzar todo lo que se propone y de alguna manera me ha transmitido ese espíritu cuando se trata de lograr lo que quiere.

A mis abuelas y mi tía, porque simplemente las adoro y siempre han estado conmigo.

A mi novio Aldo, por su paciencia, por todas las veces que me brindó su apoyo durante estos cinco años. Le agradezco por estar siempre ahí sin pedir nada a cambio.

A mis amigas de siempre Idalmis, Yanelis, Tamara y Anisley, porque juntas hemos pasado muy buenos momentos. Porque nunca me han fallado y me aceptan como soy, les doy muchas gracias. Las quiero.

Finalmente, quiero agradecer a mi grupo 8106, porque en él conocí a personas muy importantes en mi vida, surgieron grandes amistades, que estoy segura voy a extrañar muchísimo. Creo que no hubiese tenido sentido pasar por la Universidad sin conocerlos.

A la UCI por darme la oportunidad y a nuestro Comandante por forjar jóvenes como nosotros.

*A todos muchas gracias
Elizabeth*

DEDICATORIA

A mis padres por estar siempre conmigo y apoyarme en todas mis decisiones. Porque estoy segura de que están orgullosos de mí por llegar hasta el final. Le doy gracias por saber darme fuerzas, por confiar en mí y por inculcarme que no importan los obstáculos que se interpongan, siempre hay que salir adelante.

A mi hermano, por estar a mi lado cuando lo necesité, porque muchas veces sin darse cuenta me ayudó esa seguridad que lo caracteriza como persona.

A mis abuelos porque son lo más grande que tengo en la vida.

A mi tía Odalys porque es una persona que admiro y aunque hoy no se encuentra aquí conmigo, sé que está muy orgullosa de mí.

A toda mi familia le dedico uno de los momentos más importantes de mi vida.

Elizabeth

RESUMEN

RESUMEN

El presente trabajo consiste en el análisis del módulo Gestión de Tiempo, que permita aplicar la Gestión de Portafolios a los diferentes proyectos que se desarrollan en la Universidad de las Ciencias Informáticas y en el país. El sistema que se propone contribuye a una mejor planificación del tiempo de varios proyectos, con el objetivo de que los mismos se entreguen en la fecha requerida por el cliente. De ahí que el trabajo esté encaminado a la creación de una herramienta que integre todos los procesos de la Gestión de Tiempo, descritos en La Guía de los Fundamentos de la Dirección de Proyectos (*Guía del PMBOK*), permitiendo aplicar la Gestión de Portafolios a todos los proyectos que se desarrollan, no sólo en la Universidad, sino también en Cuba. El desarrollo de este trabajo facilitará la labor de las personas que están vinculadas directamente con los diferentes proyectos, logrando con ello una mejor centralización y organización de los mismos.

Palabras Claves

Proyecto, Gestión de Tiempo, Gestión de Portafolios de Proyectos

ÍNDICE

Índice de Contenidos

| | |
|---|----|
| <i>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</i> | 6 |
| <i>INTRODUCCIÓN</i> | 6 |
| <i>1.1 GESTIÓN DE PORTAFOLIOS</i> | 6 |
| 1.1.1 <i>GESTIÓN DE PORTAFOLIOS</i> | 6 |
| <i>1.2 GESTIÓN DE PROYECTOS</i> | 6 |
| 1.2.1 <i>GESTIÓN DE TIEMPO</i> | 7 |
| 1.2.2 <i>PRINCIPALES EXPONENTES DE LA GESTIÓN DE TIEMPO</i> | 7 |
| 1.2.2.1 <i>PMBOK</i> | 7 |
| 1.2.2.2 <i>IPMA</i> | 11 |
| 1.2.3 <i>FUNDAMENTACIÓN DE LA SELECCIÓN DE PMBOK</i> | 12 |
| <i>1.3 SISTEMAS INFORMÁTICOS RELACIONADOS CON LA GESTIÓN DE PORTAFOLIOS</i> | 12 |
| 1.3.1 <i>MICROSOFT ENTERPRISE PROJECT MANAGEMENT (EPM)</i> | 12 |
| 1.3.2 <i>ORACLE PROJECT PORTFOLIO MANAGEMENT</i> | 14 |
| 1.3.3 <i>RATIONAL FOCAL POINT</i> | 14 |
| <i>1.4 ANÁLISIS DE LOS SISTEMAS INFORMÁTICOS</i> | 15 |
| <i>1.5 METODOLOGÍA Y HERRAMIENTAS PARA LA MODELACIÓN DE SOFTWARE</i> ... | 17 |
| 1.5.1 <i>METODOLOGÍA DE DESARROLLO</i> | 17 |
| 1.5.2 <i>LENGUAJE DE MODELADO</i> | 18 |
| 1.5.2.1 <i>UML (Unified Modeling Language)</i> | 18 |
| 1.5.3 <i>HERRAMIENTA CASE</i> | 18 |
| 1.5.3.1 <i>Visual Paradigm</i> | 18 |
| <i>1.6 INGENIERÍA DE REQUERIMIENTOS</i> | 19 |
| 1.6.1 <i>CARACTERÍSTICAS DE LOS REQUERIMIENTOS</i> | 20 |
| 1.6.2 <i>REQUISITOS FUNCIONALES Y NO FUNCIONALES</i> | 20 |
| 1.6.2.1 <i>Categoría de los requisitos no funcionales</i> | 21 |
| <i>1.7 PATRONES DE CASOS DE USO</i> | 22 |
| <i>1.8 CAPTURA DE REQUISITOS</i> | 23 |
| 1.8.1 <i>ENTREVISTAS</i> | 24 |
| 1.8.2 <i>CUESTIONARIOS</i> | 25 |
| 1.8.3 <i>DESARROLLO CONJUNTO DE APLICACIONES (JAD)</i> | 25 |
| 1.8.4 <i>TORMENTA DE IDEAS O BRAINSTORMING</i> | 26 |
| 1.8.5 <i>CASOS DE USOS</i> | 26 |
| <i>1.9 CONCLUSIONES</i> | 27 |
| <i>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</i> | 28 |
| <i>INTRODUCCIÓN</i> | 28 |

ÍNDICE

| | |
|--|--------------------------------------|
| 2.1 PROPUESTA DEL SISTEMA | 28 |
| 2.2 ¿QUÉ ES UN MODELO DEL DOMINIO? | 30 |
| 2.2.1 GLOSARIO DE TÉRMINOS..... | 30 |
| 2.2.2 DIAGRAMA CONCEPTUAL DEL MODELO DE DOMINIO | 31 |
| 2.3 RESULTADOS DE LA APLICACIÓN DE LAS TÉCNICAS DE CAPTURA DE REQUISITOS..... | 31 |
| 2.4 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE | 32 |
| 2.4.1 REQUISITOS FUNCIONALES..... | 32 |
| 2.4.2 REQUISITOS NO FUNCIONALES..... | 35 |
| 2.5 DEFINICIÓN DE LOS CASOS DE USO | 38 |
| 2.5.1 DEFINICIÓN DE LOS ACTORES DEL SISTEMA..... | 38 |
| 2.5.2 DIAGRAMA DE PAQUETES | 38 |
| 2.5.3 DIAGRAMAS DE CASOS DE USO..... | 39 |
| 2.5.4 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA | 40 |
| 2.5.4.1 Descripción del Caso de Uso Gestionar Actividad..... | ¡Error! Marcador no definido. |
| 2.5.4.2 Descripción del Caso de Uso Gestionar Predecesores ... | ¡Error! Marcador no definido. |
| 2.5.4.3 Descripción del Caso de Uso Mostrar Diagramas | ¡Error! Marcador no definido. |
| 2.5.4.4 Descripción del Caso de Uso Adicionar actividad recurrente | ¡Error! Marcador no definido. |
| 2.6 CONCLUSIONES | 41 |
| CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS | 42 |
| 3.1 TÉCNICAS DE VALIDACIÓN DE REQUISITOS..... | 42 |
| 3.2 PASOS PARA LA VALIDACIÓN DE REQUISITOS | 42 |
| 3.3 MÉTRICAS PARA LA VALIDACIÓN DE REQUISITOS | 43 |
| 3.4 ESTRATEGIA PARA LA VALIDACIÓN..... | 44 |
| 3.4.1 FASE 1 VALIDACIÓN DEL LISTADO DE REQUISITOS | 44 |
| 3.4.1.1 Revisiones..... | 44 |
| 3.4.1.2 Lista de Chequeo para validar el listado de requisitos..... | 45 |
| 3.4.2 FASE 2 VALIDACIÓN DEL DOCUMENTO ESPECIFICACIÓN DE REQUISITOS..... | 47 |
| 3.4.2.1 Métricas para Calidad de la Especificación de Requisitos | 47 |
| 3.4.2.2 Lista de Chequeo para validar del documento ERS | 49 |
| 3.4.3 FASE 3 VALIDACIÓN DEL DOCUMENTO MODELO DE CASOS DE USO DEL SISTEMA..... | 51 |
| 3.4.3.1 Métricas para la validación del Modelo de Casos de Uso del Sistema | 51 |
| 3.4.3.2 Matriz de Trazabilidad | 55 |
| 3.5 CASOS DE PRUEBA..... | 56 |
| 3.5.1 PRUEBAS DEL SISTEMA | 56 |
| 3.6 CONCLUSIONES | 56 |

ÍNDICE

| | |
|----------------------------|--------------------------------------|
| CONCLUSIONES..... | 57 |
| RECOMENDACIONES..... | 58 |
| BIBLIOGRAFIA..... | ¡ERROR! MARCADOR NO DEFINIDO. |
| ANEXOS..... | ¡ERROR! MARCADOR NO DEFINIDO. |
| GLOSARIO DE TÉRMINOS | 63 |

ÍNDICE

Índice de Figuras

| | |
|---|----|
| Figura 1: Método de Diagramación por Precedencia (PMI, 2004) | 9 |
| Figura 2: Método de Diagramación con Flechas (PMI, 2004) | 10 |
| Figura 3: Modelo de Dominio..... | 31 |
| Figura 4: Diagrama de Paquetes..... | 39 |
| Figura 5: Diagrama de Casos de uso del Paquete de Reportes..... | 39 |
| Figura 6: Diagrama de Casos de uso del Paquete de Funciones Básicas | 40 |
| Figura 7: Diagrama de Casos de uso del Paquete de Funciones Generales | 40 |
| Figura 8: Resultados de las Revisiones..... | 45 |
| Figura 9: Valores de cada factor aplicados al Modelo de Casos de Uso | 54 |

ÍNDICE

Índice de Tablas

| | |
|--|--------------------------------------|
| Tabla 1: Comparación de Herramientas | 16 |
| Tabla 2: Actores del sistema | 38 |
| Tabla 3: CU Gestionar Actividad | ¡Error! Marcador no definido. |
| Tabla 4: CU Gestionar Predecesores | ¡Error! Marcador no definido. |
| Tabla 5: CU Mostrar Diagramas | ¡Error! Marcador no definido. |
| Tabla 6: CU Adicionar actividad recurrente | ¡Error! Marcador no definido. |
| Tabla 7: Características que deben cumplir los requisitos | 44 |
| Tabla 8: Lista de verificación del listado de requisitos | 46 |
| Tabla 9: Lista de chequeo para validar el documento ERS | 51 |
| Tabla 10: Factores vinculados a las métricas de calidad | 54 |
| Tabla 11: Matriz de Trazabilidad | 56 |
| Tabla 12: Caso de Prueba: CU Gestionar Actividad ... | ¡Error! Marcador no definido. |
| Tabla 13: Escenario Adicionar campos de tiempo a una actividad. | ¡Error! Marcador no definido. |
| Tabla 14: Escenario Modificar campos de tiempo a una actividad. | ¡Error! Marcador no definido. |
| Tabla 15: Escenario Mostrar información de tiempo de una actividad | ¡Error! Marcador no definido. |
| Tabla 16: Caso de Prueba: CU Gestionar Predecesores | ¡Error! Marcador no definido. |
| Tabla 17: Escenario Adicionar predecesores | ¡Error! Marcador no definido. |
| Tabla 18: Escenario Mostrar información de predecesores | ¡Error! Marcador no definido. |
| Tabla 19: Caso de Prueba: CU Mostrar Diagramas | ¡Error! Marcador no definido. |

ÍNDICE

Tabla 20: Caso de Prueba: CU Adicionar actividad recurrente. ;**Error! Marcador no definido.**

Tabla 21: Escenario Adicionar actividad recurrente ;**Error! Marcador no definido.**

INTRODUCCIÓN

INTRODUCCIÓN

Un proyecto surge de una idea; convertirlo en un producto usable requiere de una eficiente gestión, que sería en este caso la encargada de organizar y administrar recursos, de manera que se pueda culminar el trabajo requerido en el proyecto dentro del alcance, tiempo y costes definidos (Nunes, 2008).

La Gestión de Proyectos es utilizada en diferentes campos como por ejemplo en los bancos, desarrollo de sistemas, en telecomunicaciones, entre otros. Esta área es de relevancia a nivel mundial, debido a que se ha convertido en una necesidad para los proyectos que se llevan a cabo en las grandes empresas, logrando una alta competitividad a nivel internacional, lo que ha provocado la aparición de proyectos más complejos que requieren de habilidades más específicas. La Gestión de Proyectos provee una metodología ordenada y rigurosa para administrar los proyectos, además de nuevas alternativas de organización, control y seguimiento a las empresas.

En una empresa se pueden llevar a cabo un conjunto de proyectos, lo que implica la necesidad de obtener un mejor control sobre los mismos para facilitar el manejo efectivo del trabajo y maximizar las posibilidades de éxito. Es decir, la aplicación de la Gestión de Portafolios, que no es más que el manejo centralizado de uno o más portafolios y envuelve la identificación, priorización, autorización, gestión y control de proyectos, programas y otros trabajos relacionados, que permite alcanzar las metas estratégicas del negocio (Saborío, 2006).

Inmerso en la Gestión de Portafolios de Proyectos se encuentra la Gestión de Tiempo, que no sólo posibilita una mejor planificación para un proyecto determinado sino que se aplica a todo un conjunto de proyectos. Es un proceso de planeación que permite desarrollar con prontitud tareas determinadas e incluye los procesos necesarios para la culminación del proyecto a tiempo. La realización eficaz de una Gestión de Tiempo brinda a las empresas una mejor productividad y eleva la competitividad, además de facilitar la toma de decisiones. En la Gestión de Tiempo desempeña un papel fundamental la planificación, que posibilita una mejor organización para el desarrollo de diversos proyectos, además se realizan las asignaciones de recursos disponibles, para lo que es necesario estimar costos y plazos para los proyectos a realizar. Sin embargo, hoy día resulta muy difícil planificarse; los negocios van camino al fracaso por la ausencia de un plan de proyecto que distribuya bien el tiempo y las responsabilidades, lo que contribuye a una mala calidad de la Gestión de Tiempo.

INTRODUCCIÓN

Para la Gestión de Portafolios existen algunas herramientas en su mayoría propietarias, como Microsoft Office Enterprise Project Management (EPM) que permite la administración centralizada de portafolios de proyectos, además Primavera Enterprise Project Portfolio Management, entre otras. Estas herramientas poseen un elevado costo en el mercado que impide que el país pueda utilizarlas, por ejemplo, EPM tiene un costo de 999.95 USD, mientras que Primavera tiene un valor de 1794.00 euros, lo que promedia un total de 1396.9 euros. Esto dificulta que el país no pueda aplicar la Gestión de Portafolios, ya que no puede utilizar dichas herramientas pues presentan un elevado costo en el mercado y son propietarias.

En Cuba se han realizado grandes esfuerzos por mantenerse actualizado con respecto a las nuevas tecnologías que se han venido desarrollando, por lo que una de sus principales tareas no es sólo la informatización de la sociedad, sino insertarse en el mundo del software por sus perspectivas económicas. Un ejemplo clave de cómo se está llevando a cabo este proceso es la Universidad de las Ciencias Informáticas (UCI), en la cual se organizó una Infraestructura Productiva (IP) que se encarga de gestionar los proyectos de producción de software. A lo largo de su proceso de maduración por lograr una producción con la calidad requerida surge la idea de los centros productivos, constituido por pequeñas empresas encaminadas a la producción de software en una rama establecida. De ahí el surgimiento del Centro de Tecnologías de Gestión de Datos (DATEC), donde se desarrollan una serie de proyectos como por ejemplo el Sistema de Gestión, en el área de comunicación e información del Ministerio del Poder Popular de la Comunicación y la Información (MINCI), en el cual no se realiza una Gestión de Portafolios que permita gestionar todos los subproyectos que se llevan a cabo de forma centralizada, sino que se gestionan de manera funcional e independiente. Por lo que el centro, específicamente la línea de Soluciones Integrales necesita desarrollar una herramienta que posibilite a proyectos como este aplicar la Gestión de Portafolios para un mejor control y centralización de los mismos.

La Universidad además, está vinculada a la empresa cubana Albet Ingeniería y Sistemas, institución en la cual se llevan a cabo diferentes líneas de trabajo, donde se ofertan diversas soluciones desarrolladas de acuerdo a las necesidades de los clientes. Esta empresa es la encargada de gestionar todos los proyectos de software que se realizan en el país y la Universidad, por lo que también se hace necesario la utilización de una herramienta que les proporcione una mejor centralización y organización de los proyectos que se desarrollan.

De ahí la necesidad de la creación de una nueva herramienta que facilite a la Universidad, empresas como Albet o proyectos como el MINCI aplicar una Gestión de Portafolios que les permita gestionar

INTRODUCCIÓN

paquetes de proyectos para una mejor organización y centralización, haciendo énfasis en la Gestión de Tiempo ya que no sería la planeación de un proyecto específico sino de un grupo de proyectos logrando que los procesos de esta área se ejecuten de manera eficiente.

Por otro lado la carencia de un sistema automatizado que soporte el proceso de Gestión de Tiempo de Portafolios de Proyectos en la Universidad, está provocando que los proyectos se planifiquen de forma independiente debido a la falta de integración de la información de los mismos. Esto impide la capacidad de analizar adecuadamente la totalidad del portafolio de los proyectos y además se afecta la entrega oportuna y correcto establecimiento de secuenciación de las actividades.

Debido a esta situación se necesita una plataforma que permita monitorear de forma continua los proyectos y los portafolios para asegurar que se adopten diversas acciones claves para la toma de decisiones, garantizando que se entreguen de forma oportuna los proyectos asociados a cada uno de los portafolios.

Problema a resolver

¿Cómo realizar el análisis para el establecimiento de una adecuada secuencia de actividades y estimación de estas para una gestión eficiente del tiempo de un portafolio de proyectos?

Objeto de Estudio

Gestión de Portafolios de Proyectos.

Objetivo general

Realizar el análisis del módulo Gestión de Tiempo del Sistema Integral para la Gestión de Portafolios que posibilite posteriormente su diseño e implementación.

Objetivos específicos

- ✓ Obtener el modelo de dominio.
- ✓ Identificar los requerimientos funcionales y no funcionales del módulo Gestión de Tiempo.
- ✓ Modelar los casos de uso del sistema.
- ✓ Validar los requisitos del sistema.

Campo de acción

INTRODUCCIÓN

Gestión de Tiempo de Portafolios de Proyectos.

Idea a Defender

El desarrollo del análisis de forma correcta y detallada del módulo Gestión de Tiempo, permitirá sentar las bases para el posterior diseño e implementación, así como una mejor planificación del tiempo de un conjunto de proyectos.

Tareas de la investigación

- ✓ Sistematización del estado actual de la Gestión de Portafolios de Proyectos, específicamente la Gestión de Tiempo.
- ✓ Estudio y evaluación de los sistemas informáticos de Gestión de Portafolios de Proyectos.
- ✓ Análisis del proceso de Ingeniería de Requisitos.
- ✓ Realización del flujo de análisis del módulo Gestión de Tiempo.
- ✓ Validación de los requisitos identificados a través de técnicas y métricas.

Métodos Científicos

Métodos Empíricos

Entrevistas: Permitió realizar un conjunto de entrevistas a personas especializadas en el tema de la Gestión de Portafolios de Proyectos, aportando datos de relevancia a la investigación.

Métodos Teóricos

Histórico–Lógico: Permitió estudiar la trayectoria real del objeto de estudio y los acontecimientos a lo largo de la historia.

Análisis–Sintético: Este método sirvió para profundizar en el conocimiento de los conceptos fundamentales relacionados con la Gestión de Portafolios de Proyectos y la Gestión de Tiempo.

Aportes esperados de la Investigación

Debido al gran desarrollo de las tecnologías a nivel universal en la actualidad el uso de una herramienta para la Gestión de Portafolios de Proyectos será un impacto no sólo social, sino económico tanto para la Universidad como para Cuba. Aportará al país una herramienta que permitirá la organización y

INTRODUCCIÓN

centralización de los diferentes proyectos que se llevan a cabo en todo tipo de instituciones. Facilitará el trabajo de las personas que se relacionan directamente con los diferentes proyectos. Contribuirá además a la realización de diversos proyectos en el país que aportará resultados de relevancia al desarrollo del mismo. Además, otro aporte importante es la investigación que se realiza, donde se demuestra la necesidad de contar con una herramienta que permita aplicar la Gestión de Portafolios de Proyectos.

Estructura del documento

En el **Capítulo 1** se describen los conceptos asociados a la Gestión de Portafolios de Proyectos y a la Gestión de Tiempo. Además, se fundamentan las técnicas, metodologías, herramientas y lenguajes a utilizar para el posterior análisis del módulo Gestión de Tiempo.

En el **Capítulo 2** se describe la solución propuesta, donde se presenta el modelo de dominio y el diagrama de casos de uso del sistema. Se determinan las funcionalidades que tendrá el sistema a desarrollar, así como los casos de uso propuestos con sus respectivas descripciones.

El **Capítulo 3** es donde se realiza el análisis de los resultados, donde se hace uso de una metodología para la validación de los artefactos, se aplican un conjunto de técnicas y se elaboran las pruebas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se describen los conceptos fundamentales asociados a la Gestión de Portafolios de Proyectos. Se realiza un análisis del proceso Gestión de Tiempo, así como sus principales conceptos. Además, se establece la metodología de trabajo y las herramientas informáticas a utilizar.

1.1 Gestión de Portafolios

1.1.1 Gestión de Portafolios

Un portafolio se refiere a una colección de proyectos o programas y otros trabajos que son agrupados para facilitar una administración más efectiva. La Gestión de Portafolios se refiere a la administración centralizada de uno o más portafolios e incluye identificación, priorización, autorización, gestión y control de proyectos, programas y otros trabajos relacionados para alcanzar determinados objetivos estratégicos del negocio. Se centra en garantizar que los proyectos y programas sean revisados para dar prioridad a la asignación de recursos y que la Gestión de Portafolios sea consistente y esté alineada con las estrategias de la organización (Ronald Saborío, 2006).

Es una herramienta que da soporte a la toma de decisiones usado por los ejecutivos para balancear de forma continua los riesgos y beneficios de un portafolio.

1.2 Gestión de Proyectos

La Gestión de Proyectos ha cobrado un gran auge en la actualidad por la importancia que posee a la hora de gestionar un proyecto determinado, pues constituye un factor clave para determinar su éxito o fracaso. Existen diferentes definiciones sobre la Gestión de Proyectos que serán expuestas a continuación:

- ✓ La Gestión de Proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto. La Gestión de Proyectos se logra mediante la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento, control y cierre (Project Management Institute, 2004).
- ✓ Se describe como un proceso de planteamiento, ejecución y control de un proyecto, desde su comienzo hasta su conclusión con el propósito de alcanzar un objetivo en un plazo de tiempo, coste y nivel de calidad determinados, a través de la movilización de recursos técnicos, financieros y humanos (Nunes, 2008).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

De manera general la Gestión de Proyectos no es más que la planificación de un proyecto para alcanzar una meta específica en un período de tiempo determinado, con un coste y calidad definidos, de manera que los procesos que intervienen en la Gestión de Proyectos estén integrados con el objetivo de lograr la culminación del proyecto de manera satisfactoria. Uno de los procesos fundamentales es la Gestión de Tiempo, que incluye una serie de procesos que serán descritos más adelante.

1.2.1 Gestión de Tiempo

La Gestión de Tiempo del Proyecto incluye los procesos necesarios para lograr la conclusión del proyecto a tiempo.

1.2.2 Principales exponentes de la Gestión de Tiempo

1.2.2.1 PMBOK

La Guía de los Fundamentos de la Dirección de Proyectos (*Guía del PMBOK*) es un modelo reconocido para la Gestión de Proyectos. Al igual que con otras profesiones tales como las leyes y la medicina, el conocimiento comprendido dentro de dicho modelo ha evolucionado de las buenas prácticas de los profesionales de la Gestión de Proyectos, quienes han contribuido al desarrollo de esta guía. En esta se define la Gestión de Proyectos y conceptos relacionados con este tema.

Una de las secciones que describe esta guía se denomina Áreas de Conocimientos, en la cual está incluida la Gestión de Tiempo que detalla los procesos necesarios para lograr la conclusión del proyecto a tiempo. Estos procesos contienen lo siguiente:

- I. **Definición de las Actividades:** identifica las actividades específicas del cronograma que deben ser realizadas para producir los diferentes productos entregables del proyecto.
- II. **Establecimiento de la Secuencia de Actividades:** identifica y documenta las dependencias entre las actividades del cronograma.
- III. **Estimación de Recursos de las Actividades:** estima el tipo y las cantidades de recursos necesarios para realizar cada actividad del cronograma.
- IV. **Estimación de la Duración de las Actividades:** estima la cantidad de períodos laborables que serán necesarios para completar cada actividad del cronograma.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- V. **Desarrollo del Cronograma:** analiza las secuencias de las actividades, la duración de las actividades, los requisitos de recursos y las restricciones del cronograma para crear el cronograma del proyecto.
- VI. **Control del Cronograma:** controla los cambios del cronograma del proyecto (Project Management Institute, 2004).

En algunos proyectos el establecimiento de la secuencia de las actividades, la estimación de recursos de las actividades, la estimación de la duración de las actividades y el desarrollo del cronograma están tan estrechamente relacionados, que se consideran un proceso único que se puede realizar en un período de tiempo relativamente corto (Project Management Institute, 2004).

Para estos procesos existen una serie de herramientas y técnicas diferentes. A continuación se abordará de manera muy general las características de algunas de ellas. Para el Establecimiento de la Secuencia de Actividades una de las técnicas que se utilizan es el Método de Diagramación por Precedencia (PDM). Es un método para crear un diagrama de red del cronograma del proyecto que utiliza nodos que simbolizan las actividades que están conectadas con flechas para mostrar las dependencias. Este método es utilizado por la mayoría de los paquetes de software de la Gestión de Proyectos. Incluye cuatro tipos de dependencia o relaciones de precedencia:

- ✓ **Final a Inicio**
- ✓ **Final a Final**
- ✓ **Inicio a Inicio**
- ✓ **Inicio a Fin**

En este método la relación de precedencia más comúnmente usada es final a inicio (**Ver Figura 1**).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

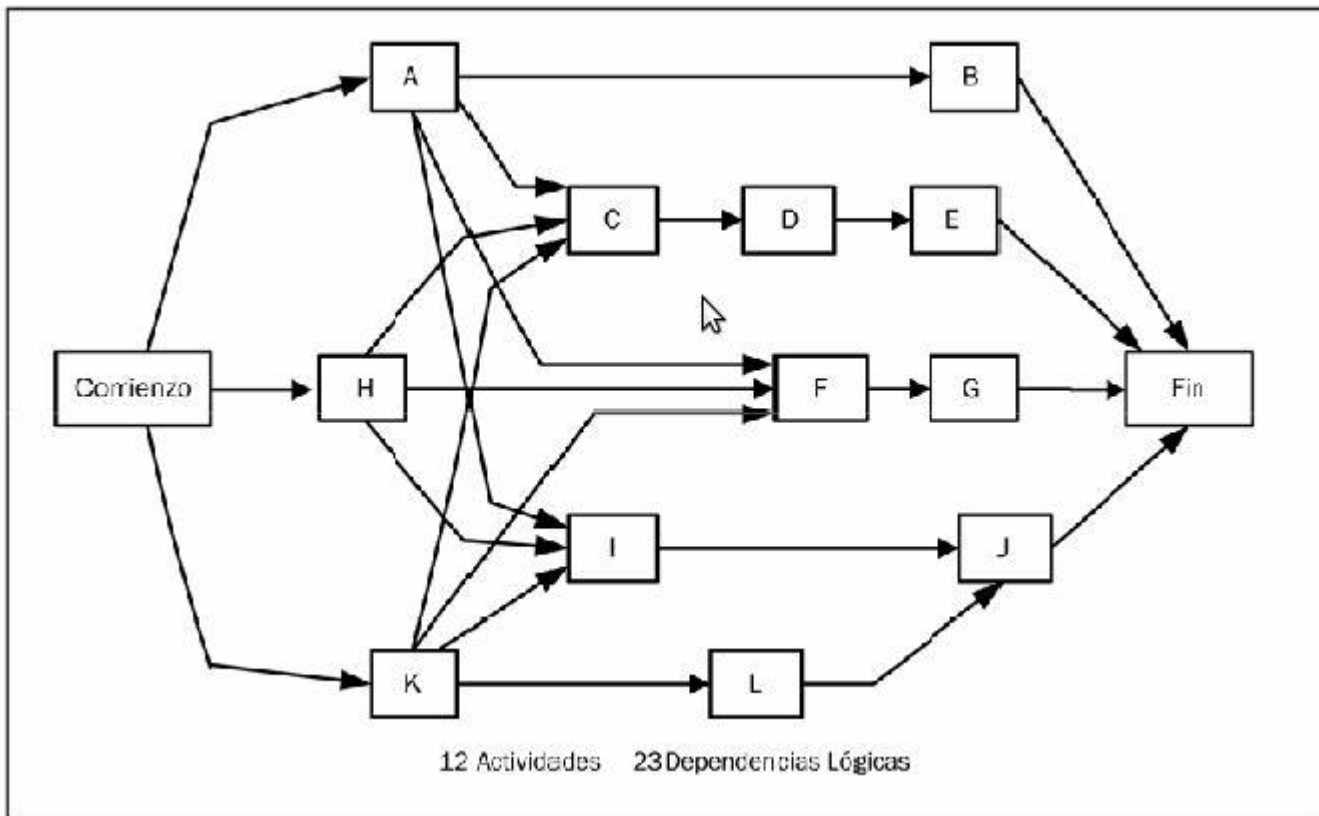


Figura 1: Método de Diagramación por Precedencia (PMI, 2004)

Otra técnica importante es el Método de Diagramación con Flechas (Ver Figura 2). Es un método para crear un diagrama de red del cronograma de proyecto, que en este caso utiliza flechas para representar las actividades, que se conectan en nodos para mostrar las dependencias. Utiliza sólo dependencia final a inicio y puede requerir el uso de relaciones ficticias, denominadas actividades ficticias, las cuales se muestran con una línea de puntos (Project Management Institute, 2004).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

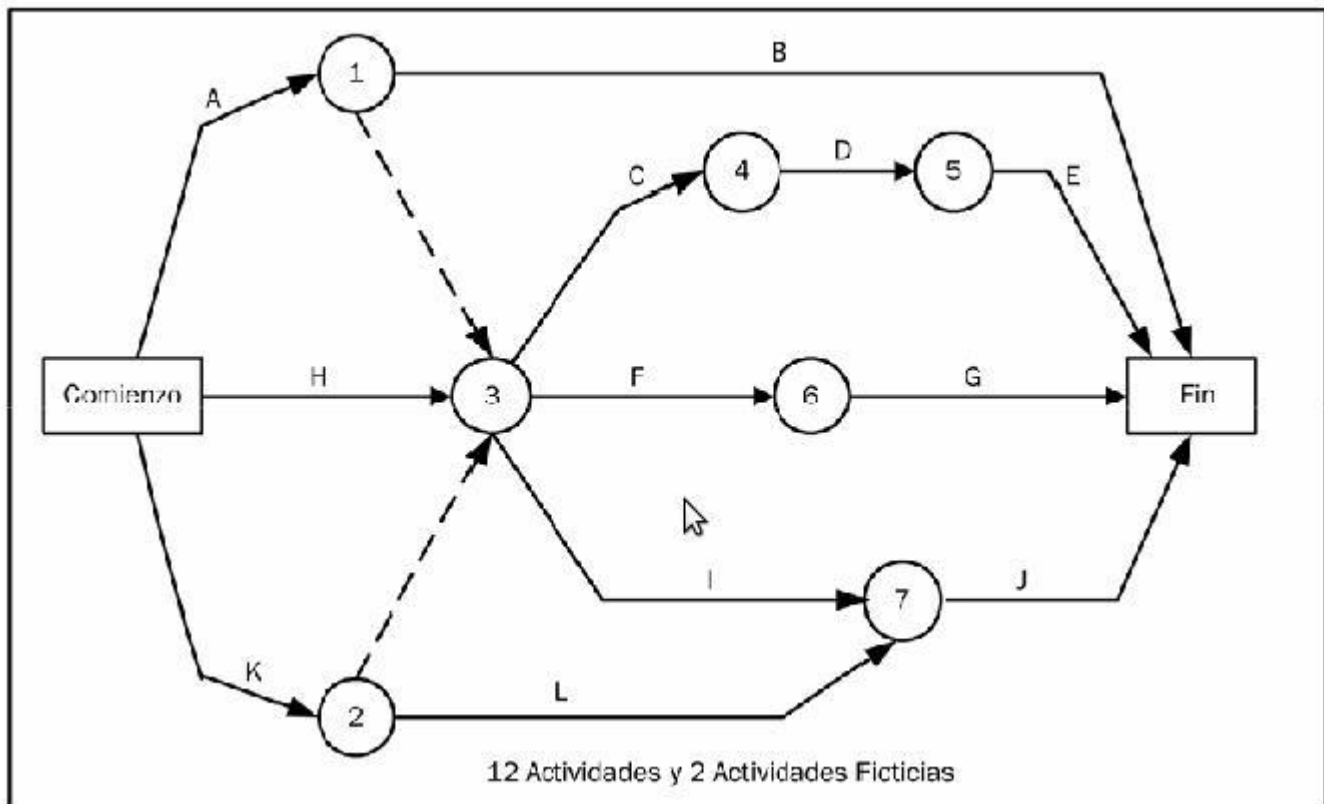


Figura 2: Método de Diagramación con Flechas (PMI, 2004)

En la Estimación de Recursos de las Actividades una de las técnicas más importantes es la Estimación Ascendente donde se estiman las necesidades de los recursos de cada una de las partes inferiores y más detalladas del trabajo, donde estas estimaciones se suman en una cantidad total para cada uno de los recursos de la actividad del cronograma. En caso de existir dependencias entre las actividades del cronograma, este patrón de uso de recursos se refleja en los requisitos estimados de la actividad del cronograma y se documenta (Project Management Institute, 2004).

Por último, pero no menos importante, la Estimación de la Duración de las Actividades que utiliza técnicas como por ejemplo la Estimación por Analogía, que significa utilizar la duración real de una actividad del cronograma anterior como base para la estimación de la duración de una actividad del cronograma futuro. Esta técnica es más fiable cuando las actividades previas son realmente similares y no sólo en apariencia,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

y los miembros del equipo de proyecto que preparan las estimaciones tienen la experiencia necesaria (Project Management Institute, 2004).

La Estimación Paramétrica es otra de las técnicas utilizada por este proceso, que plantea que la estimación de las bases para la duración de las actividades se puede determinar de manera cuantitativa multiplicando la cantidad de trabajo a realizar por los ratios de productividad. Para determinar la duración de la actividad en períodos laborables se multiplican las cantidades totales de recursos por las horas de trabajo por período laborable o la capacidad de producción por el período laborable y se dividen por la cantidad de recursos que se aplican.

Existe además, las Estimaciones por tres valores que se basa en determinar tres tipos de estimaciones:

- ✓ Más probable.
- ✓ Pesimista.
- ✓ Optimista.

Para elaborar una estimación de la duración de la actividad se utiliza el promedio de las tres duraciones estimadas. Este promedio suministra una estimación de la duración de la actividad más precisa (Project Management Institute, 2004).

1.2.2.2 IPMA

Puede ser utilizada como guía para la preparación de materiales de capacitación y como documento de referencia general para la búsqueda de información sobre la aplicación de proyectos. Ofrece acceso a elementos de competencia técnica, de comportamiento y contextual de la Gestión de Proyectos. Contiene además, términos básicos, funciones, habilidades, métodos, técnicas y herramientas que son usadas en una buena Gestión de Proyectos (Management(IPMA), 2006).

En él se refiere al tiempo como un proceso que cubre la estructuración, secuenciación, duración, estimación y la programación de las actividades, incluyendo la asignación de recursos a las mismas, estableciendo plazos para los proyectos y monitoreando y controlando su ejecución. La programación del tiempo depende de la prioridad que posea una tarea determinada y de la disponibilidad de los recursos (Management(IPMA), 2006).

Entre sus procesos se encuentran:

- ✓ Definición y secuenciación de las actividades.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Estimación de la duración de las actividades.
- ✓ Horario del proyecto.
- ✓ Asignación de recursos.
- ✓ Comparar las metas, las fechas actuales y previstas y actualizar este pronóstico en caso de ser necesario.
- ✓ Controlar el tiempo de programación con respecto a los cambios.
- ✓ Documento de las lecciones aprendidas y aplicarlas a futuros proyectos (Management(IPMA), 2006).

1.2.3 Fundamentación de la selección de PMBOK

Luego del análisis realizado anteriormente se llegó a la conclusión de que La Guía de los Fundamentos de la Dirección de Proyectos (*Guía del PMBOK*) es el principal exponente de la Gestión de Tiempo, pues define los procesos necesarios para que el proyecto culmine en la fecha requerida, brinda una detallada descripción de cada proceso planteando los elementos de entradas y salidas, así como las técnicas y herramientas que utilizan.

1.3 Sistemas informáticos relacionados con la Gestión de Portafolios.

La Gestión de Portafolios es de vital importancia en el desarrollo de proyectos de software. En la actualidad existen una serie de herramientas a nivel mundial, en su mayoría propietarias, que permiten la Gestión de Portafolios de Proyectos y obligan a los clientes a aceptar soluciones con la personalización limitada o exigen la compra de una costosa licencia. Las herramientas que se muestran más adelante fueron valoradas como posibles soluciones teniendo en cuenta las funcionalidades que presentan. A continuación se hace un análisis de las herramientas de uso actual.

1.3.1 Microsoft Enterprise Project Management (EPM)

Es una solución de Project Portfolio Management que permite compartir y administrar los recursos de forma centralizada. Ayuda a las organizaciones a obtener una mejor visibilidad, información detallada y

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

control de todos los trabajos. Mejora la toma de decisiones, maximiza el uso de los recursos e incrementa la eficacia operacional (Microsoft Corporation, 2010).

La solución EPM incluye productos de Microsoft Office Project 2007 con el objetivo de proporcionar a las organizaciones una solución completa de la Gestión de Portafolios de Proyectos. Estos productos son:

- ✓ **Microsoft Office Project Professional 2007**
- ✓ **Microsoft Office Project Server 2007**
- ✓ **Microsoft Office Project Portfolio Server 2007**

Ventajas que posee la solución EPM:

- ✓ Administra y controla todos los tipos de trabajo, ya que administra desde propuestas sencillas hasta programas de proyectos complejos.
- ✓ Mejora la visibilidad y la información detallada para mejorar la toma de decisiones, donde se seleccionan los portafolios de proyectos que más se adecuan a la estrategia empresarial y proporciona vistas de los informes para obtener una mayor claridad en los proyectos, programas y aplicaciones.
- ✓ Permite la comunicación y la colaboración eficazmente con todos los participantes del proyecto, se mantiene la sintonía de los equipos con las asignaciones de las tareas y se realiza el seguimiento de los proyectos dentro y fuera de la oficina.
- ✓ Administra eficazmente los recursos, pues ayuda a crear planes de recursos para sintonizar las contrataciones y subcontrataciones estratégicas con los objetivos empresariales a largo plazo.
- ✓ Realiza el seguimiento de todos los proyectos, programas y aplicaciones durante todo el ciclo de vida, de manera que se puedan identificar los posibles problemas, tomar decisiones y garantizar que los portafolios maximicen el rendimiento de la inversión (Microsoft Corporation, 2010).

Posibilita, además, la administración de portafolios de proyectos, identifica, prioriza y selecciona mediante simulaciones los proyectos que soportan los objetivos estratégicos y alinea los recursos eficientemente. Pueden interactuar además, los diferentes niveles involucrados: los ejecutivos pueden ver el estado de los

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

proyectos, el administrador de proyectos puede asignar responsabilidades y el equipo de trabajo puede actualizar el avance de sus actividades.

1.3.2 Oracle Project Portfolio Management

Es una herramienta que proporciona soluciones que ayudan a las empresas a conseguir y mantener una ventaja estratégica. Sustenta, además el ciclo de vida completo de la Gestión de Proyectos y Portafolios dentro de una perspectiva única y exacta de todas las actividades relacionadas con el proyecto. Ha sido reconocida gracias a las sólidas funcionalidades de contabilidad que ofrece. Ayuda durante todo el ciclo de vida de la gestión global e incluye contratación, planificación de proyectos, estimación de presupuestos, provisiones, gestión de recursos, colaboración, monitorización de riesgos y contabilidad de proyectos. Permite a las empresas ejecutar proyectos y programas con diferentes divisas, idiomas y unidades de negocio. Ofrece un único almacén con información del proyecto completa y segura a la que pueden acceder todos los miembros del equipo de trabajo, permite además, el control eficaz de los cambios y problemas que surgen en el proyecto, así como la posibilidad de compartir la información del proyecto en cualquier momento y lugar (Oracle, 2010).

Presenta una serie de ventajas significativas que lo hacen diferente de otras herramientas:

- ✓ Clasifica los proyectos mediante un conjunto de criterios de evaluación clave con el análisis de portafolios.
- ✓ Establece un criterio común a la hora de seleccionar los proyectos con el análisis de portafolios.
- ✓ Supervisa continuamente el progreso de los proyectos.
- ✓ Organiza la disponibilidad conjuntamente con la capacitación de los recursos con las futuras necesidades.
- ✓ Permite aumentar la productividad con funcionalidades de colaboración agilizadas (Oracle, 2010).

1.3.3 Rational Focal Point

Rational Focal Point es dirigido a la Gestión de Portafolios ayudando a ejecutivos y equipos de trabajo a tomar las decisiones correctas en la ejecución de un negocio determinado. Captura las entradas de los clientes, analistas y prioriza cuál de ellos proporcionará mayor valor al negocio. Utiliza visualización,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

priorización y las capacidades de planificación para asegurar que los planes sean innovadores. Integra el plan de arquitectura con la ejecución de proyectos dentro de la Gestión de Portafolios de manera que las decisiones de la empresa estén alineadas con las necesidades financieras. Reduce los riesgos y mejora las estimaciones de costos de las inversiones (IBM, 2010).

1.4 Análisis de los sistemas informáticos

Después de haber analizado las características de las herramientas expuestas anteriormente se concluye que la Microsoft Enterprise Project Management (EPM) es la solución más completa, sin embargo, posee como desventaja que es una herramienta propietaria, por lo que, a pesar de tener una serie de beneficios claves para la Gestión de Portafolios de Proyectos, se le dificulta al país hacer uso de la misma por el elevado costo que tiene su licencia en el mercado mundial.

No obstante, existen herramientas de código abierto que también tributan a la Gestión de Portafolios de Proyectos, pero que no cumplen con un conjunto de funcionalidades. Un ejemplo claro de estas herramientas es Project.net, la cual está diseñada para capturar, mostrar, reportar y resolver las relaciones complejas de las organizaciones a la hora de planear mejores iniciativas. Es adoptada por cientos de corporaciones worldwide, proporciona a los administradores de proyectos y a las Oficinas de Gestión de Proyectos información detallada para mantener los proyectos alineados. Además, es desarrollado para resolver la mejor de las alternativas (Project.net, 2010).

Otro ejemplo lo constituye el DotProject, el cual brinda la posibilidad de gestionar un conjunto de proyectos. Esta herramienta se aplica a la Gestión de Tiempo, pues permite la realización de un conjunto de procesos que influyen en la culminación exitosa del proyecto. Por ejemplo, permite asignar estimación de esfuerzos por tarea y el calendario de cada tarea, donde cada una puede ser actualizada. A través de un gráfico de Gantt resume el plan de proyecto por usuario o por departamento, lo que posibilita que cada integrante del proyecto tenga una visualización de las tareas a realizar. Estas pueden actualizarse mediante un log de novedades, las cuales incluyen esfuerzos reales, re-planificación y los costos. A través de este log se puede obtener una base de datos para próximas estimaciones en proyectos de igual similitud. Permite además, la gestión y planificación de proyectos en entornos colaborativos, además de visualizar las horas asignadas por usuario o proyecto para un período de tiempo, así como las horas asignadas y las realmente incurridas, con el fin de extraer porcentajes de trabajos realizados y de eficiencia en base a tareas completadas (Fernández Lugo, 2009).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Después de realizar un análisis sobre estas herramientas es muy importante definir por qué utilizar soluciones de código abierto. Se dio respuesta a esta interrogante basada en una serie de ventajas que poseen dichas soluciones. Son adaptables, pues no limita las opciones del vendedor, que pueden ser modificadas por los desarrolladores de la empresa, quienes son libres de modificar la aplicación con el objetivo de satisfacer las necesidades de los clientes. Las aplicaciones de código abierto no implican costos de licencia y pueden ser adaptadas a las necesidades de una organización determinada. Presentan soluciones flexibles para satisfacer las necesidades de la Gestión de Portafolios de Proyectos. Proporcionan visibilidad de todos los proyectos y sus relaciones (Business Management, 2010).

Sin embargo, a pesar de ser herramientas de código abierto y gratuito, debido al análisis de sus características y desventajas, ninguna cumple con las particularidades que debe poseer una herramienta que garantice todos los procesos de la Gestión de Tiempo. De ahí la necesidad de analizar detalladamente todos los procesos que intervienen en la Gestión de Tiempo para garantizar una eficiente Gestión de Portafolios de Proyectos. **(Ver Tabla 1)**

| Característica | | Nivel de cumplimiento de las Herramientas | |
|-----------------------------------|--|---|---------------|
| | | ProjectNet (9.0.0) | Redmine (0.8) |
| Perfil del proyecto | | Insuficiente | No |
| Gestionar calendario y cronograma | | Suficiente | No |
| | Desarrollo del cronograma | Suficiente | Suficiente |
| | Gestionar tiempos muertos (vacaciones) | Suficiente | No |
| | Reporte de estado | Insuficiente | Insuficiente |
| | General del proyecto | Suficiente | Insuficiente |
| | Personal | Insuficiente | Suficiente |
| | Por componentes | No | Suficiente |
| | Por roles | No | Suficiente |
| | Por etapas | No | Suficiente |
| | Controlar los cambios realizados al cronograma | Insuficiente | Insuficiente |
| | Diagramas de Red, de barras, o de hitos | Suficiente | No |

Tabla 1: Comparación de Herramientas

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5 Metodología y herramientas para la modelación de software.

Un paso de vital importancia para un analista a la hora de realizar su trabajo es decidir la metodología que puede usar, sin embargo, no existe una metodología definida para el desarrollo de un producto de software, sino que de acuerdo con las características de un proyecto determinado se aplique la más adecuada. Pueden clasificarse en dos categorías: Metodologías Tradicionales y Metodologías Ágiles. El uso de una metodología brinda muchas ventajas como la mejora de los procesos de desarrollo, de los productos, así como de las relaciones con el cliente. Existen además, herramientas y lenguajes de modelado que se utilizan en el desarrollo de un software de manera que se pueda obtener una breve representación del sistema. A continuación se realizará un estudio sobre estas metodologías y herramientas.

1.5.1 Metodología de Desarrollo

El centro DATEC presenta una metodología basada en la adaptación de metodologías para el desarrollo de software en equipo como son SCRUM y Open Up. Se centra en la organización propuesta para la aplicación del Modelo de Líneas de Productos de Software en DATEC. A continuación se explican brevemente un conjunto de características que presenta Open Up:

Open Up es un proceso de desarrollo iterativo del software que es mínimo, completo y extensible. Presenta las siguientes características:

1. Proceso de desarrollo del software. Es completo en el sentido que puede ser manifestado como todo el proceso para construir un sistema.
2. Es extensible ya que en el proceso se puede agregar o adaptar según lo vayan requiriendo los sistemas. Open Up es un proceso ágil.
3. Es ligero y proporciona una comprensión detallada del proyecto, beneficiando a clientes y desarrolladores sobre productos a entregar y su formalidad.
4. Se centra en una arquitectura temprana para reducir al mínimo los riesgos y organizar el desarrollo.
5. Es la metodología utilizada por desarrolladores de alto nivel en casi todo el mundo por sus altas cualidades administrativas (Flores, 2008).

Además, ofrece la administración de diferentes áreas del proyecto, manejando el ciclo de vida de manera apropiada. Por lo tanto, por políticas del centro se presenta la Metodología para el Desarrollo de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Productos Basada en Modelo de Producción de Líneas de Productos de Software para el análisis de la propuesta.

1.5.2 Lenguaje de Modelado

1.5.2.1 UML (Unified Modeling Language)

Se utilizará por políticas del centro para modelar el sistema el Lenguaje de Modelación Unificado (UML) Versión 2.0, por ser un lenguaje gráfico que se utiliza para especificar, visualizar, construir y documentar los artefactos de un sistema de software. UML es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto (Jacobson, 2000).

Es un lenguaje más expresivo, claro y uniforme para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero sí mejora notablemente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software sin especificar qué metodología o proceso utilizar. UML es independiente, por lo que se puede adaptar a cualquier método logrando perfeccionar el Desarrollo de Software posibilitando el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos (García, 2005).

En el modelo gráfico de UML se identifican:

- ✓ **Elementos:** abstracciones que constituyen los bloques básicos de construcción.
- ✓ **Relaciones:** ligan los elementos.
- ✓ **Diagramas:** es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

1.5.3 Herramienta CASE.

1.5.3.1 Visual Paradigm

Visual Paradigm es una herramienta CASE que permite facilitar el trabajo durante la confección de un software y garantiza el desarrollo de un producto rápidamente. Es colaborativa pues brinda la posibilidad de soportar múltiples usuarios trabajando sobre un mismo proyecto, permitiendo así generar la documentación del proyecto automáticamente en varios formatos y permite el control de versiones (Free Download Manager, 2010).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Entre sus características fundamentales se pueden mencionar las siguientes:

- ✓ Permite realizar los diagramas UML, aunque cada diagrama se restringe a un modelo.
- ✓ Validación en tiempo real del modelo.
- ✓ Posee una interfaz de usuario configurable.
- ✓ Brinda una estructura y organización automática de diagramas.
- ✓ Soporta sub-diagramas para todos los modelos de UML.

Posibilita escribir la especificación de un caso de uso sin necesidad de utilizar una herramienta externa como editor de texto y permite crear Especificaciones de Casos de Uso utilizando plantillas que se encuentran definidas o pueden ser creadas por el mismo usuario (Free Download Manager, 2010).

Por ser una herramienta multiplataforma que se integra con varios IDEs (Ambiente Integrado de Desarrollo), poseer una serie de características fundamentales que contribuyen al desarrollo del software y por políticas del centro se decide utilizar Visual Paradigm como herramienta CASE.

1.6 Ingeniería de Requerimientos

La Ingeniería de Requerimientos trata de los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora, de forma sistemática y repetible (Pressman, 2005).

Existen diferentes definiciones referentes a la Ingeniería de Requerimientos, algunas de las más generales son:

“Ingeniería de Requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo de proyectos” (Dávila, 2001).

“La Ingeniería de Requerimientos es la ciencia y disciplina a la cual le concierne el establecer y documentar los requerimientos” (Dávila, 2001).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

De manera general se resume que la Ingeniería de Requerimientos se encarga de analizar y verificar las necesidades de los clientes para el desarrollo de un sistema. Deben describir lo que se debe hacer y no cómo realizarlo.

Aporta beneficios en aspectos como:

1. Minimiza los riesgos de fracaso del proyecto.
2. Contribuye a cumplir aspectos de calidad, tiempo y presupuesto.
3. Mejora la comunicación entre equipos.
4. Permite gestionar la necesidad del proyecto de forma estructurada.

1.6.1 Características de los requerimientos.

Entre las características fundamentales de los requerimientos se presentan las siguientes:

Deben ser:

1. Especificados por escritos. Como todo contrato o acuerdo entre dos partes.
2. Posibles de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿Cómo sabemos si cumplimos con él o no?
3. Descritos como una característica del sistema a entregar. Esto es lo que el sistema debe hacer (y no cómo debe hacerlo) (IEEE, 1990).

1.6.2 Requisitos funcionales y no funcionales.

¿Qué son los requerimientos?

1. Según la IEEE Standard Glossary of Software Engineering Terminology se define un requerimiento como una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

3. Una representación documentada de una condición o capacidad documentada como las descritas en 1 y 2 (Herrera).

Los requisitos se pueden clasificar en: requisitos funcionales y requisitos no funcionales, estos últimos están divididos en diferentes categorías.

- ✓ **Requisitos funcionales:** son las capacidades o condiciones que el sistema debe cumplir.
- ✓ **Requisitos no funcionales:** son las propiedades o cualidades que el producto debe tener, que no son más que las características que hacen que el producto sea agradable para el usuario, fácil de utilizar y comprender, confiable y rápido. Estos a su vez se dividen en una serie de categorías.

1.6.2.1 Categoría de los requisitos no funcionales

Según el IEEE existen un conjunto de categorías para clasificar los requisitos no funcionales:

- ✓ Requisitos de Software: se debe mencionar el software con el cual se dispone.
- ✓ Requisitos de Hardware: enunciar los elementos de hardware con los cuales se dispone.
- ✓ Restricciones en el diseño y la implementación: especifica o restringe la codificación o construcción de un sistema, son restricciones que son ordenadas y deben cumplirse estrictamente.
- ✓ Requisitos de apariencia o interfaz externa: describe la apariencia del producto. No muestra el diseño de la interfaz en detalle, sino que especifica cómo será la interfaz externa del producto.
- ✓ Requisitos de Seguridad: es el requisito más difícil que puede presentar una serie de riesgos si no se maneja de forma correcta. Este requisito trata tres aspectos fundamentales:
 - Confidencialidad: plantea que la información manejada por el sistema está protegida de algún acceso no autorizado y de su divulgación.
 - Integridad: la información que maneja el sistema será protegida contra la corrupción y estados inconsistentes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Disponibilidad: se le garantizará acceso a la información a los usuarios autorizados y que los mecanismos de seguridad no impedirán que los usuarios accedan a una información determinada en un momento dado.
- ✓ Requisitos de Usabilidad: describen los niveles apropiados de usabilidad, dados los usuarios finales del producto. Se derivan de una combinación de lo que el cliente está tratando de lograr con el producto.
- ✓ Requisitos de Soporte: estos requisitos se relacionan con la instalación, la configuración, la adaptabilidad y la compatibilidad.

1.7 Patrones de Casos de Uso

Un patrón describe una situación que ocurre una y otra vez en un ambiente, para luego plantear la solución de la misma, de manera que esta solución se pueda utilizar más de un millón de veces sin hacerlo dos veces de la misma forma.

Para identificar los casos de uso se utilizan un conjunto de patrones que posibilitan que los requisitos sean mucho más precisos de modo que el trabajo resulte más fácil con el sistema que se va a implementar, permitiendo ganar en tiempo y organización. A continuación se describen brevemente los patrones que se van a utilizar:

➤ Múltiples Actores:

- ✓ Roles Comunes: ocurre cuando dos actores juegan un mismo rol. Este rol lo representa un actor que es heredado por los actores que comparten dicho rol. Es aplicable cuando, desde el punto de vista del caso de uso, sólo exista una entidad externa interactuando con cada una de las instancias del caso de uso.
- ✓ Roles Diferentes: los actores desempeñan roles separados. Es utilizado cuando dos actores juegan roles diferentes en un caso de uso.

➤ CRUD (Crear, Leer, Modificar, Eliminar):

- ✓ Completo: gestiona información en los casos de uso que se quiere crear, visualizar, modificar y eliminar información.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Parcial: modela una de las vías de los casos de uso como un caso de uso separado. Es utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.
- **Extensión o Inclusión Concreta:** está dividido en concreta extensión o concreta inclusión.
 - ✓ Inclusión Concreta: en este patrón existe una inclusión entre el caso de uso base y el caso de uso incluido. El caso de uso base puede ser concreto o abstracto. Se utiliza este patrón cuando un flujo de datos se puede incluir en el flujo de datos de otro caso de uso y también realizarse por sí solo.
 - ✓ Extensión Concreta: consiste en la relación de extensión entre dos casos de uso. El caso de uso extendido es concreto, o sea, una extensión del caso de uso base. El caso de uso base puede ser concreto o abstracto. Es aplicable cuando un flujo de datos puede ser extendido del flujo de datos de otro caso de uso.
- **Concordancia:** extrae una sub-secuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. La sub-secuencia común de casos de uso, extiende los casos de uso compartiendo la sub-secuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la sub-secuencia. (Övergaard, 2004)

1.8 Captura de Requisitos

El adecuado uso de las técnicas de Captura de Requisitos permite que los mismos queden con la calidad esperada por el usuario final, pues muchas veces el cliente no tiene bien definido las funcionalidades que desea que tenga el software a desarrollar, lo que provoca que no sea capaz de explicar con claridad lo que desea. Es por ello que se deben aplicar dichas técnicas, ya que los requisitos constituyen una parte fundamental en el desarrollo del software.

Las Técnicas más habituales en la Elicitación de Requisitos son las Entrevistas, el Desarrollo Conjunto de Aplicaciones (JAD), la Tormenta de Ideas o el Brainstorming y la utilización de escenarios más conocidos como Casos de Uso (Durán T., 2000).

A continuación se describen las técnicas que serán utilizadas abordando sus características fundamentales.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.8.1 Entrevistas

Es la técnica más utilizada, pues es una de las formas de comunicación más natural entre personas. Estructuradas por preguntar cómo y a quién, favoreciendo el contacto directo. Presenta tres fases, las cuales siguen diferentes pasos que se mencionarán a continuación:

Preparación de la entrevista

Se deben tener en cuenta un conjunto de actividades para obtener mejores resultados:

- ✓ Primeramente conocer el dominio del problema para entender las principales necesidades del cliente, logrando una mayor comunicación para con ellos.
- ✓ Seleccionar a las personas que se van a entrevistar, fundamentalmente a los futuros usuarios que serán los que más utilizarán el software.
- ✓ Preparar el contenido de las entrevistas para lograr una entrevista con resultados satisfactorios.
- ✓ Planificar las entrevistas en lugares donde no surjan inconvenientes o interrupciones, de manera que la entrevista quede natural.

Realización de la entrevista

Está compuesta por tres fases:

- Inicio: donde el entrevistador informa al entrevistado la razón de la entrevista, los objetivos que persigue y los tipos de preguntas a realizar.
- Desarrollo: se deben evitar los monólogos y mantener el control por parte del entrevistador. Se pueden aplicar distintas técnicas:
 - ✓ Preguntas abiertas: son más reveladoras, pues a la hora de contestar el entrevistado no tiene límites.
 - ✓ Mostrar interés en todo momento: es fundamental durante la entrevista animar a la persona a hablar repitiendo algunas respuestas dadas, poniendo una postura de atención.
- Terminación: se revisa la entrevista con el objetivo de verificar que no exista error alguno en el contenido de la entrevista.

Análisis de la entrevista

Luego de concluida la entrevista es necesario realizar una lectura detallada y minuciosa de las notas que fueron tomadas y pasarlas en limpio. Una vez elaborada la información se puede enviar al entrevistado para confirmar el contenido.

Aspectos importantes para la realización de una entrevista:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Realizar una preparación de la entrevista.
- ✓ Se recomienda que participen en este proceso dos ingenieros con habilidades de analistas.
- ✓ Realizarlas en lugares con el menor número de interrupciones.

1.8.2 Cuestionarios

Son preguntas presentadas a un grupo de personas para darles respuesta. Se clasifican en dos categorías: abiertas y cerradas. Las primeras son más relevadoras, ya que los encuestados no tienen límites a la hora de dar sus respuestas. Son utilizadas en la etapa exploratoria de la investigación, cuando el analista se propone penetrar en el pensamiento del encuestado. Por otra parte, las preguntas cerradas se utilizan cuando se quiere establecer el criterio de priorización de los casos de uso con el cliente, además se puede utilizar nuevamente a la hora de negociar algún requerimiento con el cliente (Komer, 1993).

1.8.3 Desarrollo Conjunto de Aplicaciones (JAD)

Esta técnica se basa en cuatro principios: dinámica de grupo, el uso de ayuda visual para mejorar la comunicación, mantener el proceso organizado y racional y una filosofía de documentación (Durán T., 2000). Tiene dos grandes pasos, el JAD/Plan que tiene como objetivo elicitar y especificar requisitos, y el JAD/Desing donde se aborda el Diseño de Software (Durán T., 2000). Permite que los usuarios participen con eficacia en los requisitos que modelan. Presenta un conjunto de ventajas con respecto a las entrevistas:

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

Dentro de esta técnica se especifican tres etapas:

- Etapa de Adaptación: esta etapa comienza por definir el proyecto a alto nivel, para lo cual pueden ser necesarias algunas entrevistas a clientes y usuarios. Luego de obtener una idea de los objetivos del proyecto, se seleccionan los participantes para citarlos a reuniones e informarles los principales temas a tratar.
- Etapa de Celebraciones de las sesiones JAD: en esta etapa se exponen las ideas y se discuten para llegar a un acuerdo final.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Etapa de Conclusión: luego de terminar las sesiones se transforman las notas y demás documentación. Se siguen tres pasos fundamentales:
 - ✓ Completar la documentación.
 - ✓ Revisar la documentación.
 - ✓ Validar la documentación.

1.8.4 Tormenta de Ideas o Brainstorming

Su objetivo es la generación de ideas, ayuda a generar varias vistas del problema y a formularlo de diferentes formas. Tiene como ventaja que es fácil de aprender y requiere de poca organización. Su principal objetivo es generar la máxima cantidad de requerimientos para el sistema. Es muy utilizada cuando se necesitan ideas originales acerca del tema de elicitación. Se distinguen las siguientes etapas:

- Etapa de Preparación: se seleccionan los participantes y al jefe de sesión, y luego se prepara la sala donde se realizará la sesión.
- Etapa de Generación: durante esta etapa se definen las siguientes reglas:
 - ✓ Se prohíbe la crítica de ideas con el objetivo de que cualquier participante pueda expresar libremente sus ideas.
 - ✓ Se fomentan las ideas más avanzadas, que aunque no sean las más factibles, incentiva a los demás para crear ideas más creativas.
 - ✓ Es necesario generar todas las ideas posibles, pues cuantas más ideas surjan, existen más posibilidades de obtener mejores ideas.
 - ✓ Se debe alentar a los participantes a combinar o completar las ideas de otros participantes.
- Etapa de Consolidación: se evalúan y organizan las ideas que surgieron en la etapa anterior. Se siguen un conjunto de pasos:
 - ✓ Revisar las ideas.
 - ✓ Descartar ideas.
 - ✓ Priorizar ideas.
 - ✓ Documentación.

1.8.5 Casos de Usos

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Es una técnica bastante utilizada que se encarga de capturar cada función del sistema y sobre la base de cada una de ellas especifica los requisitos. Permiten mostrar el contorno y el alcance de un sistema (Fernández, 2006). Resultan fáciles de entender si van acompañados de visualizaciones gráficas. Los casos de uso permiten realizar una mejor elicitación de requisitos y son de fácil comprensión para los clientes y usuarios.

1.9 Conclusiones

En este capítulo se analizaron de manera general los conceptos que se aplican a la Gestión de Portafolios de Proyectos y a la Gestión de Proyectos, además se describieron los principales procesos que intervienen en la Gestión de Tiempo, así como las herramientas y metodologías que se utilizarán para el posterior análisis de la propuesta. Además, se trataron conceptos relacionados con la Ingeniería de Requerimientos y sus prácticas para el tratamiento de requerimientos. A partir de aquí se comenzará el análisis de la propuesta como tal.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En el presente capítulo se tratarán aspectos relacionados con la modelación del dominio, planteando una propuesta del mismo y de otros artefactos involucrados en el flujo de trabajo de Modelación del Negocio. Se plantean los requisitos funcionales y no funcionales que tendrá la propuesta a desarrollar. Se determinan los casos de uso del sistema y los actores que interactúan con este, elaborándose una descripción detallada de cada uno.

2.1 Propuesta del Sistema

El hecho de no contar con una herramienta que garantice todos los procesos que se llevan a cabo en la Gestión de Tiempo para la realización de la Gestión de Portafolios de Proyectos, hace aún más relevante la construcción de este sistema, ya que va encaminado no sólo a la planificación de un proyecto, sino de un conjunto de proyectos, que proporcionará mejores resultados a la culminación de los mismos.

Primeramente el sistema debe funcionar para todas las instituciones u organizaciones del país que se dedican a la Gestión de Proyectos, incluyendo la Universidad. Tendrán acceso a él los usuarios que tengan un dominio básico de la tecnología y posean los permisos pertinentes.

El desarrollo de este sistema aportará diferentes servicios, entre los que se encuentran adicionar, modificar y mostrar las diferentes actividades que se llevan a cabo en el proyecto. Estas presentan campos asociados que permiten visualizar la información más específica de cada actividad relacionada con el tiempo de duración como: fecha de inicio, fecha de fin, tiempo dedicado a cada actividad y porcentaje dedicado. Permite mostrar los diferentes reportes de las actividades, representando por ejemplo las que no han comenzado, las que ya han sido completadas y las críticas, además muestra las actividades predecesoras y sucesoras, crea dependencias entre las mismas, posibilita adicionar recursos y mostrar información sobre ellos. Permite visualizar un conjunto de diagramas, entre ellos el diagrama de Gantt, donde se representa el camino crítico, además la red del cronograma del proyecto y gráficos relacionados con el estado de los recursos que son asignados a las actividades.

Además, propone introducir un editor de fórmulas que se encarga de realizar todos los cálculos relacionados, por ejemplo, con la duración de la actividad, el por ciento de tiempo dedicado a cada una de estas, la velocidad de iteración, determinar la fecha mínima y máxima para una actividad, así como

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

calcular el esfuerzo horas/hombres. Asimismo para calcular la duración de una actividad se aplican también una serie de técnicas que serán descritas a continuación:

✓ **Estimación Ascendente**

Se estiman las necesidades de los recursos de cada una de las partes más detalladas del trabajo de cada actividad, y estas estimaciones se suman luego en una cantidad total para cada uno de los recursos de la actividad del cronograma. Si existen dependencias entre las actividades del cronograma, este patrón de uso de recursos se refleja en los requisitos estimados de la actividad del cronograma y los documenta. (Institute, 2004)

✓ **Estimación Paramétrica**

La estimación de la base para las duraciones de las actividades puede determinarse cuantitativamente multiplicando la cantidad de trabajo a realizar por el ratio de productividad. Por ejemplo para determinar la duración de una actividad en períodos laborables, las cantidades totales de recursos se multiplican por las horas de trabajo por período laborable o la capacidad de reproducción por período laborable, y se dividen por la cantidad de recursos que se aplican.

✓ **Estimaciones por Tres Valores**

Estas estimaciones se basan en determinar tres tipos de estimaciones:

Más probable: la duración de la actividad del cronograma, teniendo en cuenta los recursos que probablemente serán asignados, su productividad, las expectativas realistas de disponibilidad para la actividad del cronograma, las dependencias de otros participantes las interrupciones.

Optimista: la duración de la actividad se basa en el mejor escenario posible de lo que se describe en la estimación más probable.

Pesimista: la duración de la actividad se basa en el peor escenario posible de lo que se describe en la estimación más probable.

Asimismo se utilizan técnicas para representar el camino crítico del cronograma del proyecto. Un ejemplo es el Método del Camino Crítico descrito a continuación.

✓ **Método del Camino Crítico**

Este método es una técnica para analizar la red del cronograma. Calcula las fechas de inicio y finalizaciones tempranas y tardías teóricas para todas las actividades del cronograma, realizando un

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

análisis de recorrido hacia adelante y hacia atrás a través de los caminos de red del cronograma del proyecto. Estas fechas no son necesariamente el cronograma del proyecto, sin embargo, indican los períodos dentro de los cuales deberían programarse las actividades del cronograma, dadas las duraciones de las actividades, los adelantos y retrasos.

Los caminos críticos tienen una holgura total igual a cero o negativa y las actividades del cronograma en un camino crítico se denominan actividades críticas. Una vez que la holgura total para un camino de red es igual a cero o positiva, se puede determinar la holgura libre, que es la cantidad de tiempo que una actividad del cronograma puede ser demorada sin demorar la fecha de inicio temprana de cualquier actividad sucesora inmediata dentro del camino de red. Este método se puede representar a través del diagrama de Gantt.

En Cuba se lleva a cabo la Gestión de Proyectos, sin embargo no se cuenta con una herramienta que garantice los procesos de la Gestión de Tiempo para realizar la Gestión de Portafolios de Proyectos, que responda a las necesidades de los proyectos que se desarrollan en instituciones u organizaciones del país. De ahí que se proponga la realización de esta herramienta.

2.2 ¿Qué es un Modelo del Dominio?

El Modelo de Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde se encuentra el sistema a construir. Este modelo facilitará un mejor entendimiento para el usuario de los objetivos de la investigación y de los procesos del negocio propuesto y además visualiza los conceptos fundamentales que se tratan en el dominio de la propuesta a desarrollar. La descripción de este modelo se realizó mediante un diagrama de clases UML, donde se identifican las principales clases del dominio.

2.2.1 Glosario de Términos

Como conceptos utilizados en el diagrama se identifican los siguientes:

Proyecto: esfuerzo temporal para obtener un producto o resultado único.

Actividad: es una tarea simple asociada a un proyecto.

Tiempo: se refiere a la duración que tiene una actividad dentro del proyecto.

Campo: definiciones que pertenecen a una categoría de actividad y poseen nombre y valor.

Recurso: fusión de un conjunto de elementos, recursos o insumos que contribuyen al mejor funcionamiento de la organización.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Material: bienes tangibles con que cuenta la empresa para poder ofrecer sus servicios.

Humano: trabajo que aporta el personal de la organización.

Reporte: son informes que se realizan sobre las actividades del proyecto.

Cronograma: diagrama a través del cual se visualizan las diferentes actividades que se realizan en el proyecto.

2.2.2 Diagrama conceptual del Modelo de Dominio

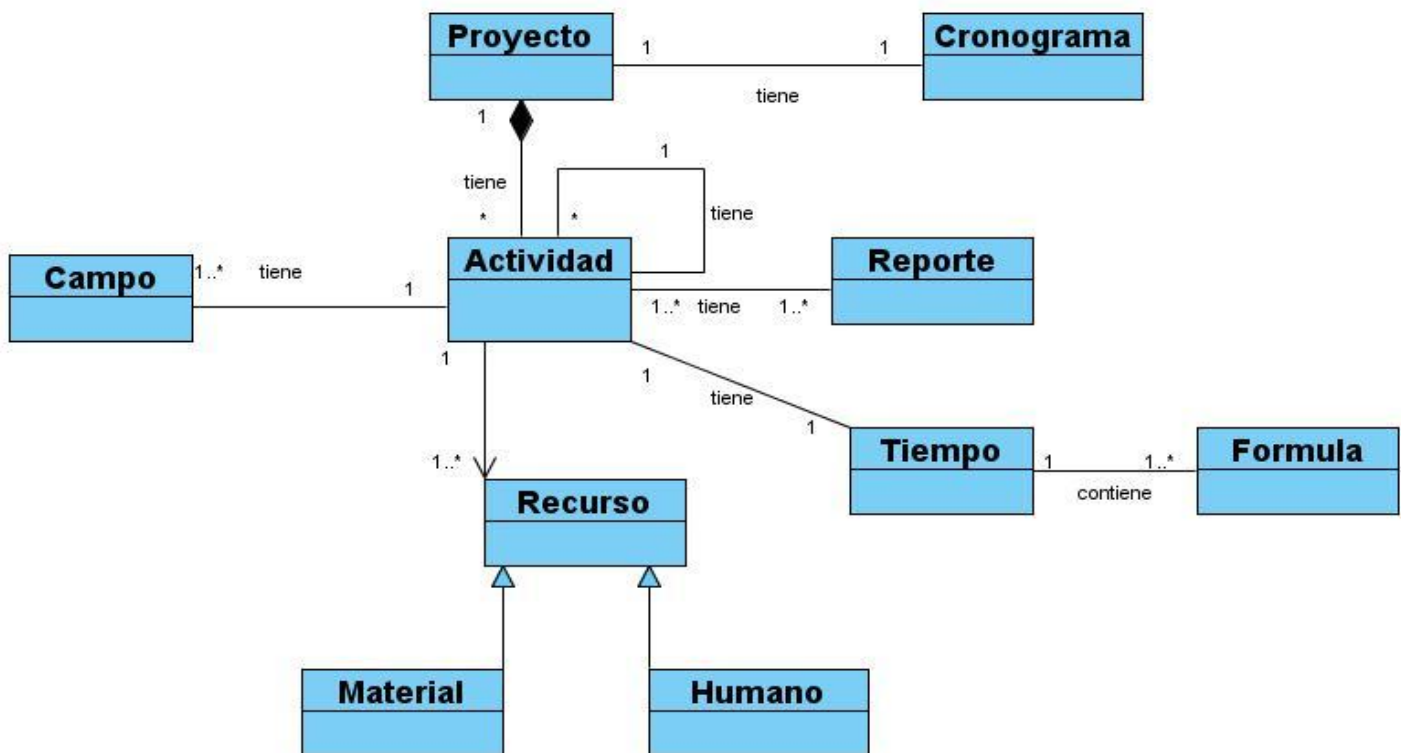


Figura 3: Modelo de Dominio

2.3 Resultados de la aplicación de las Técnicas de Captura de Requisitos

Para lograr una eficiente Captura de Requisitos se aplicaron un conjunto de técnicas, mediante las cuales se obtuvieron resultados de relevancia para este proceso. Es importante destacar que se realizaron dos entrevistas en la empresa Albet a personas especializadas en este tema, las que sirvieron para conocer el grado de aceptación que existe entre los clientes con respecto al sistema que se desea desarrollar. Como

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

resultado de las mismas se definieron 37 requisitos funcionales. Además, se realizó un cuestionario de preguntas a personas que laboran en la Oficina de Gestión de Proyectos, en el cual queda plasmado la importancia de contar con un sistema que integre todos los procesos de la Gestión de Tiempo para el desarrollo de proyectos. Se coordinaron reuniones con los clientes y usuarios del sistema para analizar las funcionalidades que deberá realizar, en las cuales se discutieron un grupo de ideas, aportando otras funciones importantes para el mismo. Como resultado de la Tormenta de Ideas se obtuvo una lista con nuevos requisitos potenciales que podría tener el sistema. Finalmente, para una mejor interpretación y comprensión de los requisitos se representaron a través de diagramas de casos de uso.

2.4 Especificación de los requisitos de software

Como resultado de las técnicas aplicadas se obtuvieron los siguientes requisitos funcionales y no funcionales. Para obtener más información se puede remitir al Expediente de Proyecto.

2.4.1 Requisitos Funcionales

A continuación se presentan los requisitos funcionales:

RF1: El sistema debe ser capaz de realizar la gestión de actividades.

- ✓ **RF1.1:** Adicionar campos de tiempo a una actividad

El sistema debe permitir adicionar una actividad.

- ✓ **RF1.2:** Modificar campos de tiempo de una actividad

El sistema debe permitir actualizar una actividad determinada luego de seleccionarla.

- ✓ **RF1.3:** Mostrar información de tiempo sobre una actividad determinada.

El sistema debe permitir cuando se seleccione una actividad determinada mostrar la información de la misma.

RF2: Adicionar predecesores

El sistema debe permitir adicionar predecesores a una actividad.

RF3: Mostrar información de predecesores de la actividad

El sistema debe permitir que al seleccionar una actividad determinada muestre los predecesores a la actividad determinada.

RF4: Mostrar información de los sucesores de la actividad.

El sistema debe permitir mostrar los sucesores de una actividad determinada.

RF5: Visualizar el diagrama de Gantt

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El sistema debe permitir mostrar el diagrama de Gantt del proyecto.

RF6: Visualizar diagrama de red del cronograma mediante PDM

El sistema debe permitir representar las actividades conectadas a través de flechas para mostrar las dependencias utilizando las relaciones de precedencia final a inicio, final al final, inicio a inicio, inicio a fin.

RF7: Visualizar la estructura de desglose de recursos

El sistema debe permitir mostrar la estructura de desglose de recurso mediante gráficas donde los recursos se identifican por categoría y tipo de recurso.

RF8: Visualizar camino crítico del cronograma

El sistema debe permitir calcular las fechas de inicio y finalizaciones tempranas y tardías para todas las actividades del cronograma, realizando un análisis de recorrido hacia adelante y hacia atrás a través de los caminos de red del cronograma. Estas flechas indican los períodos dentro de los cuales se deben programar las actividades del cronograma.

RF9: Adicionar actividades recurrentes

El sistema debe permitir crear actividades que se repiten en el transcurso del proyecto.

RF10: Mostrar reportes de actividades no comenzadas

El sistema debe permitir mostrar los siguientes datos en los reportes de las actividades que aún no están comenzadas.

RF11: Mostrar reportes de actividades que comienzan pronto

El sistema debe permitir mostrar los datos de los reportes de las actividades que comienzan pronto.

RF12: Mostrar reportes de actividades que están en progreso

El sistema debe permitir mostrar los datos de los reportes de las actividades que están en progreso.

RF13: Mostrar reportes de actividades completadas.

El sistema debe permitir mostrar los datos de los reportes de las actividades completadas.

RF14: Mostrar reportes de las actividades que debían haber comenzado

El sistema debe permitir mostrar los datos de los reportes de las actividades que debían haber comenzado.

RF15: Mostrar reportes de actividades críticas

El sistema debe permitir mostrar los datos de los reportes de las actividades que son críticas.

RF16: Crear dependencias entre actividades

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El sistema debe permitir vincular las actividades definiendo una dependencia entre sus fechas de comienzo y fin.

RF17: Asignar tipos de dependencias a una actividad

El sistema debe permitir asignar tipos de dependencias para definir la secuencia entre las actividades.

RF18: Asignar tiempo de adelanto a una actividad

El sistema debe permitir asignar el tiempo que se puede adelantar una actividad sucesora con respecto a una actividad predecesora.

RF19: Asignar tiempo de posposición a una actividad

El sistema debe permitir asignar el tiempo que se necesita retrasar una actividad.

RF20: Adicionar recurso

El sistema debe permitir adicionar un recurso a una actividad.

RF21: Mostrar información de los recursos de la actividad.

El sistema debe permitir mostrar la información de recursos relacionada con una actividad una vez seleccionada la misma.

RF22: Asignación de recurso a una actividad

El sistema debe permitir asignar un recurso a una actividad determinada.

RF23: Editor de Fórmulas

El sistema debe incluir un editor de fórmulas que haga la edición más fácil de un grupo de fórmulas comunes, permitiendo insertar un grupo de campos predeterminados.

RF24: Calcular la Estimación Ascendente de los recursos asignados a las actividades

El sistema debe permitir estimar las necesidades de los recursos de cada una de las partes detalladas del trabajo que aparece en cada actividad, para luego sumarla en una cantidad total para cada uno de los recursos de la actividad del cronograma.

RF25: Calcular duración de la actividad mediante Estimación Paramétrica.

El sistema debe permitir calcular la duración de una actividad multiplicando las cantidades totales de recursos por las horas de trabajo por período laborable y luego dividirlo por la cantidad de recursos que se aplican.

RF26: Determinar el esfuerzo horas/hombre

El sistema debe permitir estimar la cantidad de esfuerzo de trabajo necesario para completar la actividad del cronograma.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

RF27: Calcular fecha mínima

El sistema debe permitir seleccionar la fecha más antigua dado un grupo de valores Fecha.

RF28: Calcular fecha máxima

El sistema debe permitir seleccionar la fecha más actual dado un grupo de valores Fecha.

RF29: Calcular la holgura libre

El sistema debe permitir calcular la cantidad de tiempo que una actividad del cronograma puede ser demorada sin demorar la fecha de inicio temprana de cualquier actividad sucesora inmediata del camino de red.

RF30: Visualizar el estado actual del cronograma

El sistema debe permitir mostrar la información sobre las fechas de inicio y finalización reales, y las duraciones restantes para las actividades del cronograma que aún no están completadas, además del porcentaje completado de las actividades en curso del cronograma.

RF31: Mostrar avance del cronograma

El sistema debe mostrar el avance del cronograma mediante gráficas.

RF32: Calcular estado de hitos

El sistema debe permitir calcular qué parte de los hitos se han cumplido respecto a los planificados.

RF33: Calcular cumplimiento del cronograma

El sistema debe permitir calcular qué parte del cronograma se ha cumplido.

RF34: Calcular la desviación de línea base

El sistema debe permitir calcular cuánto se ha desviado el proyecto de su plan inicial.

RF35: Calcular % de tiempo dedicado

El sistema debe permitir calcular el tiempo que se dedica a cada actividad del proyecto.

RF36: Calcular trabajo terminado

El sistema debe permitir calcular qué parte del trabajo se ha terminado según el plan.

RF37: Calcular velocidad de iteración

El sistema debe permitir calcular el avance de una iteración.

2.4.2 Requisitos No Funcionales

Seguidamente se presentan los requisitos no funcionales:

1. Usabilidad

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

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

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.

2. Fiabilidad

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

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

3. Eficiencia

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

RNF7: Los tiempos de respuesta de la aplicación no deben exceder de los 900ms \pm 400ms 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.

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

RNF9: El sistema debe contar con una ayuda, que permita encontrar las funcionalidades fácilmente.

RNF10: El sistema debe contar con un manual de usuario, que permita conocer las características y las formas de funcionamiento del software.

5. Interfaz

RNF11: 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.

6. Hardware

RNF12: Se requiere de un servidor de 2 GB de RAM como mínimo y 4 GB de espacio libre en disco duro.

RNF13: 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.

7. Software

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

RNF14: El sistema debe ser desarrollado en Lenguaje de script PHP 5 o superior.

RNF15: Para la instalación del servidor:

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

RNF16: Para interpretación por el cliente:

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

8. Disponibilidad

RNF17: El sistema deberá estar disponible 24x7.

9. Restricciones de diseño

RNF18: El software contará con las siguientes restricciones:

- El lenguaje de programación a utilizar es el PHP.
- 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.

10. Interfaces de Comunicación

RNF19: El Sistema se comunicará con los gestores de bases de datos mediante protocolo TCP/IP, mientras que los clientes accederán al sistema vía HTTP/HTTPS.

11. Soporte

RNF20: 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.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.5 Definición de los casos de uso

Los casos de uso son artefactos que describen cómo se comporta el sistema desde el punto de vista del usuario, por lo que establece las condiciones y posibilidades que deberá cumplir el sistema a desarrollar (Jacobson, y otros, 2000).

2.5.1 Definición de los actores del sistema

A continuación se muestra la siguiente tabla donde se presentan los actores que quedaron definidos como resultados de las técnicas aplicadas en la Captura de Requisitos.

| Actor | Descripción |
|--------------|--|
| Gerente | El actor "Gerente" es un usuario del sistema que tiene los permisos necesarios para acceder a la funcionalidad de mostrar reportes relacionados con las actividades. |
| Usuario | Es el encargado de la creación, modificación y eliminación de las acciones específicas y de sus actividades. |
| Planificador | Encargado de acceder a los niveles del sistema que le permita realizar funciones básicas y generales. |

Tabla 2: Actores del sistema

2.5.2 Diagrama de Paquetes

Los casos de uso organizados mediante paquetes permiten mantener un mayor orden a la hora de consultar la documentación obtenida en esta etapa. En este caso se agrupan los casos de uso por paquetes para lograr una mejor comprensión y organización de los mismos. El siguiente diagrama presenta los paquetes que fueron definidos para cada caso de uso respectivamente.

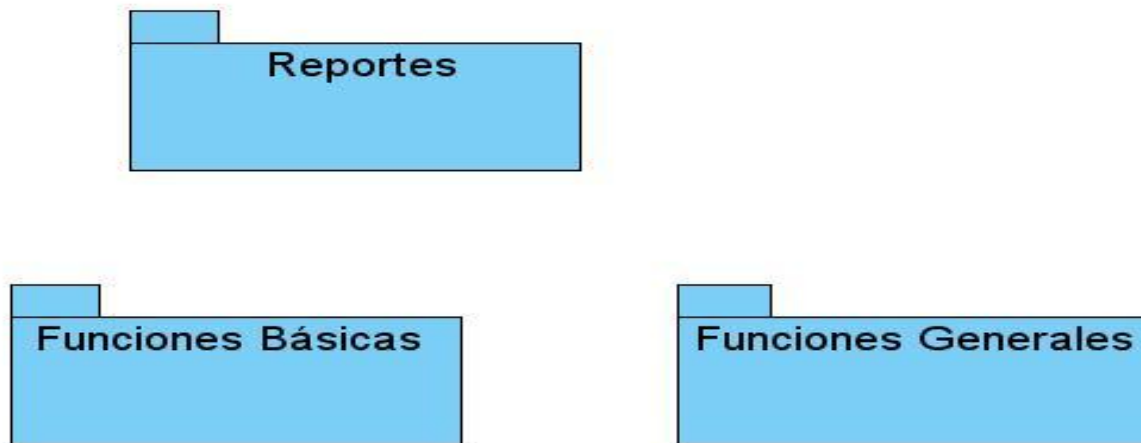


Figura 4: Diagrama de Paquetes

2.5.3 Diagramas de Casos de Uso

El diagrama de casos de uso del sistema representa fragmentos de funcionalidad del sistema, describiendo las acciones que realiza un actor a la hora de interactuar con el mismo. A continuación se presentan los casos de uso asociados a cada paquete.



Figura 5: Diagrama de Casos de uso del Paquete de Reportes

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

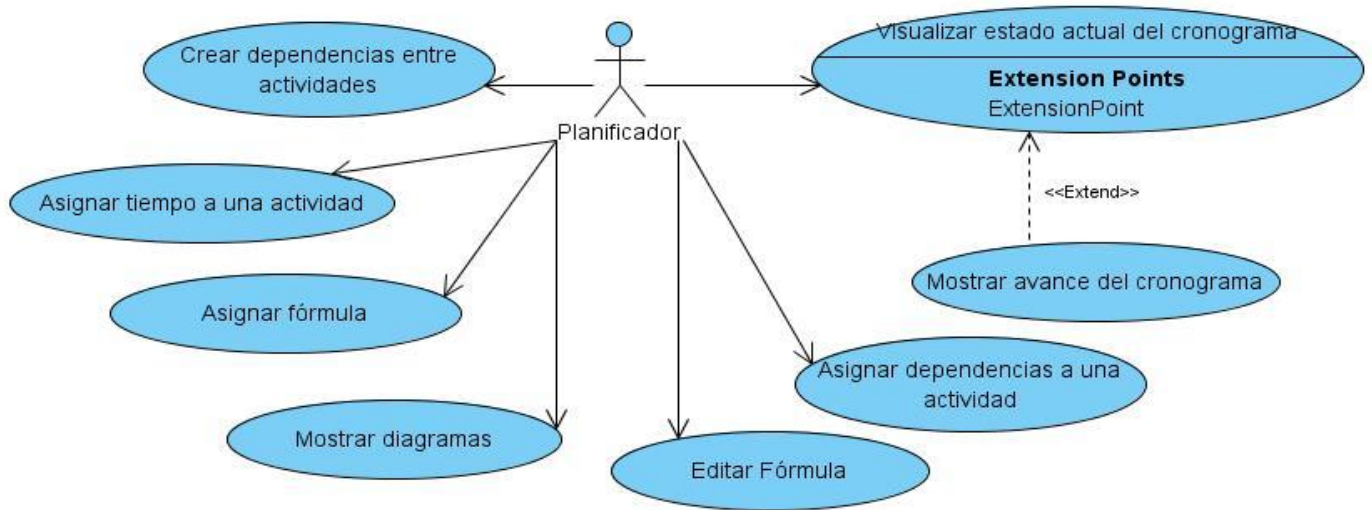


Figura 6: Diagrama de Casos de uso del Paquete de Funciones Básicas

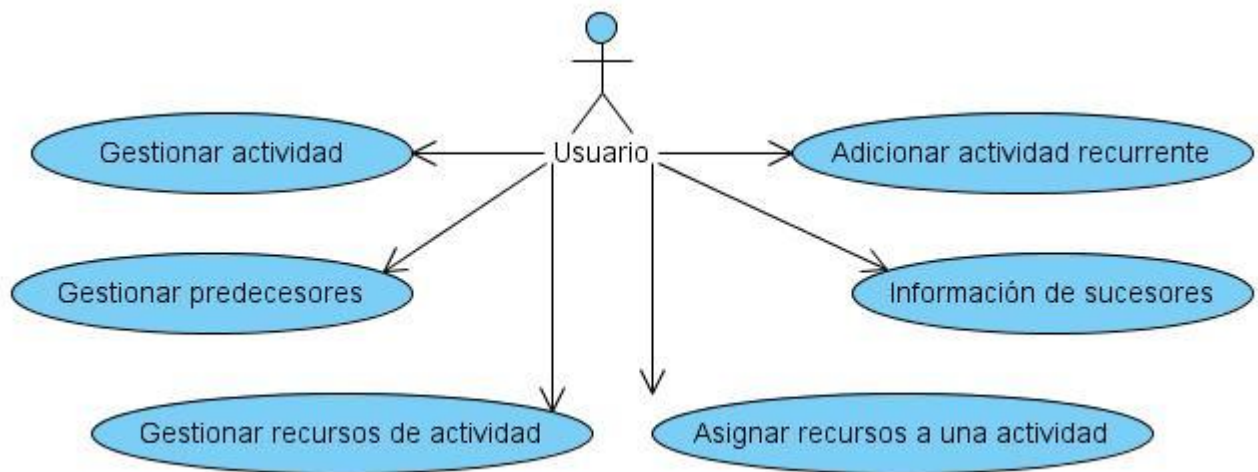


Figura 7: Diagrama de Casos de uso del Paquete de Funciones Generales

2.5.4 Descripción de los casos de uso del sistema

Según los resultados obtenidos de aplicar las técnicas para la Validación de Requisitos, se acordó que los casos de uso descritos a continuación son los más importantes para el sistema, porque cubren las principales tareas y funciones que el mismo debe realizar. El resto de los casos de uso se pueden

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

encontrar en el Expediente de Proyecto. Para facilitar su comprensión se incluyó la descripción de los casos de uso en los anexos. **(Ver Anexo 3)**

2.6 Conclusiones

En este capítulo se realizó el modelo de dominio y se determinó una lista de las funcionalidades que debe tener el sistema a desarrollar. Como consecuencia, se obtuvieron 15 casos de uso, los cuales fueron agrupados en forma de paquetes para facilitar una mejor comprensión y organización de los mismos. Se identificaron, además, los actores que serán los encargados de interactuar con el sistema y se describieron de manera detallada los casos de uso. Partiendo de este punto se puede comenzar el desarrollo del sistema, teniendo en cuenta los requisitos y funcionalidades analizados en el presente capítulo.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Introducción

En este capítulo se realiza un análisis con el objetivo de evaluar los resultados de los requisitos obtenidos en el capítulo anterior. Se emplean técnicas y métricas de validación para evaluar la calidad del trabajo realizado.

3.1 Técnicas de Validación de Requisitos

La validación de requisitos permite comprobar que los requisitos que quedaron definidos en el sistema se corresponden con los que realmente necesita el cliente y el usuario, verifica que no existan omisiones, así como que sean completos y consistentes. De ella depende que los costos para el mantenimiento del software que se desarrolla no sean tan elevados.

Las técnicas a utilizar se describen a continuación:

- ✓ **Reviews o Walk throughs:** es la revisión y corrección de los requisitos que fueron definidos.
- ✓ **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.
- ✓ **Matriz de Trazabilidad:** esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario verificar qué objetivos cubre cada requisito, de manera que se puedan detectar inconsistencias u objetivos no cubiertos. (Durán A., (1999))
- ✓ **Casos de Pruebas:** consiste en definir las entradas, salidas del software y acciones del usuario para que se complete lo que expresa el requisito.

3.2 Pasos para la Validación de Requisitos

Para lograr que el sistema a desarrollar cumpla con lo establecido por el cliente se hace necesario llevar a cabo un análisis de los resultados obtenidos, con el propósito de conocer si lo que se está realizando es correcto.

Existen tres pasos fundamentales que se realizan para la validación:

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

- ✓ Revisión Técnica por los Analistas Principales: su propósito es verificar que se hayan generado correctamente los artefactos correspondientes de la Ingeniería de Requisitos.
- ✓ Revisión Funcional: su objetivo es validar que las funcionalidades que fueron descritas cumplan con las expectativas de los interesados y se hayan descrito correctamente.
- ✓ Taller de Aprobación: tiene como objetivo presentar las soluciones a las consideraciones detectadas en los pasos anteriores y aprobar la especificación de requisitos según los criterios de evaluación y aceptación de los requisitos.

3.3 Métricas para la Validación de Requisitos

Existen un conjunto de métricas que se utilizan para validar los requisitos definidos de manera que cumplan con las necesidades del cliente y tengan la calidad requerida. Así mismo se encuentran métricas para la validación de los casos de uso, las cuales permiten verificar de manera eficiente que los mismos están descritos correctamente y cumplan con las funcionalidades del sistema.

La aplicación de métricas de la calidad de software constituye un factor importante para los resultados finales con respecto a la calidad, lo que trae consigo resultados satisfactorios para el usuario final. La medición permite tener una visión más concisa proporcionando un mecanismo para la evaluación objetiva. Sin embargo, existen una serie de criterios a tener en cuenta para la evaluación y aceptación de los requisitos que son necesarios para la Especificación de Requisitos:

| Atributo | Descripción | Cumplimiento (%) |
|-------------|--|--------------------------------------|
| Correcta | La ERS es correcta si contiene todos los requisitos que el software debe satisfacer. | El 99% ERS son correctas. |
| No Ambigua | La ERS es no ambigua si cada requisito tiene una única interpretación. | El 97 % de las ERS son no ambiguas. |
| Completa | La ERS es completa si están incluidas todas las funciones requeridas por el cliente. | El 100% de las ERS son completas. |
| Consistente | La ERS es consistente sí ningún subconjunto de la misma entra en contradicción con otro subconjunto. | El 100% de las ERS son consistentes. |
| Clara | La ERS es clara si está escrita en un lenguaje apropiado. | El 98% de las ERS son claras. |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Tabla 3: Características que deben cumplir los requisitos

3.4 Estrategia para la Validación

En el presente trabajo la validación de los resultados obtenidos se realizará a través de la aplicación del procedimiento descrito en el trabajo de diploma “Procedimiento para la validación de requisitos de software”, el cual consta de tres fases. A continuación se presenta cada fase, donde se hace referencia a sus objetivos fundamentales, las tareas a realizar y las técnicas que se van a utilizar para validar los diferentes artefactos.

3.4.1 Fase 1 Validación del listado de requisitos

Esta fase tiene como objetivo fundamental evaluar los requisitos que fueron identificados, de manera que se puedan detectar errores que puedan ser analizados y corregidos a tiempo. Entre sus tareas principales se encuentran verificar:

- ✓ Que los requisitos no sean ambiguos.
- ✓ Que los requisitos sean especificados por escrito.
- ✓ Que sean posibles de probar o verificar.
- ✓ Que sean descrito como una característica del sistema.
- ✓ Que sean lo más abstracto y conciso posible.

Las técnicas que se aplicarán en esta fase son las Revisiones y la Lista de Chequeo.

3.4.1.1 Revisiones

Para validar el listado de requisitos se aplicó la técnica conocida como Revisiones, con el objetivo de verificar que todos los requisitos estén especificados en un documento tangible para los clientes, que sean posibles de probar o verificar, así como, que estén escritos como una característica del sistema. Para lograr dicho objetivo se realizaron tres revisiones con los resultados siguientes:

Revisión 1: en esta primera revisión se encontró que de los 37 requisitos levantados, se habían especificado 35 en el documento, lo que representa un 94,6%. Asimismo, se hallaron 31 requisitos que fueron posibles de probar, es decir, que aseguraban ser una funcionalidad capaz de satisfacer las necesidades del sistema, lo que significa un 83,7%. Se encontraron además, 33 requisitos que describen con claridad lo que debe hacer el sistema, para un resultado de un 89,1%. Finalmente, se detectaron 32 requisitos que estaban descritos detalladamente, lo que figura un 86,4%.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Revisión 2: en la segunda revisión se detectaron menos problemas. De los 37 requisitos levantados se especificaron los 37 en el documento, para un resultado de un 100%. Se detectó que eran posibles de probar 35 de los requisitos, lo que representa un 94,5%. Asimismo se describieron de forma concisa 36 requisitos, lo que figura un 97,2%. Igualmente se describieron detalladamente 34 requisitos, para un 91,8%.

Revisión 3: luego de analizar con detalles las revisiones anteriores se encontraron menos dificultades en los requisitos: fueron posibles de probar 36 requisitos, para un resultado de un 97,2%. Además, se lograron describir de forma concisa todos los requisitos, para un 100%. Por último, se describieron detalladamente 35 requisitos, lo que representa un 94,5%.

Después de analizar las tres revisiones expuestas anteriormente se llegó a la conclusión de que el listado de requisitos tiene la calidad requerida, pues están descritos detalladamente, son posibles de probar o verificar, se describen como características del sistema y están todos contenidos en el documento Especificación de Requisitos. A continuación se muestra una gráfica que muestra el resultado de las tres revisiones que fueron realizadas.

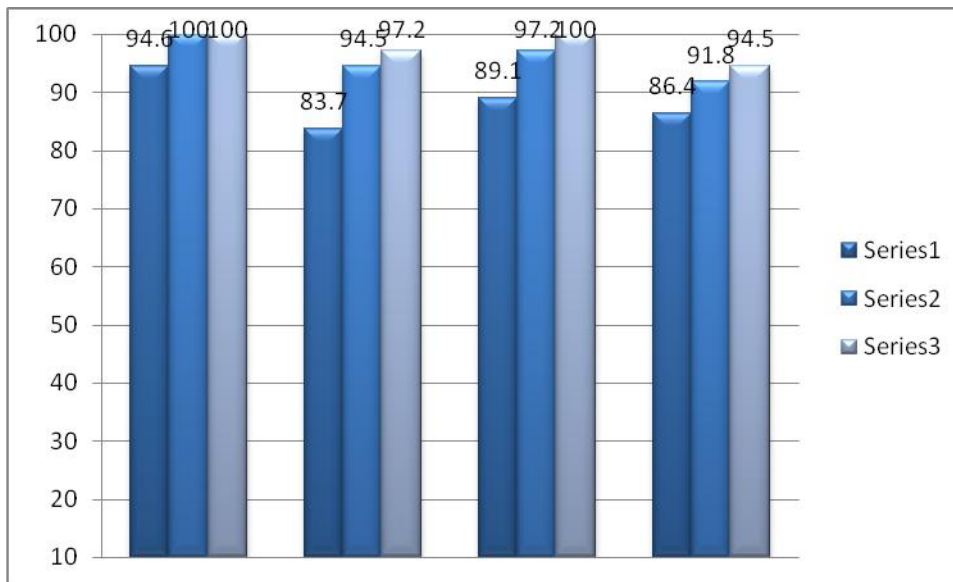


Figura 8: Resultados de las Revisiones

3.4.1.2 Lista de Chequeo para validar el listado de requisitos

A continuación se muestra la lista de chequeo definida por Calidad UCI aplicada al listado de los requisitos con el objetivo de verificar que estén correctos y no presenten inconsistencias.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| Indicadores a evaluar | Nivel de importancia | Procedimiento | Respuesta |
|--|----------------------|-----------------------|-----------|
| 1. ¿Se refleja el seguimiento del documento? | B | Revisión de Documento | SI |
| 2. ¿Se especifica el número de cada requisito evaluado? | A | Revisión de Documento | SI |
| 3. ¿Se especifica el nombre de cada Requisito Solicitado? | A | Revisión de Documento | SI |
| 4. ¿Se especifica si están identificados los elementos de entrada de cada requisito? | A | Revisión de Documento | SI |
| 5. ¿Se especifica si están identificados los elementos de salida de cada requisito? | A | Revisión de Documento | SI |
| 6. ¿Se especifica si cada requisito no es ambiguo? | A | Revisión de Documento | SI |
| 7. ¿Se especifica si cada requisito está correcto? | A | Revisión de Documento | SI |
| 8. ¿El requisito puede ser verificado? | A | Revisión de Documento | SI |

Tabla 4: Lista de verificación del listado de requisitos

La aplicación de esta técnica permitió validar el listado de requisitos, demostrando que los mismos tienen la calidad esperada, ya que satisfacen las necesidades del cliente. Están descritos correctamente, pues cada uno tiene identificado los elementos de entradas y salidas; además pueden ser probados o verificados y no poseen ningún nivel de ambigüedad.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

3.4.2 Fase 2 Validación del documento Especificación de Requisitos

Esta fase consiste en la validación del documento Especificación de Requisitos, con el propósito de verificar que los requisitos describan de manera correcta y en un lenguaje comprensible lo que debe realizar el sistema. Entre sus tareas principales está verificar:

- ✓ Que la ERS sea correcta.
- ✓ Que la ERS no sea ambigua.
- ✓ Que la ERS sea completa.
- ✓ Que la ERS sea consistente.

Se utilizarán para esta fase las métricas para la Calidad de la Especificación de Requisitos y la Lista de Chequeo.

3.4.2.1 Métricas para Calidad de la Especificación de Requisitos

Los requisitos definidos fueron revisados a través de la métrica para la calidad de especificación de los requisitos, con el propósito de analizar la ambigüedad de los mismos. La métrica es el resultado de dividir el número de requisitos para los cuales los revisores tuvieron interpretaciones idénticas (**Nu1**) entre el total de requisitos en una especificación (**NR**).

$$Q1 = Nu1 / NR$$

$$NR = NF + NNF$$

Donde **NF** es el número de Requisitos Funcionales y **NNF** el número de Requisitos No Funcionales.

La interpretación del resultado de esta métrica representa que cuanto más cerca de 1 esté el valor de **Q1** menor será la ambigüedad de los requisitos.

Para la aplicación de esta métrica se realizaron tres iteraciones con el objetivo de especificar claramente las necesidades del cliente. A continuación se presentan los resultados obtenidos en estas iteraciones.

Iteración 1: Durante esta iteración se detectó que cuatro requisitos no estaban definidos con la claridad suficiente. En el caso de los requisitos que fueron capturados en este trabajo los revisores, de los 57 requisitos identificados tuvieron interpretaciones parecidas para 54 de ellos. A continuación se plantea la ecuación con los siguientes valores:

$$NR = 37 + 20$$

$$NR = 57$$

$$Q1 = Nu1 / NR$$

$$Q1 = 54 / 57$$

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

$$Q1 = 0.94$$

Iteración 2: En esta iteración todavía los revisores estaban en desacuerdo con uno de los requisitos que anteriormente tuvo problemas. Por lo que para un total de 57 requisitos, los revisores tuvieron interpretaciones idénticas para 56 de ellos. A continuación se muestran los resultados:

$$Q1 = \text{Nu1} / \text{NR}$$

$$Q1 = 56 / 57$$

$$Q1 = 0.98$$

Iteración 3: Finalmente en esta iteración para un total de 57 requisitos los revisores tuvieron interpretaciones idénticas para los 57.

$$Q1 = \text{Nu1} / \text{NR}$$

$$Q1 = 57 / 57$$

$$Q1 = 1$$

Luego de analizar las tres iteraciones se llegó a la conclusión de que los requisitos no presentan ningún nivel de ambigüedad, ya que en cada iteración el valor de Q1 se acerca a 1, lo que indica un resultado satisfactorio.

No obstante existen una serie de aspectos con los cuales también deben cumplir los requisitos para demostrar la calidad esperada por el cliente. Para que la Especificación de Requisitos sea reconocida de manera satisfactoria debe cumplirse que:

- ✓ La correctitud sea menor que 1,10.
- ✓ La completitud sea mayor que 0,90.
- ✓ La consistencia sea menor que 0,20.

Correctitud

Plantea que la Especificación de Requisitos es correcta si contiene todos los requisitos que el sistema debe satisfacer.

$$X = \text{TO} / \text{TR}$$

Donde TO será la cantidad de observaciones realizadas en la validación, TR el total de requisitos revisados y X la correctitud.

$$X = 2/57$$

$$X = 0.03$$

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

Compleitud

Determina que la Especificación de Requisitos está completa.

$$X = 1 - OF / TS$$

Donde OF será el total de observaciones de formato, TS el total de secciones del documento y X la completitud.

$$X = 1 - (0/6)$$

$$X = 1$$

Consistencia

Esta métrica determina si la Especificación de Requisitos es consistente.

$$X = TO / TR$$

Donde TO será el total de observaciones de consistencia, TR el total de requisitos revisados y X la consistencia.

$$X = 1/57$$

$$X = 0.01$$

Luego de haber aplicado estas métricas a los requisitos definidos se llegó a la conclusión de que cumplen con los valores correspondientes a la Correctitud, Compleitud y Consistencia, por lo que la Especificación de Requisitos puede ser aceptada, pues tiene la calidad esperada por el cliente.

3.4.2.2 Lista de Chequeo para validar del documento ERS

La lista que se muestra a continuación está definida por Calidad UCI, la cual se aplica con el objetivo de validar el documento de Especificación de Requisitos.

| Estructura del Documento | | |
|---|------------|---------------------------------|
| Aspectos a evaluar | Evaluación | Cantidad de elementos afectados |
| 1. ¿Está el documento acorde con la plantilla estándar del proyecto o del expediente de proyecto? | B | Ninguno |
| Semántica del Documento | | |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| Aspectos a evaluar | Evaluación | Cantidad de elementos afectados |
|---|------------|---------------------------------|
| 1. ¿Ha identificado errores ortográficos? | B | Ninguno |
| 2. ¿Se entiende claramente lo que se ha especificado en el documento? | B | Ninguno |
| 3. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento? | B | Ninguno |
| Elementos definidos por la metodología | | |
| Aspectos a evaluar | Evaluación | Cantidad de elementos afectados |
| 1. ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro? | R | 2 elementos afectados |
| 2. ¿Debería especificarse algún requisito con más detalle? | R | 4 elementos afectados |
| 3. ¿Han sido abordados e identificados los valores de entradas y salidas? | R | 5 elementos afectados |
| 4. ¿Se han identificado los requerimientos de software y de hardware? | B | Ninguno |
| 5. ¿Han sido identificadas las restricciones de diseño e implementación? | B | Ninguno |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| | | |
|---|---|-----------------------|
| 6. ¿No aparece un mismo requisito en más de un lugar del documento de especificación? | R | 3 elementos afectados |
|---|---|-----------------------|

Tabla 5: Lista de chequeo para validar el documento ERS

3.4.3 Fase 3 Validación del documento Modelo de Casos de Uso del Sistema

El objetivo de esta fase se centra en corregir los casos de uso, haciendo énfasis en su descripción, de manera que se pueda lograr que los requisitos se vean reflejados en cada uno de ellos. Entre sus tareas se encuentra verificar:

- ✓ Que cada requisito esté asociado a un caso de uso.
- ✓ La completitud de los casos de uso.
- ✓ La correctitud de los casos de uso.
- ✓ La consistencia de los casos de uso.
- ✓ La complejidad de los casos de uso.
- ✓ Que cada caso de uso se vea reflejado en un prototipo de interfaz.

Para aplicar esta fase se utilizarán las métricas para la validación del documento Modelo de Casos de Uso y la técnica Matriz de Trazabilidad.

3.4.3.1 Métricas para la validación del Modelo de Casos de Uso del Sistema

A continuación se aplican un conjunto de métricas a los casos de uso con el objetivo de medir la calidad de los mismos.

| Factor Completitud | Métrica Asociada | Sistema |
|--|--|---|
| Factor 7 ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso? | <p>Métrica 7: Número de requisitos omitidos por caso de uso.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Revisar la lista de requisitos para ver cuáles serán apoyados por cada caso de uso.</p> | <p>Total de casos de uso: 15</p> <p>Número de requisitos omitidos por caso de uso: 0</p> <p>Representa: 0%</p> |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| | | |
|---|--|--|
| <p>Factor 8 ¿Existen requisitos que no han sido considerados en algún caso de uso?</p> | <p>Métrica 9: Número de requisitos que no son considerados en ningún caso de uso.</p> | <p>Total de RF: 37</p> <p>Número de RF que no son considerados en ningún caso de uso: 0</p> <p>Representa: 0%</p> |
| <p>Factor 13 ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia?</p> | <p>Métrica 15: Número de casos de uso que no han sido clasificados.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Hacer reuniones con el usuario para analizar y priorizar los requisitos de acuerdo a su relevancia.</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso sin clasificar: 0</p> <p>Representa: 0%</p> |
| | | 100% |
| Factor Correctitud | Métrica Asociada | Sistema |
| <p>Factor 22 ¿Existe para cada caso de uso por lo menos un usuario responsable?</p> | <p>Métrica 26: Número de casos de uso que no tienen un usuario responsable.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Analice la responsabilidad que representa el caso de uso y acuerde con los usuarios cuál es el responsable directo a éste.</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso que no tienen usuario responsable: 0</p> <p>Representa: 0%</p> |
| <p>Factor 23 ¿Representa el caso de uso requisitos comprensibles para el usuario?</p> | <p>Métrica 28: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario.</p> <p>Umbral < 5%</p> <p>Acción sugerida: Discuta con el usuario la interacción que describe el</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 2</p> <p>Representa: 2,7%</p> |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| | | |
|--|---|---|
| | caso de uso y ajuste dicha descripción de manera que sea comprensible por el usuario. | |
| Factor 25 ¿Las interacciones definidas describen la funcionalidad requerida del sistema? | <p>Métrica 31: Número de casos de uso que deben ser modificados para adecuarlo a la funcionalidad del sistema.</p> <p>Umbral < 10%</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso que deben ser modificados para adecuarlo a la funcionalidad del sistema: 0.</p> <p>Representa: 0%</p> |
| | | 97,3% |
| Factor Consistencia | Métrica Asociada | Sistema |
| Factor 18 ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos? | <p>Métrica 22: Número de casos de uso no aceptados.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Realice reuniones con el usuario para mejorar la descripción de los casos de uso.</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso no aceptados: 4</p> <p>Representa: 5.3%</p> |
| Factor 21 ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados? | <p>Métrica 25: Números de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Estructure el caso de uso de manera que separe su funcionalidad básica de la funcionalidad repetitiva o alternativa.</p> | <p>Total de Casos de Uso: 15</p> <p>Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 1</p> <p>Representa: 1,3%</p> |
| | | 93,6% |
| Factor Complejidad | Métrica Asociada | Sistema |
| Factor 29 ¿Los | Métrica 36: Número de elementos | Total de Casos de Uso: 15 |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| | | |
|--|---|--|
| <p>elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p> | <p>del diagrama que requieren reubicación.</p> <p>Umbral < 30%</p> <p>Acción sugerida: Modifique la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible.</p> | <p>Número de elementos del diagrama que requieren reubicación: 0</p> <p>Representa: 0%</p> |
| | | <p>100%</p> |

Tabla 6: Factores vinculados a las métricas de calidad

Luego de aplicar las métricas anteriores para validar el modelo de Casos de Uso del Sistema se llegó a la conclusión de que los casos de uso están descritos correctamente mostrando todos los flujos correspondientes de manera que el usuario obtenga una mejor comprensión de los mismos. Además, cumple con los factores Completitud, Correctitud, Consistencia y Complejidad, demostrando que cada requisito identificado está asociado a cada caso de uso. Finalmente, se puede decir que el artefacto tiene la calidad requerida para comenzar con el diseño del sistema. A continuación se muestra una gráfica donde se representan los resultados por cada factor.

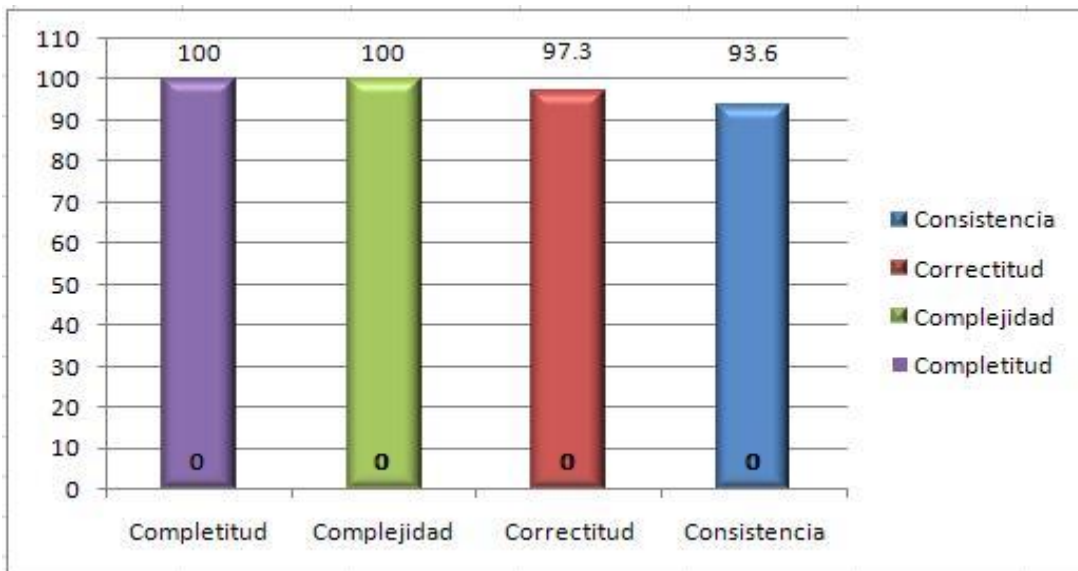


Figura 9: Valores de cada factor aplicados al Modelo de Casos de Uso

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

3.4.3.2 Matriz de Trazabilidad

Una técnica de validación muy importante y que también fue aplicada en este trabajo fue la matriz de trazabilidad. La misma tiene como propósito verificar que no existan inconsistencias a la hora de incluir los requisitos en los casos de uso. Esto se representa a través de una matriz, en la cual se muestran los requisitos funcionales de forma vertical y los casos de uso horizontal, de manera que se correspondan unos con otros. A continuación se muestra la matriz de trazabilidad de los requisitos de software.

| CU \ RF | CU1 | CU2 | CU3 | CU4 | CU5 | CU6 | CU7 | CU8 | CU9 | CU10 | CU11 | CU12 | CU13 | CU14 | CU15 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|
| 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 | |

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

| | | | | | | | | | | | | | | | | | |
|------|--|--|--|--|--|--|--|--|--|--|--|--|---|---|--|---|--|
| RF29 | | | | | | | | | | | | | | | | X | |
| RF30 | | | | | | | | | | | | | X | | | | |
| RF31 | | | | | | | | | | | | | | X | | | |
| RF32 | | | | | | | | | | | | | | | | X | |
| RF33 | | | | | | | | | | | | | | | | X | |
| RF34 | | | | | | | | | | | | | | | | X | |
| RF35 | | | | | | | | | | | | | | | | X | |
| RF36 | | | | | | | | | | | | | | | | X | |
| RF37 | | | | | | | | | | | | | | | | X | |

Tabla 7: Matriz de Trazabilidad

Como se evidencia en la matriz de trazabilidad cada requisito de software está incluido en al menos un caso de uso.

3.5 Casos de Prueba

El objetivo fundamental de esta técnica es verificar que lo que realiza el sistema realmente está correcto. En este proceso se incluyen las entradas, los resultados esperados y las condiciones bajo las cuales se debe probar.

3.5.1 Pruebas del Sistema

Las pruebas del sistema tienen como objetivo principal comprobar que lo que se está desarrollando cumple con las funcionalidades descritas en los casos de uso. Se comienzan a realizar una vez definidos los requisitos funcionales, lo que permite encontrar errores u omisiones cuando todavía se pueden corregir. Para facilitar la comprensión de los casos de pruebas más importantes según lo acordado entre el cliente y el analista, se decidió incluirlos en los anexos. **(Ver Anexo 4)**

3.6 Conclusiones

Un paso importante para verificar que los requisitos tienen la calidad esperada por el cliente es la corrección de los mismos. De ahí, la importancia de detectar los errores a tiempo, ya que si no efectúa una eficiente validación, estos errores continuarán para las fases siguientes, provocando costos elevados y pérdida de tiempo. Las métricas utilizadas permitieron comprobar que los requisitos que fueron definidos en el capítulo anterior, son los que realmente desea el cliente y poseen la calidad requerida para su aprobación.

CONCLUSIONES

CONCLUSIONES

Como resultado de la investigación vinculada al presente trabajo de diploma se llegó a las siguientes conclusiones:

- ✓ Para realizar el análisis del módulo Gestión de Tiempo, se realizó un estudio de los diferentes sistemas informáticos sobre los cuales se basa la propuesta a desarrollar. Concluyendo que los más completos fueron Microsoft Enterprise Project Management, Redmine y Project-Open, aunque no incluyen todos los procesos de la Gestión de Tiempo necesarios para garantizar una eficiente Gestión de Portafolios de Proyectos.
- ✓ Para el levantamiento de requisitos se aplicaron técnicas de Captura de Requisitos, ya que constituyen un factor importante para que el trabajo quede con la calidad esperada. La aplicación de estas técnicas permitió identificar 37 funcionalidades, de ellas 11 son las más significativas para el sistema a desarrollar.
- ✓ Estas funcionalidades fueron especificadas en el documento Especificación de Requisitos de Software, lo que permitió realizar el Modelo del Sistema donde se visualizaron los diferentes subsistemas de manera que el usuario pudiera conocer mejor el funcionamiento del sistema.
- ✓ Se diseñaron los prototipos de interfaz del sistema a desarrollar, permitiendo su posterior implementación.
- ✓ Se seleccionaron y aplicaron una serie de métricas para validar los artefactos generados durante el proceso de análisis que garantizaron la calidad de los mismos.

Finalmente, se concluye que los objetivos propuestos inicialmente para este trabajo han sido cumplidos de manera satisfactoria.

RECOMENDACIONES

RECOMENDACIONES

Al final de este trabajo se hacen algunas recomendaciones para obtener resultados satisfactorios en las siguientes fases de trabajo:

- ✓ Continuar con las investigaciones para lograr un mayor acercamiento a las necesidades de los clientes y añadir nuevas funcionalidades al sistema.
- ✓ Continuar con el desarrollo del módulo y su integración a la plataforma.
- ✓ Aplicar la propuesta en todos los proyectos productivos de la Universidad y el país.
- ✓ Utilizar el presente trabajo como bibliografía para posteriores investigaciones.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Andriole, Stephen J. 1996. *Managing Systems Requirements: Methods, Tools, and Cases.* McGraw Hill. 1996.

Braude, Eric J. 2003. *Ingeniería de Software: Una perspectiva orientada a objetos.* s.l. . s.l. : Editorial Ra-Ma, 2003., 2003.

Bredemeyer, Ruth Malan and Dana. 2001. *Defining Non-Functional Requirements.* 2001.

Business Management. 2010. Business Management . [En línea] 2010. [Citado el: 10 de febrero de 2010.] Disponible en: <http://www.busmanagement.com//article/Open-Source--the-Next-Wave-for-Project-and-Portfolio-Management-Applications/>

Capítulo 5: Herramientas CASE.

2010. Ciencia y Técnica Administrativa. [En línea] 2010. [Citado el: 14 de febrero de 2010.]

Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>

Dávila, N. D. 2001. *"Ingeniería de Requerimientos. Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. "* . 2001.

[En línea] [Citado el: 10 de enero de 2010.] Disponible en: <http://webs.montevideo.com.uy/nicolasd>

Durán T., A., Bernárdez J. y Beatriz. 2000. *"Metodología para la Elicitación de Requisitos de Software del Sistema".* 2000.

Durán T., A., Bernárdez J. y Beatriz. 2000. *"Metodología para la Elicitación de Requisitos de Software del Sistema".* 2000.

Durán A., Bernárdez, B., Ruiz, A., Toro M. (1999). *A Requirements Elicitation Approach Based in Templates and Patterns.* Buenos Aires, Argentina : s.n., (1999).

Dorfman, Richard H. Thayer and Merlin. 1997. *Software Requirements Engineering, 2nd Edition.* IEEE Computer Society Press. 1997.

Fernández., Ing. Leidy Sánchez. 2006. *"Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR)".* 2006.

Fernández Lugo, Jorge Gerardo. 2009. *Estudio Comparativo de Herramientas de Software para la Administración de Proyectos.* 2009.

Free Download Manager. 2010. Free Download Manager. [En línea] 2010. [Citado el: 15 de enero de 2010.] Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/

BIBLIOGRAFÍA

- 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 Computación, 2009. 2009.
- Geidys Patrón Blanco, Ailec Rodríguez Iglesia. 2009.** *Procedimiento para la validación de requisitos de software*. 2009.
- Grady, Robert. 1992.** *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall : s.n., 1992. 1992.
- Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®) Tercera Edición. . 2004.
- Herrera, L. J.** *"Ingeniería de Requerimientos, Ingeniería de Software"*.
Monografias.com S.A. 2010. Monografias.com. [Online] 2010. [Cited: enero 10, 2010.]
Disponible en: <http://www.monografias.com/trabajos6/resof/resof.shtml>
- Holtzblatt, Hugh Beyer and Karen. 1998.** *Contextual Design*. San Francisco : s.n., 1998.
- IBM. 2010.** IBM. [En línea] 2010. [Citado el: 10 de febrero de 2010.]
Disponible en: <http://www-01.ibm.com/software/rational/products/rpc/index.html>
- ICB- IPMA Competence Baseline, Versión 3.0. 2006.
- 2002.** . *"Ingeniería del Software. Un enfoque práctico"* . 2002
- . **2006.** *"Ingeniería del Software: Un enfoque práctico"*, Sexta Edición. México DF : McGraw Hill., 2006.
- IEEE. 1990.** *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Standard Institute of Electrical and Electronics Engineers. 1990.
- Institute, Project Management. 2004.** *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®) Tercera Edición*. 2004.
- IPMA. 2006..** *International Project Management Association (IPMA)*. 2006. 2006.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Lenguaje Unificado de Modelado*. . España : s.n., 2000.
Kioskea. [Online] [Cited: febrero 12, 2010.]
Disponible en: <http://es.kioskea.net/genie-logiciel/methodes-agiles.php3>
- Jacobson, Ivar y Booch, Grady, Rumbaugh, James. 2000.** *"El Proceso Unificado de Desarrollo de Software"*. 2000.
- Komer, P. 1993.** *Dirección de la Mercadotenia*. España, Prentice Hall. : s.n., 1993.

BIBLIOGRAFÍA

- Kovitz, Benjamin L. 1999.** *Practical Software Requirements-A Manual of Content & Style.* 1999.
- Larman, C. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Mexico : s.n., 1999.
- María José Escalona, Nora Koch. 2009.** *Ingeniería de Requisitos en Aplicaciones para la Web- Un estudio comparativo.* 2009.
- Microsoft Corporation. 2010.** Office Online. [En línea] 2010. [Citado el: 10 de febrero de 2010.] Disponible en: <http://office.microsoft.com/es-es/epmsolution/HA101656443082.aspx>
- Moreno, Yasmiany Marrero.** *La Oficina de Gestión de Proyectos, una propuesta para administrar proyectos de software.*
- Núñez, Rolando Kindelán. 2009..** *Servicio de Gestión de Proyectos.* La Habana : s.n., 2009.
- Oracle. 2010.** Oracle. [En línea] 2010. [Citado el: 10 de febrero de 2010.] Disponible en: <http://www.oracle.com/global/es/products/applications/project-management.html>
- Övergaard, Gunnar y Palmkvist, Karin. 2004.** *Use Cases Patterns and Blueprints.* 2004.
- Paulo Nunes. 2008.** Knowow. [En línea] 1 de enero de 2008. [Citado el: 16 de febrero de 2010.] Disponible en: <http://www.knoow.net/es/cieeconcom/gestionde proyectos.htm>
- Pressman, Roger S. 1997.** *“Ingeniería del Software: Un enfoque práctico. Cuarta Edición.* Madrid : s.n., 1997.
- 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 . 2002.*
- Pressman, Roger S. 2005.** *Ingeniería de Software. Un Enfoque Practico.* 2005.
- Pressman, Roger S. 2006.** *“Ingeniería del Software: Un enfoque práctico”, Sexta Edición.* México DF : McGraw Hill., 2006.
- Project.net. 2010.** Project.net. [En línea] 2010. [Citado el: 10 de febrero de 2010.] Disponible en: <http://www.project.net/about>
- Richard Thayer, Merlin Dorfam. 2000.** *Software Requeriments Engineering.* California : s.n., 2000.
- Roman Zamitiz, Ing. Carlos Alberto.** *El Lenguaje Unificado de Modelado.*
- Universidad Nacional Autónoma de México. 2010.** División de Ingeniería Eléctrica. [En línea] 2010. [Citado el: 15 de febrero de 2010.] Disponible en: <http://www.fi-b.unam.mx/pp/profesores/carlos/aydoo/uml.html>
- Ronald Saborío, Omar Quesada. 2006.** *Gestión del Portafolio de Proyectos, Alineando el Negocio.* 2006.

BIBLIOGRAFÍA

Sawyer, Ian Sommerville and Pete. 1997. *Requirements Engineering-A Good Practice Guide.* New York, NY: John Wiley & Sons, Inc. 1997.

Spence, Kurt Bittner and Ian. 2003. *Use Case Modeling.* Addison Wesley Longman. 2003.

The Standard for Portfolio Management. 2006.

Wesly, Addison. 2004. *Use Case: Patterns and Blueprints.* OVERGAARD, Gunnar : PALKVIST, 2004. 2004.

Widrig, Dean Leffingwell and Don. 2000. *Managing Software Requirements – a Unified Approach.* Addison Wesley Longman. 2000.

GLOSARIO DE TÉRMINOS

GLOSARIO DE TÉRMINOS

Actividad: es una tarea simple asociada a un proyecto.

Acta de Constitución del Proyecto: el acta de constitución del proyecto es el documento que autoriza formalmente un proyecto.

Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Captura de Requisitos: proceso durante el cual se identifica un problema y se especifica los requisitos que debe cumplir un producto de software.

Caso de Uso: secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Cronograma: diagrama través del cual se visualizan las diferentes actividades que se realizan en un proyecto.

Campo: definiciones que pertenecen a una categoría de actividad y poseen nombre y valor.

Establecimiento de la Secuencia de Actividades: implica identificar y documentar las relaciones lógicas entre las actividades del cronograma.

Estimación de los Recursos de las Actividades: involucra determinar cuáles son los recursos y qué cantidad se utilizará de cada uno, así como estarán disponibles para realizar las actividades del proyecto.

Estructura de Desglose de Recursos: estructura jerárquica de los recursos identificados por categoría y tipo.

Entrevista: acción y efecto de entrevistar o entrevistarse. Vista, concurrencia y conferencia de dos o más.

Especificación de Requisitos: documento que describe lo que hace un sistema de software: sus funciones y atributos. Generalmente escritas desde el punto de vista del usuario.

Gestión de Tiempo: la Gestión de Tiempo del Proyecto incluye los procesos necesarios para lograr la conclusión del proyecto a tiempo.

Gestión de Proyectos: consiste en la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos y alcanzar sus objetivos.

GLOSARIO DE TÉRMINOS

Gestión de Portafolios de Proyectos: se refiere a la administración centralizada de uno o más portafolios e incluye identificación, priorización, autorización, gestión y control de proyectos, programas y otros trabajos relacionados para alcanzar determinados objetivos estratégicos del negocio.

Humano: trabajo que aporta el personal de la organización.

Ingeniería de Requerimientos: se encarga de analizar y verificar las necesidades de los clientes para el desarrollo de un sistema. Deben describir lo que se debe hacer y no cómo realizarlo.

Método de Diagramación por Precedencia (PDM): es un método para crear un diagrama de red del cronograma del proyecto que utiliza casillas o rectángulos, denominados nodos, para representar actividades.

Método de Diagramación con Flechas (ADM): es un método para crear un diagrama de red del cronograma de proyecto que utiliza flechas para representar las actividades.

Material: bienes tangibles con que cuenta la empresa para poder ofrecer sus servicios.

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

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

Proceso: se define como un conjunto de actividades enlazadas entre sí que partiendo de una o más entradas las transforman y genera un resultado.

Requisitos: una condición o capacidad para un usuario para resolver un problema o alcanzar un objetivo.

Requisitos Funcionales: los requisitos funcionales van a ser las capacidades o condiciones que deberá tener el sistema a construir.

Requisitos no Funcionales: los requisitos no funcionales son las cualidades o propiedades que deberá tener el producto.

Tiempo: magnitud física que permite ordenar la secuencia de los sucesos, estableciendo un pasado, un presente y un futuro.

UML: lenguaje de modelado visual consistente en el cual se expresan los resultados de numerosas metodologías de orientación a objetos existentes.