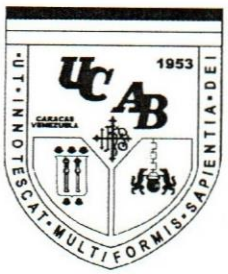
AAR8714

TRAB
ITC2010
P3.



UNIVERSIDAD CATÓLICA ANDRÉS BELLO

CONSEJO DE FACULTAD DE INGENIERÍA

TRABAJO DE ASCENSO

SISTEMA DE GESTIÓN DE EQUIPOS AUDIOVISUALES (PLAN DE EJECUCIÓN UCAB BASADO EN EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE)

Presentado por:

Pazo Marín, Jacinto Rafael

optar Prof. Asistente

TESIS
ITC2010
P3

Caracas, Abril de 2010

Universidad Católica Andrés Bello
Vicerrectorado Académico
Dirección General de los Estudios de Postgrado
Área de Ciencias Administrativas y de Gestión
Postgrado en Gerencia de Proyectos

SISTEMA DE GESTIÓN DE EQUIPOS AUDIOVISUALES (PLAN DE EJECUCIÓN UCAB BASADO EN EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE)

Autor: Jacinto Rafael Pazo Marín

Asesor: Gustavo Henrique Roa Eugenio

Resumen

El objetivo principal de este trabajo de investigación fue aplicar ciertas técnicas para transformar requisitos de un usuario en un Plan de Ejecución para un Sistema de Gestión de Equipos Audiovisuales, a través de la construcción de artefactos o modelos que describe una visión del sistema desde un punto de vista específico y de manera gráfica y sencilla, enfocados bajo el paradigma del Proceso Unificado de Desarrollo de Software (RUP, por sus siglas en inglés). Todas estas herramientas o modelos fueron empleados (los que aplican) e integrados en los 42 procesos que están agrupados lógicamente según la Guía de Fundamentos para la Gestión de Proyectos (PMBOK, 2008, por sus siglas en inglés) del PMI[1]. El trabajo de investigación se realizó bajo el enfoque de proyecto factible apoyado en un diseño bibliográfico, ya que se buscó generar una solución práctica para la Dirección de Tecnologías de la Información de la Universidad Católica Andrés Bello, por lo que se creó un modelo práctico de un Plan de Ejecución que servirá como guía de futuros proyectos que incluye procesos y herramientas, tanto de la Gerencia de Proyectos como del Proceso Unificado de Desarrollo de Software.

Palabras claves: PMI, PMBOK, RUP, UML[2], Proceso Unificado, Lenguaje de Modelado, Gerencia de Proyectos, Desarrollo de Software

[1] Project Management Institute, Instituto de Gerencia de Proyectos por sus siglas en inglés.
[2] Unified Modeling Language, Lenguaje Unificado de Modelado por sus siglas en inglés.

Lista de Abreviaturas, Acrónimos, Siglas y Símbolos

BIOUCAB: Sistema de Intranet de la Universidad Católica Andrés Bello.

CASE: Computer Aided Software Engineering.

CIS: Computer and Information Science Deparment at University of Michigan.

COCOMO: Constructive, Cost, Model .

DTI: Dirección de Tecnologías de Información.

EDT: Estructura Desagregada del Trabajo.

GP: Gerencia de Proyecto.

HTML: HyperText Markup Language.

KLCD: Miles de líneas de códigos.

LAMP: Acrónimo de Linux, Apache, MySQL, PHP.

LOC/FP: Lines Of Code, Function Point.

MCU: Modelo de Casos de Uso.

OMG: Object Management Group.

PMBOK : Project Management Body of Knowledge.

PMI: Project Management Institute.

RUP: Rational Unified Process .

UCAB: Universidad Católica Andrés Bello.

UPEL: Universidad Pedagógica Experimental Libertador.

UML: Unified Modeling Language.

W3C: World Wide Web Consortium.

Índice de Contenido

Lista de Abreviaturas, Acrónimos, Siglas y Símbolos iii

Índice de Contenido iv

Listado de Cuadros y Figuras vii

Introducción 1

Capítulo 1 – El Problema 3

Contexto 3

Planteamiento del problema 4

Justificación 5

Objetivos de la Investigación 6

Alcance 7

Limitaciones 7

Capítulo 2 – Marco Teórico 8

Marco Organizacional 8

Fundamentos Teóricos de la Gerencia de Proyectos 9

Plan de Ejecución del Proyecto 19

Fundamentos Teóricos de Desarrollo de Software 20

El Proceso Unificado de Desarrollo de Software (RUP). 20

Dirigido por los Casos de Uso. 21

Centrado en la Arquitectura 22

Es un proceso Iterativo e Incremental 22

Fases del Proceso Unificado 24

¿Qué es un modelo? 25

Lenguaje Unificado de Modelado (UML) 26

El Modelo de Estimación COCOMO. 27

Técnicas de Estimación de Costos del Software 28

Capítulo 3. Marco Metodológico 30

Tipo de Investigación 30

Procedimiento de la investigación 31

Unidad de Análisis 33

Técnicas de recolección de datos 33

Capítulo 4 – Desarrollo 35

Manejo del Alcance 35

Descripción contextual de los principales Casos de Uso. 43

Manejo de los costos. 52

Manejo del tiempo 56

Manejo de la calidad 57

Manejo de los Riesgos 58

Manejo de las Comunicaciones 60

Manejo de la Integración 61

Manejo del Recurso Humano. 62

Manejo de las Adquisiciones 62

Capítulo – 5. Conclusiones y Recomendaciones 64

Conclusiones 64

Recomendaciones 65

Referencias Bibliográficas 66

Listado de Cuadros y Figuras

Ilustración 1. Descripción general de las áreas de conocimiento en la GP..... 10

Ilustración 2. Flujos de trabajos tienen lugar sobre las cuatro fases 25

Ilustración 3. Modelos que implementa UML 2.0 27

Ilustración 4. Modelo de negocio Sistema de Gestión de Equipos Audiovisuales 37

Ilustración 5. Diagrama de Jerarquía de Consultar Agenda..... 37

Ilustración 6. Diagrama de Jerarquía de Reservar Equipos 38

Ilustración 7. Diagrama de Jerarquía de Entrega de Equipos 38

Ilustración 8. Modelo de Casos de Uso del Negocio 42

Ilustración 9. Estructura Desagregada del Trabajo (EDT)..... 51

Ilustración 10. Indicadores de la fase 1 del modelo COCOMO 53

Ilustración 11. Pesos de los factores de complejidad, Modelo COCOMO..... 54

Ilustración 12. Cálculo del factor LOC/FP del modelo COCOMO 55

Ilustración 13. Cálculo de la complejidad del desarrollo..... 55

Ilustración 14. Cronograma de desarrollo de las actividades..... 57

Ilustración 15. Lista de Riegos..... 59

Introducción

Hoy en día el gerenciar o administrar proyectos de Desarrollo de Software constituye una disciplina de complejidad moderada, dado que la mayoría de los proyectos de esta naturaleza sufren modificaciones durante su ciclo de vida. Es por ello que existen diversas metodologías que ayudan a aplicar ciertos paradigmas que contribuyen a la correcta aplicación de todos los procesos involucrados en el Desarrollo del Software.

Sin embargo, estas metodologías sólo se enfocan en los procesos propios de la creación de los componentes y artefactos necesarios que constituyen el software a crear, por lo cual carecen de ciertos fundamentos necesarios a la hora de emprender cualquier proyecto de esta naturaleza.

Por otro lado, existe un Organismo Internacional para la Gerencia de Proyectos (PMI – Project Management Institute, por sus siglas en inglés), el cual se ha encargado de especializar la Gerencia de Proyecto como una disciplina bien estructurada la cual cuenta con un conjunto de publicaciones (PMBOK, 2008) – entre otras cosas – sobre el estudio de los procesos más efectivos a aplicarse en el Desarrollo de Proyectos en general, todo esto producto de muchos años de investigación en el área.

Sin embargo, ambas disciplinas son complementarias, ya que una se enfoca en los procesos para la creación de un software y la otra en la administración de los procesos en el desarrollo de un proyecto. Es por ello que este trabajo de investigación se centra en la aplicación de ambas disciplinas con el fin de

crear un Plan de Ejecución para la elaboración de un Sistema de Manejos de Audiovisuales.

El presente trabajo de investigación se esquematiza de la siguiente manera:

El Capítulo 1 define el problema de la investigación, se centra en el contexto que da origen al presente trabajo de investigación detallando el planteamiento del problema, la justificación del estudio, los objetivos de la investigación, así como el alcance y las limitaciones de la misma.

El Capítulo 2 expone el marco teórico y se compone del marco organizacional, los antecedentes de la investigación y los fundamentos teóricos que sustentaron el estudio.

El Capítulo 3 comprende el marco metodológico donde se diseñó el plan de la investigación con la metodología empleada, el tipo y diseño que se utilizó en la investigación, la población de estudio así como las técnicas empleadas para la recolección de la información.

El Capítulo 4 comprende el desarrollo, donde se detallan los procesos que componen el Plan de Ejecución involucrados en la creación de un software para la gestión de equipos audiovisuales.

Finalmente el Capítulo 5 expone las conclusiones generales y las recomendaciones para futuros proyectos.

Capítulo 1 – El Problema

Este capítulo conforma el punto de partida de esta investigación. Se especifica el contexto general que da pie a la formulación del problema, los objetivos así como el alcance y las limitaciones del trabajo de investigación.

Contexto

La Dirección de Tecnologías de Información (DTI) es un centro de investigaciones y servicios basado en el desarrollo de nuevas alternativas en el uso de la Tecnología Informática. Su labor es dar apoyo tanto académico como técnico a la comunidad de la Universidad Católica Andrés Bello (UCAB), con el fin de proyectarla corno una institución de vanguardia en el ámbito tecnológico, para lo cual constantemente se amplía la infraestructura debido al crecimiento natural de la institución así como la aparición de nuevas tecnologías que sirven de apoyo directa o indirectamente a la labor de la docencia.

En la actualidad la universidad cuenta con numerosos equipos audiovisuales que se encuentran asignados a las distintas dependencias o escuelas para ser utilizados en calidad de préstamos rotativos por parte de la comunidad para distintos tipos de uso. Cada dependencia controla y administra sus equipos, ya sea por medio de un proceso manual a través de una agenda o por medio algún sistema de información que preste apoyo a este tipo de labor. Este proceso, sea manual o automatizado, es totalmente descentralizado, lo que deriva en un uso poco eficiente de los recursos que se poseen, ya que algunas dependencias tienen muy alta demanda y no son suficientes sus recursos

para prestar el servicio a sus usuarios y por otro lado, algunas tienen muy poca demanda y subutilizan el recurso, viéndose afectado el usuario de la comunidad.

Este proceso descentralizado no permite hacer un seguimiento al mantenimiento preventivo y correctivo que requieren los equipos por parte del DTI, no genera reportes oportunos del uso de los equipos para apoyar la toma de decisiones según demanda y uso (por ejemplo) y además, no permite asegurar un resguardo seguro y efectivo de los equipos para minimizar las pérdidas de estos, entre otras.

Todo esto supone la necesidad de crear un proceso de carácter centralizado que garantice la gestión adecuada de este tipo de recursos, tanto en administración de uso, mantenimiento, ubicuidad y resguardo.

Planteamiento del problema

Es conveniente para apoyar este proceso que se elabore un Sistema de Gestión de Equipos Audiovisuales que permita controlar el adecuado manejo de los equipos por parte de los distintos usuarios que utilizan o intervienen en este proceso.

Para evitar los errores típicos en que incumben los equipos de Desarrollo de Software se quiere establecer una metodología de desarrollo rigurosa que permita tener el control de la planificación. Para ello se quiere adoptar un estándar que permita mantener una comunicación fluida tanto con los miembros del equipo de desarrollo así como los "stakeholders" o involucrados.

Es conocido que a nivel de los procesos de gestión de proyectos el PMI cuenta con una Guía de Fundamentos para la Gestión de proyectos (PMBOK, por sus siglas en inglés), que contiene una descripción de los procesos que deben tomarse en cuenta a

la hora de planificar un proyecto y las áreas de conocimientos específicas requeridas para aplicar esos proceso, sin embargo, muchos de ellos no indican cómo deberían enfocarse a la hora de emprender un proyecto de Desarrollo de Software. Así mismo, en el área de Desarrollo de Software, existan también una serie de procesos que se ejecutan desde el inicio hasta su terminación agrupados en forma de Metodologías de Desarrollo, que guían el proceso de desarrollo, tal es el caso del Proceso Unificado de Desarrollo de Software, por nombrar alguno.

Ambos enfoques tienen sus ventajas y desventajas al ser aplicados por lo que se desea crear un Plan de Ejecución piloto de desarrollo que guíe la correcta ejecución de las etapas y fases del ciclo de vida de un proyecto de software, usando las fortalezas de las dos disciplinas buscando complementarlas, como se indicó al inicio.

Justificación

Desde el punto de vista logístico se hace necesario adoptar un proceso estándar de manejo de los equipos audiovisuales que permita por un lado, prestar un servicio más eficiente hacia el usuario, un mejor aprovechamiento y mantenimiento de los recursos, un mejor resguardo, dada la cantidad de pérdida de los equipos así como un mejor manejo de la información. Todo esto conlleva indirectamente a obtener un beneficio económico al poder contar con los recursos necesarios según la demanda estimada que puede ser proyectada con un sistema de información centralizado y uso eficiente de los recursos disponibles en toda la universidad.

Desde el punto de vista práctico en el campo laboral la unidad de desarrollo del DTI ejecuta anualmente cerca de 10 a 15 proyectos de distintos alcances y dada la tendencia de que cada vez es más común que el Desarrollo de Sistemas de Software sea mucho más grandes y más complejos, y por ende su gestión mucho más complicada, es necesario tratar de estandarizar la gestión global de los proyectos de la Unidad adaptados a una metodología de desarrollo estándar, que guíe el proceso, que permitirá adoptar un esquema de trabajo más organizado, basándose en la creación de un Plan de Ejecución de Desarrollo de Software que arroje como resultado un conjunto de distintos modelos e instrumentos para ser aplicados a futuros proyectos.

Desde el punto de vista académico, sería de gran utilidad poder contar con un Plan Piloto que sirva como base para futuras investigaciones sobre la Gerencia de Proyectos aplicada a Desarrollo de Software.

Objetivos de la Investigación

Objetivo General

Desarrollar Plan de Ejecución para un Sistema de Gestión de Equipos Audiovisuales de la UCAB basados en el Proceso Unificado de Desarrollo de Software.

Objetivos Específicos

Evaluar el Proceso Unificado de Desarrollo de Software.

Definir los distintos modelos de la fase de iniciación que definan el desarrollo de un Sistema de Gestión de equipos audiovisuales.

Determinar los elementos que componen un Plan de Ejecución basado en la "Guía de los Fundamentos para la Dirección de Proyectos, PMBOK (2008)".

Alcance

En cuanto a la Gestión del Proyecto, el presente trabajo de investigación abarca la mayoría de los procesos de las diferentes áreas de conocimiento delimitados dentro del grupo de procesos de iniciación y planificación definidos por el PMBOK (2008).

En cuanto al desarrollo del software el presente trabajo sólo se enfoca en la primera iteración de la fase de inicio y en la fase de elaboración, la cual se compone de ciertos diagramas y modelos definidos por el Proceso Unificado de Desarrollo de Software.

Limitaciones

Dado que el Desarrollo de Software comprende muchos tipos de tecnologías y arquitecturas, se limitó la presente investigación para Desarrollo de Software orientado a objetos en el área de sistemas de información orientados a la Web específicamente.

Dado que el proyecto tiene una aplicación dentro de una estructura organizativa muy ajustada y poco flexible, ciertos procesos no podrán ser definidos debido a la limitante de acceso a la información de algunas unidades involucradas.

Capítulo 2 – Marco Teórico

Este capítulo expone las bases teóricas que sirven de apoyo para el estudio y desarrollo del Plan de Ejecución propuesto.

Marco Organizacional

El DTI se funda a finales de 1993 con motivo del cuadragésimo aniversario de la UCAB con el propósito de promover y divulgar el uso de herramientas computacionales y su aplicación como soporte a los diversos niveles de docencia e investigación en la universidad.

Su misión es "prestar un servicio rápido, eficaz, actualizado, profesional y de bajo costo, en el desarrollo y mantenimiento de la infraestructura tecnológica de la Universidad Católica Andrés Bello, a través de: asesoría académica, servicios de comunicaciones, adquisición y mantenimiento de equipos y software, elaboración y entrenamiento en la aplicación de proyectos educativos para la comunidad Ucabista".[3]

El DTI cuenta con una estructura organizativa compuesta de unidades funcionales, dentro de la cual existe la unidad de desarrollo que se encarga de la creación y mantenimiento de aplicaciones, que responden a diferentes necesidades dentro y fuera de la comunidad Ucabista. Desde 2002 esta unidad se ha especializado en el desarrollo de sistemas de software basados en el área Web para dar apoyo a los procesos de gestión administrativa a través de la creación de sistemas adaptados a la Intranet de la UCAB.

---

[3] *Disponible en: http://www.ucab.edu.ve/dti.html. Consultado el 27-10-2009*

Fundamentos Teóricos de la Gerencia de Proyectos

Gerencia de Proyectos

La Guía de los Fundamentos para la Dirección de Proyecto (PMBOK, 2008) define a la dirección de proyectos como "la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuadas de los 42 procesos de la dirección de proyectos, agrupados lógicamente" (Project Management Institute Inc, 2008).

Estos procesos se agrupan en 5 grandes grupos y a su vez en 9 áreas de conocimiento específicas que deben conocer un gerente de proyecto.

En cada área de conocimiento específica se deben tener ciertas habilidades para aplicar herramientas y técnicas que ayudan a que los procesos se ejecuten de forma exitosa. Es por ello, que se define un plan de manejo o gestión de cada área en específica.

La Ilustración 1 presenta un esquema general de las áreas de conocimiento en la gerencia de proyectos y los procesos que se ejecutan dentro de cada una.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Execution	4.4 Monitor and Control Project Work 4.5 Perform Integrated Change Control	4.6 Close Project or Phase
5. Project Scope Management		5.1 Collect Requirements 5.2 Define Scope 5.3 Create WBS		5.4 Verify Scope 5.5 Control Scope	
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Estimate Costs 7.2 Determine Budget		7.3 Control Costs	
8. Project Quality Management		8.1 Plan Quality	8.2 Perform Quality Assurance	8.3 Perform Quality Control	
9. Project Human Resource Management		9.1 Develop Human Resource Plan	9.2 Acquire Project Team 9.3 Develop Project Team 9.4 Manage Project Team		
10. Project Communications Management	10.1 Identify Stakeholders	10.2 Plan Communications	10.3 Distribute Information 10.4 Manage Stakeholder Expectations	10.5 Report Performance	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses		11.6 Monitor and Control Risks	
12. Project Procurement Management		12.1 Plan Procurements	12.2 Conduct Procurements	12.3 Administer Procurements	12.4 Close Procurements

Ilustración 1. Descripción general de las áreas de conocimiento en la GP.

Fuente: PMBOK, 2008

Manejo de la Integración

Comprende a todos los procesos requeridos para identificar, definir, combinar y unificar todas las actividades, procesos y elementos del proyecto. El PMBOK (2008)

define a el manejo de la integración como la relación principalmente con la integración efectiva de los procesos entre los grupos de procesos de la dirección de proyectos, que son necesarios para lograr los objetivos del proyecto dentro de los procedimientos definidos por la organización". Los procesos que incluye el manejo de la integración son los siguientes:

1. Desarrollar el Acta de Constitución del Proyecto: autorización formal del proyecto o una fase de un proyecto.
2. Elaborar el Plan de Gestión del Proyecto: Es el documento base donde se incluyen las acciones necesarias para definir, integrar y coordinar los distintos planes de gestión por área de aplicación que se dan en el proyecto.
3. Dirigir y Gestionar la Ejecución del Proyecto: hacer cumplir por parte del gerente de proyecto y su equipo las acciones necesarias para cumplir con el trabajo definido en el enunciado del alcance del proyecto y los objetivos planteados.
4. Supervisar y Controlar el trabajo del proyecto: se realiza para supervisar los procesos del proyecto relacionados con el inicio, planificación, ejecución y el cierre del proyecto.
5. Control Integrado de los Cambios: ya que es muy raro que los proyectos se realicen exactamente como fueron planificados, el control de cambios asegura que se mantenga actualizado el Plan de Gestión, el enunciado del alcance del proyecto así como cualquier otro producto entregable, ya sea por aceptación o rechazo de la efectiva gestión continua de los cambios.

6. Cerrar Proyecto: implica el cierre del proyecto según el Plan de Gestión del Proyecto. Incluye tanto el cierre administrativo del proyecto así como el cierre del contrato establecido para el proyecto.

Manejo del Alcance

La gestión del alcance del proyecto, según el PMBOK (2008), incluye todos los procesos necesarios para llevar a cabo el proyecto satisfactoriamente e inclusive establece lo que no debe formar parte del mismo. Es la declaración principal de que es lo que se quiere hacer y cómo hacer para lograrlo. Agrupa los procesos de:

1. Recolectar requerimientos: este proceso recoge y explica de forma detallada todos los requerimientos de los "stakeholder" y los convierte en objetivos del proyecto.
2. Definición del Alcance: desarrollar el enunciado del alcance del proyecto que sirva de guía para el desarrollo futuro del mismo.
3. Estructura Desagregada del Trabajo (EDT): es una herramienta que permite la descomposición jerárquica del trabajo que será ejecutado por los miembros del equipo de proyecto, para lograr los objetivos y crear los servicios o productos a ser entregados.
4. Verificación del Alcance: formalizar la aceptación de los servicios o productos definidos en el alcance del proyecto por parte de los actores involucrados.
5. Control del Alcance: establecer los mecanismos para controlar los cambios de alcance en el proyecto.

Manejo del Tiempo

Para el PMBOK (2008) la Gestión del Tiempo comprende los procesos necesarios para asegurar que el proyecto se va a realizar en el tiempo ideal estimado. Incluye los siguientes procesos:

1. Definición de las Actividades: identifica el conjunto de actividades concretas del cronograma que deben ser ejecutadas para poder obtener los productos entregables planificados.
2. Secuencia de las Actividades: determinar la dependencia y relaciones lógicas entre las actividades. Pueden o no estar ordenadas lógicamente indicando precedencia entre las actividades.
3. Estimación de los Recursos de las actividades: determinar que recursos son asignados a cada actividad, tanto a nivel de recursos humanos como de materiales y equipos, así como la cantidad y disponibilidad necesaria para cada actividad.
4. Estimación de la Duración de las Actividades: determinar el tiempo en cronograma para cada actividad a través de la información disponible por el alcance del trabajo, los tipos, cantidades y disponibilidad de recursos necesarios.
5. Desarrollo del Cronograma: determinar por medio de un proceso iterativo las fechas de inicio y fin de cada actividad planificada. Es la línea base con la cual se medirá el avance del proyecto en cuanto al tiempo.
6. Control del Cronograma: incluye las tareas necesarias que proporcionan información del estado actual del proyecto, intermediar

entre los factores que pudieran modificar el cronograma del proyecto y gestionar cualquier posible cambio dentro del mismo.

Manejo de los Costos

La Gestión del Área de Costos incluyen a todos los procesos que se deben tener en cuenta a la hora de planificar, estimar, preparar el presupuesto base y manejar el control para que el proyecto se realice dentro de los costos inicialmente aprobados. Incluye los procesos de

1. Estimación de costos: determinar una aproximación de los costos de los recursos necesarios a fin de poder completar las actividades planificadas en el cronograma.
2. Determinar los costos: qué cantidad y en qué tiempo deben desembolsarse los costos estimados de las actividades o paquetes de trabajo individuales planificados en el cronograma, con el fin de determinar una línea base de costos totales y poder medir el rendimiento del proyecto.
3. Control de Costos: Palacios (2004) define el control de costos, "implica hacer una revisión contable de los costos acumulados en el proyecto y hacer las comparaciones versus el presupuesto definitivo, determinando las variaciones, tomando los correctivos necesarios y aprendiendo las lecciones pertinentes".

Manejo de la Calidad

El manejo de la calidad en los proyectos se basa en aplicar metodologías y estándares requeridos así como procedimientos de monitoreo de los principales indicadores, con la finalidad de satisfacer que el proyecto cumpla con las necesidades por las cuales se emprendió. Esta área de conocimiento agrupa los procesos de:

1. Planificación de la Calidad: determinar cuáles son las normas o estándares principales que se pueden aplicar al proyecto y determinar de qué modo pueden satisfacerse.
2. Aseguramiento de la Calidad: Palacios (2004) afirma que "para poder asegurar que algo cumple con lo deseado, hay que identificar las cualidades del producto que le permitirán alcanzar un nivel de alta calidad". Esto se puede lograr mediante el establecimiento de un buen sistema de calidad que permita medir los atributos por medio de indicadores e instrumentos de medición.
3. Control de la Calidad: incluye el monitoreo y supervisión, a lo largo del ciclo de vida del proyecto, con el fin de determinar si se están cumpliendo las normas de calidad o estándares aplicados según la planificación y así corregir posibles resultados insatisfactorios.

Manejo de los Recursos Humanos

El PMBOK (2008) indica que el Manejo del Recurso Humano es una tarea que debe de realizarse de forma bien estructurada y planificada por parte del

gerente de proyecto. El mismo actúa como líder del equipo, por tanto debe de tratar de convertir al grupo en equipos efectivos que avancen hacia la culminación exitosa del proyecto. La gestión incluye a todos los procesos que tienen que ver con la organización y dirección del equipo. El PMBOK (2008) describe los procesos de esta área como:

1. Planificación de los Recursos Humanos: identificar los roles y responsabilidades de cada miembro del equipo y a su vez se crea el plan de cómo gestionar el recurso humano involucrado en el proyecto
2. Contratar el Equipo de Proyecto: contratar el recurso humano necesario para culminar el proyecto.
3. Desarrollar el Equipo de Proyecto: Se busca mejorar las habilidades del equipo y así aumentar las capacidades, con el objetivo de completar las actividades del proyecto. Así mismo lograr una mejor cohesión entre los miembros, esto con miras a incrementar la productividad a través de una mayor participación en equipo.
4. Gestionar el Equipo de Proyecto: hacer un seguimiento y control del rendimiento general e individual de los miembros del equipo, recibir retroalimentación y coordinar posibles cambios con miras a subir el rendimiento.

Manejo de las Comunicaciones

En todo proyecto la información es un activo intangible de incalculable valor. Es por ello que es de vital importancia que las acciones de generación, acopio,

distribución, resguardo, recuperación y destino final de la información sean realizadas de forma muy bien planificada. Las comunicaciones proveen el enlace entre las personas involucradas y la información general del proyecto. Los procesos que se contemplan en el manejo de las comunicaciones son los siguientes:

1. Identificar los Stakeholders: En este proceso se identifican a aquellas personas involucradas de alguna manera en el proyecto con la finalidad de tratar de conocer su interés, la participación, y el impacto en el éxito del proyecto. Ha de incluir los que estén a favor y en contra del proyecto.
2. Planificación de las Comunicaciones: determinar los requerimientos de información y comunicación de los involucrados en el proyecto.
3. Distribución de la información: poner la información necesaria de manera oportuna a disposición de los interesados.
4. Gestionar a los Involucrados: se refiere a gestionar las comunicaciones con la finalidad de satisfacer las necesidades de los involucrados en el proyecto y tratar de resolver posibles conflictos entre ellos. Esto deriva en aumento de la probabilidad de que el proyecto no se desvíe de sus objetivos planteados y promueve el trabajo de forma sinérgica entre los miembros del equipo.
5. Informar el Rendimiento: elaborar y distribuir los reportes de progreso y cambios a lo largo del ciclo de vida activa del proyecto en base a la recolección de todos los datos y la distribución de la información sobre el rendimiento.

Manejo de los Riesgos

La Gestión de los Riesgos está muy ligada al área de la Calidad. Se trata de determinar aquellos factores que puedan influir o alterar los resultados generales del proyecto. Incluye todos los procesos relacionados con la planificación, identificación, análisis, respuesta, así como el seguimiento y control de los riesgos que pueden presentarse en un proyecto. Se busca aumentar tanto la probabilidad como el impacto de los eventos positivos y análogamente disminuir los eventos negativos y su probabilidad de ocurrencia. Contempla los siguientes procesos:

1. Planificar la Gestión de los Riesgos: donde se establece como se van a definir, controlar y ejecutar las actividades de la gestión de riesgos.
2. Identificar los Riesgos: donde se determinan que riesgos pueden afectar el proyecto.
3. Analizar Cualitativamente los Riesgos: analizar probabilidad ocurrencia e impacto.
4. Analizar Cuantitativamente los Riesgos: analizar numéricamente el efecto de los riesgos.
5. Planificar las Respuestas a los Riesgos: plantear las opciones y acciones para mejorar las oportunidades, así como reducir las amenazas.
6. Seguimiento y Control de los Riesgos: realizar el seguimiento de los riesgos ya identificados, de nuevos riesgos que pudieran ocurrir, actualizar los planes de respuesta a los riesgos y evaluar se efectividad.

Manejo de las Adquisiciones

Aquí se agrupan a todos los procesos que se relacionan con la compra o adquisición de los productos o servicios necesarios para realizar el trabajo. Incluye los procesos siguientes:

1. Plan de Compras: determinar cuáles recursos planificados no están disponibles por el equipo y por tanto deben ser suministrados por una empresa externa. Es definir que se necesita, cuando y como hacerlo.
2. Planificar de las Requisiciones: generar en base al plan de compras y la información del mercado las respectivas órdenes de compra y los criterios de selección de los proveedores.
3. Administración del Contrato: gestionar el contrato y la relación con cada proveedor para garantizar que el contratista cumpla con lo convenido. Es un proceso continuo que debe hacer el ente contratante.
4. Cierre del Contrato: Palacios (2004), lo define como "... todas las labores que deben ejecutarse para finalizar la relación contractual, realizándose las auditorías pertinentes, analizándose los resultados y posibles aprendizajes provenientes de la experiencia".

Plan de Ejecución del Proyecto

El Plan de Ejecución puede definirse como un documento base para el planeamiento, supervisión y gerencia de un proyecto. Es una declaración de las políticas y procedimientos definidos por el Gerente del Proyecto y utilizado, mantenido y actualizado por los miembros del equipo de proyecto.

Proporciona una guía para asegurar el control de las operaciones cotidianas así como la calidad de los productos/servicios esperados.

El Plan de Ejecución de un proyecto está relacionado con la naturaleza misma del proyecto, es decir, no existe un estándar definido para la creación de un Plan de Ejecución.

Fundamentos Teóricos de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software (RUP).

También llamado RUP por sus siglas en inglés (Rational Unified Process), es un proceso iterativo de Desarrollo de Software creado en Rational Software Corporation por Jacobson, Booch y Rumbaugh (autores principales) en el año 1998, el cual se basa en la implementación de las mejores prácticas conocidas en ingeniería del software durante los últimos 30 años, agrupadas en una metodología unificada. Su principal objetivo es servir de guía en forma disciplinada para la implementación y distribución eficiente[4] de sistemas que se ajustan a las necesidades de los clientes.

RUP es un Proceso de Desarrollo de Software, es decir, el conjunto de técnicas necesarias para transformar los requisitos de un usuario en un sistema de software. Se basa en la construcción de componentes o productos que van amoldando el software a través del Lenguaje Unificado de Modelado (UML – Unified Modeling Language), lo que a su vez constituye la metodología estándar más utilizada[5] para el análisis, implementación y documentación de Sistemas Orientados a Objetos.

---

[4] Eficiencia medida en costo, calidad y tiempo

[5] Según la Object Management Group (OMG), consorcio dedicado al establecimiento de diversos estándares de tecnologías orientadas a objetos. Disponible en: www.uml.org. Consultado el 27-10-2009

Según Jacobson, RUP es un proceso iterativo a través de una serie de ciclos que constituyen la vida del sistema. Cada ciclo está compuesto de 4 fases y cada fase está compuesta por una o más iteraciones, que pasan por una serie de flujos de trabajo ordenados que sirven para guiar el desarrollo. En cada flujo de trabajo se producen una serie de diagramas o artefactos, que constituyen una pieza o producto de información tangible.

En esencia RUP está definido por tres aspectos:

- Dirigido por los Casos de Uso.
- Centrado en la Arquitectura.
- Es un proceso iterativo e incremental.

Dirigido por los Casos de Uso.

Un Caso de Uso es una especificación de una Unidad Funcional provista por un sistema, es decir, el Caso de Uso abarca la secuencia de acciones que un sistema lleva a cabo a través de una función, desde que se invoca por un usuario hasta que entrega los resultados al mismo, dicho en otras palabras, "describe un servicio completo que el sistema proporciona a un actor (usuario) que lo solicita". (Montilva, 2006)

El conjunto de todos los Casos de Uso describen a su vez la funcionalidad general de un sistema y se conoce como Modelo de Casos de Uso (MCU). El MCU se utiliza para la captura de requisitos de un sistema, sin embargo, los Casos de Uso no son sólo una herramienta para especificar los requisitos de un sistema, también guían su diseño, implementación, y prueba, es decir, guían el proceso de desarrollo, ya que a partir del MCU se obtienen una serie de distintos modelos que representan el diseño de

la arquitectura, la implementación, así como las pruebas de un sistema. Por todo esto se puede decir que el Proceso Unificado es dirigido por Casos de Uso ya que sigue un flujo de trabajo que comienzan o se inician a partir de los Casos de Uso.

Centrado en la Arquitectura

En la Ingeniería Civil (por ejemplo, la construcción de un edificio) la arquitectura representa una visión general desde distinto puntos de vistas: planos de la estructura, electricidad, planos de aguas servidas, entre otros aspectos. En un Sistema de Software la arquitectura representa los aspectos estáticos y dinámicos que componen el sistema. En ambos casos lo que se desea es tener una representación general completa antes de comenzar la construcción.

El Proceso Unificado está centrado en la arquitectura ya que según Jacobson "la arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado" (Jacobson, et al, 2000). Un sistema de software está compuesto por múltiples productos o componentes, cada uno de ellos debe tener una función y una forma específica, los Casos de Usos representa la funcionalidad mientras que la arquitectura representa la forma. El sistema es moldeado por los arquitecto de software para obtener la forma y para encontrar la misma deben trabajar sobre la comprensión de las funciones claves, es decir, sobre los Casos de Uso.

Es un proceso Iterativo e Incremental.

Dado que los proyectos software son por lo general a mediano y largo plazo es común dividir el trabajo en mini proyectos. Cada mini proyecto es una iteración que resulta a su vez en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, y los incrementos a un crecimiento en el producto. Para ser más efectivas las iteraciones deben ser controladas, es decir deben ser seleccionadas y llevadas a cabo de una forma planeada, de forma que cada una de las iteraciones constituye un mini proyecto software.

Los desarrolladores basan la selección de las tareas a abordar en cada iteración partiendo de dos factores: la iteración maneja un grupo de Casos de Uso que extienden la utilidad del producto, además las iteraciones tratan siempre con los riesgos más altos en el estado actual del proyecto. Pero el desarrollo iterativo no es dividir, sino que las sucesivas iteraciones se construyen a partir de los componentes en el estado en que fueron dejados en las iteraciones precedentes.

Cada iteración, considerada como un mini proyecto, a partir de los Casos de Uso aborda el análisis, diseño, implementación y pruebas. Un incremento no es necesariamente aditivo, ya que, especialmente en las primeras fases del ciclo de vida, los desarrolladores pueden estar reemplazando un diseño superficial por un diseño más detallado. En las fases posteriores los incrementos suelen ser aditivos.

En cada iteración los desarrolladores analizan las Casos de Uso relevantes, crean un diseño utilizando como guía la arquitectura elegida, e implementan el diseño en componentes verificando que dichos componentes aportan la funcionalidad especificada en los Caso de Uso. Si una iteración cumple sus objetivos el proceso continúa con la siguiente iteración, por contra, si una iteración

no cumple sus objetivos los desarrolladores deben revisar las decisiones previas e intentar una nueva aproximación.

Para alcanzar el mayor rendimiento, un proyecto debería proceder casi siempre a lo largo de su curso directo con solo pequeñas desviaciones del curso inicialmente planeado. Minimizar los imprevistos es uno de los objetivos de la reducción de riesgos.

Fases del Proceso Unificado

Un proyecto realizado siguiendo la metodología RUP se divide en cuatro fases principales:

- Fase de Inicio: Esta fase es donde se inicializa la puesta en marcha del proyecto con el desarrollo de una descripción del producto final (en cuanto a objetivos y alcance) a partir de una buena idea.
- Fase de Elaboración: en esta fase se especifican en detalle la mayoría de los Casos de Uso y se diseña la arquitectura inicial del sistema. La definición de la arquitectura debe tener en cuenta los requerimientos obtenidos en la etapa de inicio y proveer las alternativas para el Control de Riesgos.
- Fase de Construcción: se realiza la implementación o creación del producto con la finalidad de crear una versión operativa inicial.
- Fase de Transición: marca el fin del proyecto y puesta en producción a través de una versión beta operativa del producto.

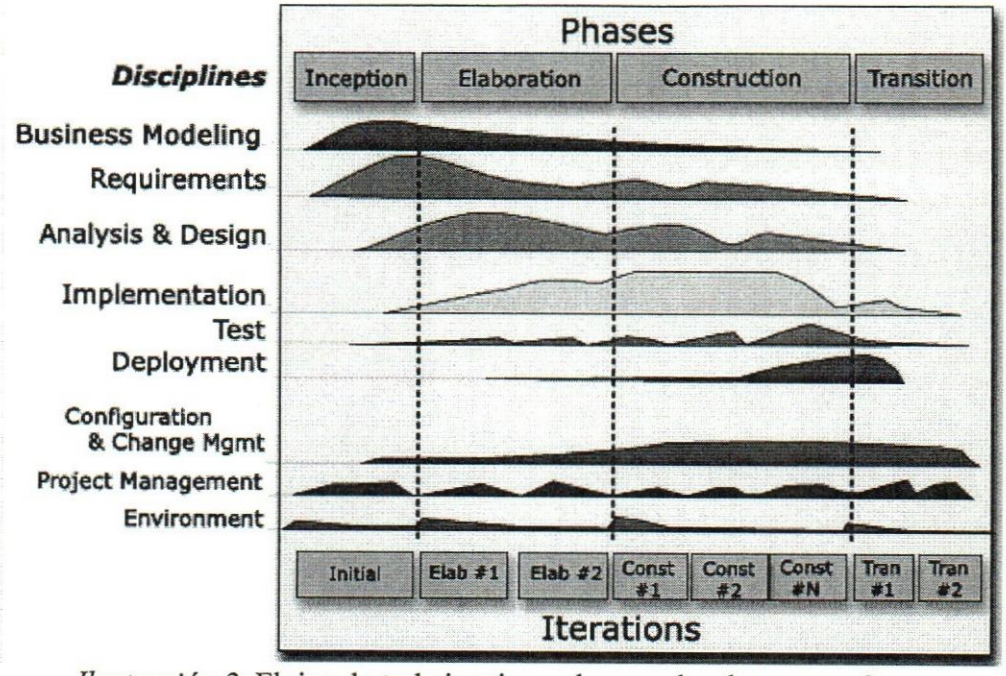


Ilustración 2. Flujos de trabajos tienen lugar sobre las cuatro fases

Fuente: (Jacobson, Booch, & Rumbaugh, 2000)

En cada fase se construyen los modelos que representa una abstracción del sistema.

¿Qué es un modelo?

Rumbaugh, *et al* (2000) define modelo como "una representación, en cierto medio, de algo en el mismo medio u otro medio". Es decir, es la representación de un sistema desde distintas perspectivas. Por ejemplo, existen modelos que simboliza requisitos funcionales, que no representa en sí como se está implementado el sistema, sino que sólo definen la funcionalidad a nivel de usuario. También hay modelos que representa al sistema desde un punto de vista más de implementación, o de comportamiento.

Cada modelo representa una vista del sistema desde un punto de vista específico y está dirigido a un tipo de usuario específico (Ej. desarrollador, cliente, director, otros).

Los modelos son la base para representar gráficamente y de un modo sencillo todos los aspectos que conforman un sistema de software.

Lenguaje Unificado de Modelado (UML)

Es un lenguaje gráfico de modelado visual utilizado para especificar, visualizar, construir y documentar componentes de un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

UML es el producto de una combinación de más de 20 estándares en uno solo. Se emplea para modelar o diseñar productos de software antes de su construcción.

Al igual que cualquier lenguaje, UML incluye aspectos de sintaxis y conceptos semánticos, define una notación que representa la sintaxis gráfica que expresa la semántica e incluye una serie de principios generales. Puede utilizarse en cualquier proceso o metodología de desarrollo, sin embargo está concebido para ser de gran utilidad de procesos de desarrollo iterativos, como el Proceso Unificado por ejemplo.

El UML cuenta con varios modelos que permiten modelar un sistema de software desde una perspectiva funcional (usos del sistema) así como estructural (abstracción del sistema), comportamiento (aspectos dinámicos) e implementación (componentes del sistema). La versión actual de UML (2.0) consta de un conjunto de 13 modelos, también llamados diagramas, que se clasifican en dos grandes grupos:

Diagramas para modelado estructural y diagramas para modelado del comportamiento.

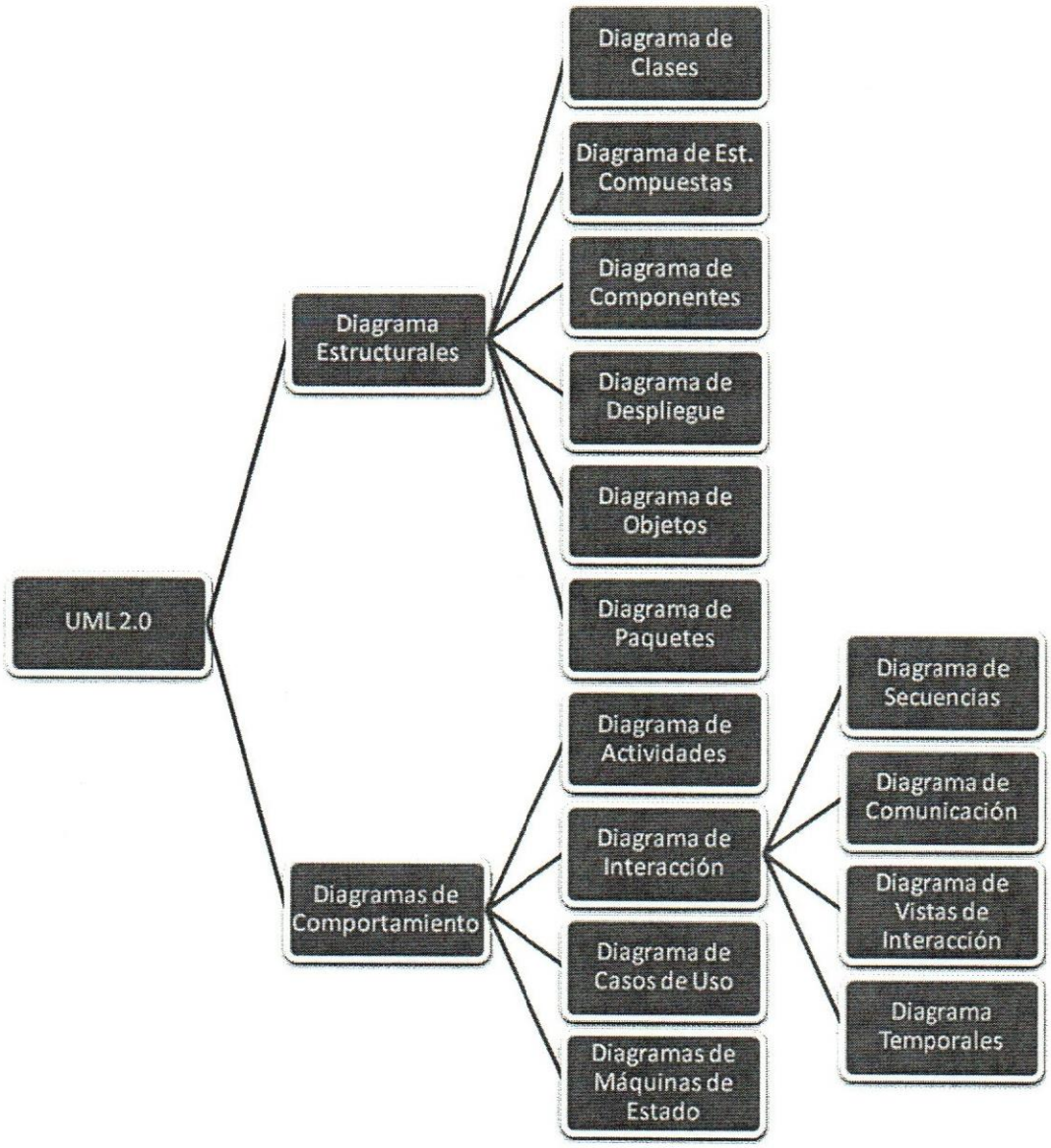


Ilustración 3. Modelos que implementa UML 2.0

Fuente: (Montilva, 2006). Elaboración propia

El Modelo de Estimación COCOMO.

Boehm (1981), en su libro clásico que data de 1981 sobre la economía de la ingeniería de Software, introdujo una jerarquía de modelos de estimación de software

con el nombre de COCOMO por su nombre en inglés (Constructive, Cost, Model), que permiten estimar esfuerzo, costo, y tiempo para los proyectos del software. La jerarquía de Boehm está constituida por los siguientes modelos:

Modelo I. El modelo COCOMO básico calcula el esfuerzo y el costo del Desarrollo de Software en función del tamaño del programa, expresado en las líneas estimadas.

Modelo II. El Modelo COCOMO intermedio calcula el esfuerzo del Desarrollo de Software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Modelo III. El Modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de software. (El Modelo de COCOMO, 2009)

Técnicas de Estimación de Costos del Software

Los Modelos COCOMO están definidos para tres tipos de proyectos de software. Utilizando la terminología de Boehm son:

Modelo Orgánico: Proyectos de software relativamente pequeños y sencillos en donde trabajan pequeños equipos, con buena experiencia en la aplicación, sobre un conjunto de requisitos poco rígidos.

Modelo Semiacoplado: Proyectos de software intermedios en los que equipos, con variados niveles de experiencia, deben satisfacer requerimientos poco o medio rígidos.

Modelo a la Medida: Proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringidas.

La ecuación del COCOMO básico tiene la siguiente forma

$$E = a_b \, KLDC \, b_b$$

$$D = c_b \, E \, d_b$$

En donde E es el esfuerzo aplicado en personas-mes, D es el tiempo de desarrollo en meses cronológicos y KLDC es el número estimado en líneas de código distribuidas (en miles) para el proyecto. Los coeficientes a_b y c_b y los exponentes b_b y d_b se muestran en la tabla siguiente:

Proyecto de Software	a_b	b_b	c_b	d_b
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3	1,12	2,5	0,35
A la medida	3,6	1,2	2,5	0,32

Capítulo 3. Marco Metodológico

El siguiente capítulo específica el tipo de estudio y el diseño de la investigación planteada así como los aspectos tanto técnicos como operacionales que se tomaron en cuenta para analizar y diseñar un Sistema de Gestión de Equipos Audiovisuales.

Tipo de Investigación

La investigación se realizó bajo el enfoque de proyecto factible apoyado en un diseño bibliográfico, ya que se planteó un problema de investigación que debía generar una solución práctica para la Dirección de Tecnologías de la Información en cuanto a los Procesos de Gerencia de Proyectos. El manual UPEL (2005) define que un proyecto factible:

> "es la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo bibliográfico, de campo o un diseño que incluya ambas modalidades".

En el diseño bibliográfico los datos se obtienen a partir de la aplicación de técnicas documentales, informes de otras investigaciones o a través de las diversas fuentes documentales.

Dado que la metodología propuesta por la Guía de los Fundamentos de la Dirección de Proyectos que propone el PMI (PMBOK, 2008) no tiene un apartado específico adaptado a Proyectos de Desarrollo de Software que considere la implantación de algún tipo de metodología dentro del ciclo de vida, se propuso una investigación bibliográfica ajustada al problema de investigación. El propósito fue buscar un punto de relación entre las bases propuestas por el PMI sobre la dirección de proyectos y la metodología de desarrollo RUP, con la finalidad de obtener una Guía Práctica de Desarrollo y Gestión de Proyectos que pueda ser utilizado y aplicado en la Dirección de Tecnologías de la Información de la UCAB para los futuros proyectos.

Fue necesario, analizar las bases fundamentales de la Gestión de Proyectos de Software que propone el PMI para la generación de Planes de Ejecución en Proyectos, a su vez fue necesario analizar la metodología RUP para Desarrollo de Software y su aplicación a sistemas basados en Web así como el estudio de la notación UML para determinar los distintos modelos necesario para definir el desarrollo de este tipo de sistemas.

Procedimiento de la investigación

Balestrini (2001) define el diseño de la investigación como "el plan general del estudio que integra coherente y adecuadamente correcto, técnicas de recolección de datos, análisis y objetivos" que respondan las preguntas planteadas. En este sentido el presente trabajo de investigación se dividió en las siguientes fases descritas a continuación

Fase I. Investigación documental

Esta fase perseguía obtener información por medio de diversos medios como lo son bibliografía sobre el tema de estudio, trabajos publicados anteriormente, documentos y revistas especializadas sobre Desarrollo de Software con RUP y UML, Gestión de Proyectos, fuentes electrónicas certificadas como las últimas especificaciones de UML 2.0 y el PMBOK (2008), entre otras; todo esto con la finalidad de generar una banco de datos con información del tema de estudio.

Fase II. Análisis de la información.

En esta fase se analizaron las distintas áreas de conocimientos y procesos que define el PMI necesarios para elaborar un Plan de Ejecución con el propósito de filtrar, clasificar y delimitar los procesos que intervienen en el mismo, adicionalmente se estudiaron las distintas fases del ciclo de vida y las iteraciones de RUP necesarias, así como los distintos modelos de UML a ser aplicados a los procesos ya seleccionados con el fin de obtener una guía de instrumentos o modelos preliminares.

Fase III. Elaboración de Modelos del Plan de Ejecución

Después de definir y seleccionar los procesos que componían el Plan de Ejecución según el PMI y los modelos de UML a ser implementados según RUP se desarrollaron los distintos instrumentos, basándose en la abstracción para la creación de un sistema de información basado en la Web para la Gestión de Equipos Audiovisuales para la UCAB.

Unidad de Análisis

Hernández, et al (2003) define "la muestra es una unidad de análisis o un grupo de personas, contextos, eventos, sucesos, comunidades, etcétera, de análisis; sobre el (la) cual se habrán de recolectar datos, sin que necesariamente sea representativo(a) del universo o población que se estudia".

En la presente investigación las unidades de análisis objeto de estudio la conformaron los distintos modelos que define el lenguaje de modelado UML utilizados mediante la aplicación de la metodología RUP para desarrollo de un sistema. Los mismos comprendieron la totalidad de los elementos que se deseaba investigar y de lo cual se estudió una fracción que pudiera aplicarse en los distintos procesos definidos en el PMBOK en la Gestión de Proyectos. Estos modelos trataron de definir el sistema en su estructura, comportamiento e iteración.

Técnicas de recolección de datos

Ballestrini (2001) define a las técnicas como aquellas que "contienen los procedimientos metódicos y críticos que permiten organizar toda la información escrita que se manejará en el proceso de investigación…". Una vez definido el tipo de investigación, el procedimiento de la misma, así como las unidades de análisis se procedió a enunciar las técnicas e instrumentos utilizados para la recolección de datos.

Para el análisis de las fuentes documental se procedió a realizar una observación documental y lectura, con la finalidad de obtener un esquema resumido para finalmente realizar un análisis más crítico de la información. Se utilizaron técnicas operacionales para manejar las fuentes tales como subrayado de ideas principales en párrafos, bocetos,

diagramación y construcción de modelos UML por medio de herramientas tipo CASE[6] , resúmenes y fichas para manejar las referencias bibliográficas.

[6] Computer Aided Software Engineering (CASE), por sus siglas en inglés. Ingeniería de Software Asistida por Computadora.

Capítulo 4 – Desarrollo

En base a las bases teóricas expuestas en el capítulo 2 sobre la Gerencia de Proyectos y sus distintas áreas de conocimientos así como Desarrollo de Software, a continuación se define el Plan de Ejecución para desarrollar un Sistema de Gestión de Equipos Audiovisuales.

Manejo del Alcance

Iniciación

En esta etapa se reconoce la necesidad de crear y planificar antes de actuar. En tal sentido se hace necesario definir un plan que permita a la Unidad de Desarrollo del DTI controlar todas las actividades a ser ejecutadas durante el desarrollo del proyecto.

La Coordinación General en conjunto con la Coordinación de la Unidad de Desarrollo son las encargadas de dar inicio formal al desarrollo del Sistema de Gestión de Equipos Audiovisuales de la UCAB.

Planificación del Alcance

Para asegurar la correcta definición del alcance basándose en los conceptos en la metodología RUP se detalla la ingeniería de requisitos como primer paso paro luego describir formalmente el Modelo de Negocios a través del diagrama UML Caso de Uso así como el detalle de los Casos de Uso más relevantes del sistema estudio.

Se detalla el proceso de verificación del alcance por medio de la aceptación del trabajo, para seguidamente construir la Estructura Desagregada del Trabajo y finalmente se define y documenta el proceso de Gestión o Control de Cambios en el alcance del proyecto.

Definición del Alcance

Objetivo general del proyecto: Desarrollar un sistema que permita controlar el uso de los equipos audiovisuales por parte de las dependencias de la UCAB de forma centralizada.

Ingeniería de Requisitos

Una buena praxis de la Ingeniería de Requisitos es modelar los Procesos de Negocios antes de definir los requisitos. Los Proceso de Negocios anteceden a la Ingeniería de Requisitos y ambos procesos son complementarios. Por medio de la cadena de valor podemos modelar los Procesos de Negocio de una empresa, que para este estudio se basa en el sistema propuesto.

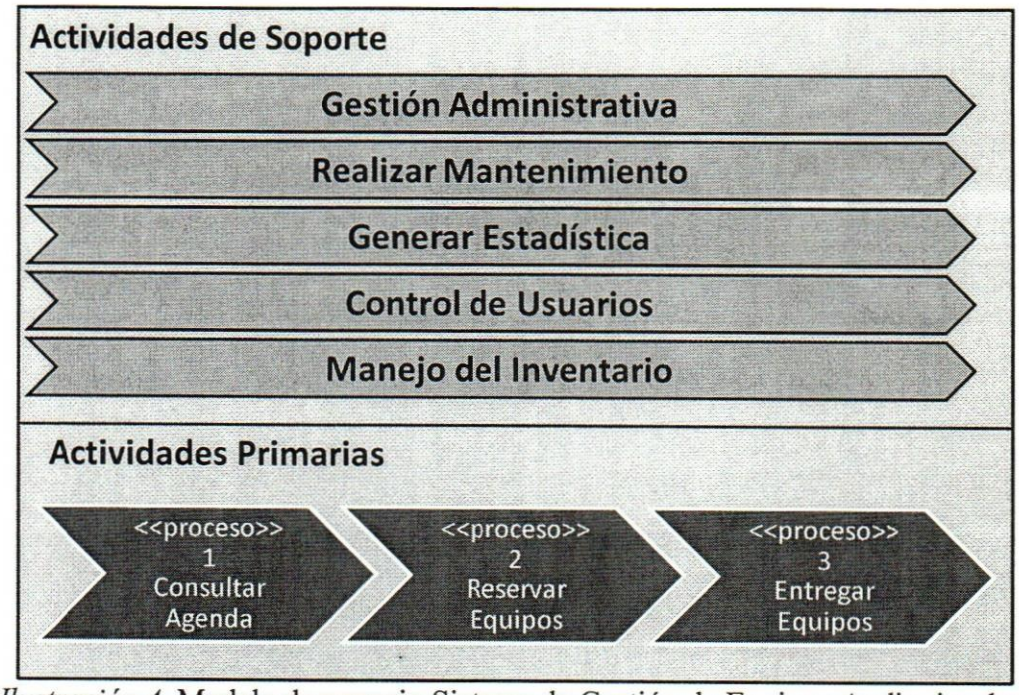


Ilustración 4. Modelo de negocio Sistema de Gestión de Equipos Audiovisuales

Fuente: Elaboración propia.

A partir de la cadena valor y por medio de UML Business se definen los siguientes diagramas que describen la jerarquía de los procesos o actividades primarias.

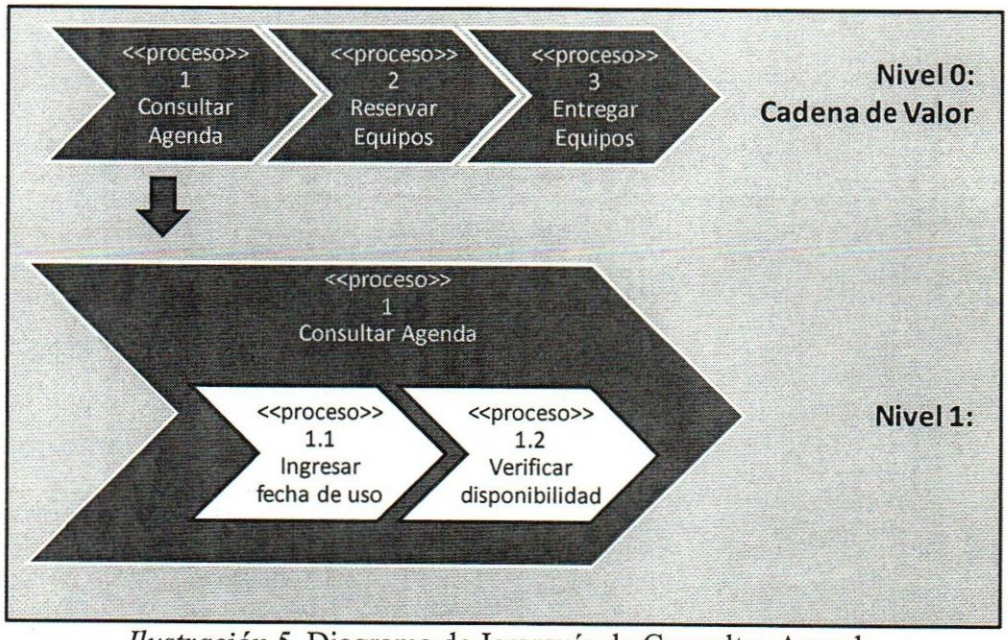


Ilustración 5. Diagrama de Jerarquía de Consultar Agenda

Fuente: Elaboración propia.

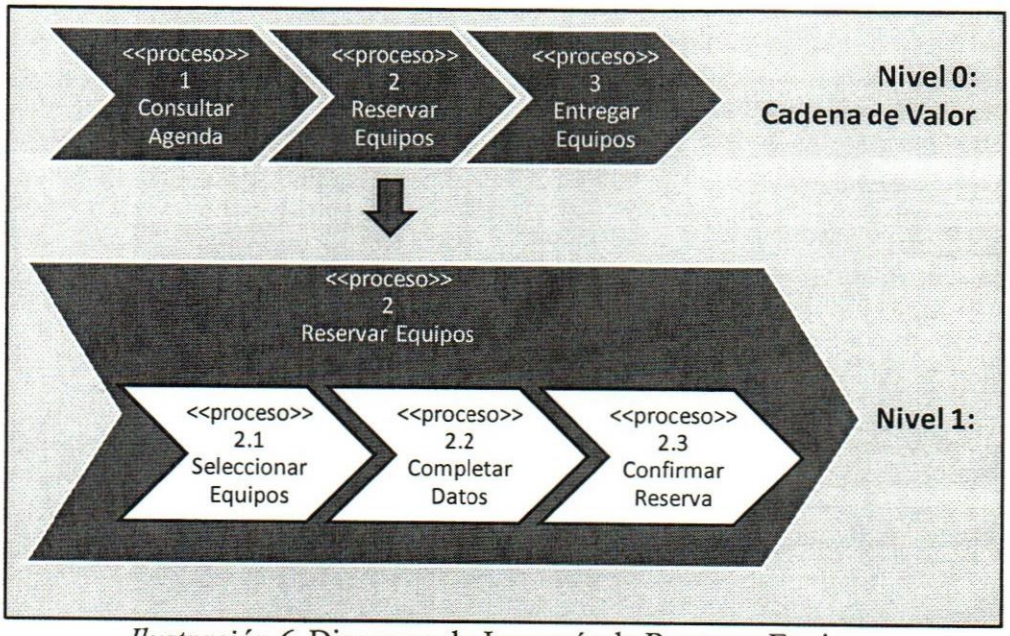


Ilustración 6. Diagrama de Jerarquía de Reservar Equipos

Fuente: Elaboración propia.

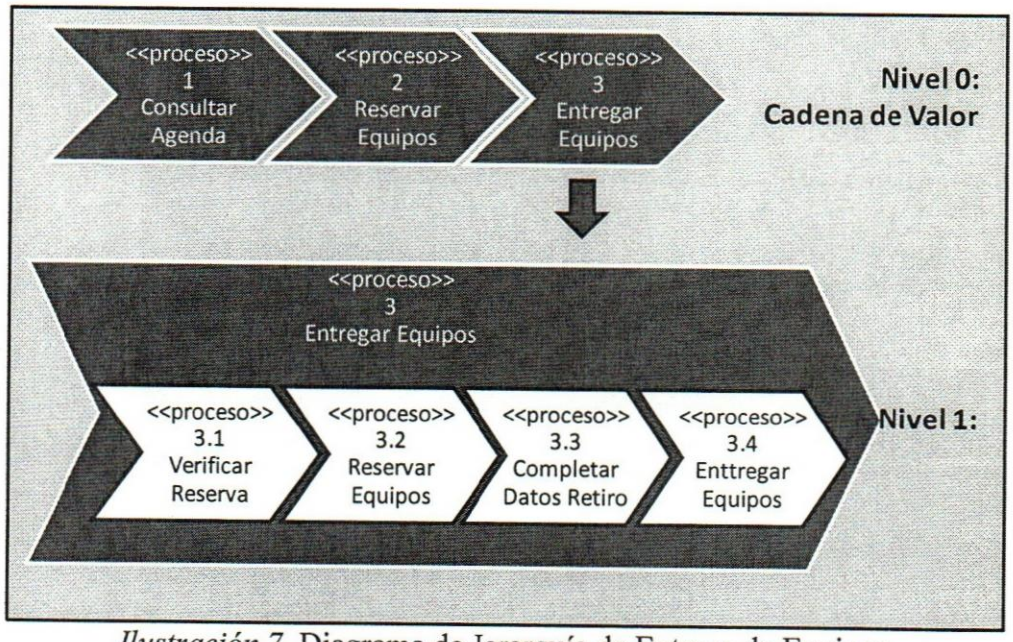


Ilustración 7. Diagrama de Jerarquía de Entrega de Equipos

Fuente: Elaboración propia.

La información que a continuación se incluye ha sido extraída a partir del estudio del Modelo de Negocio, así como de las diferentes reuniones que se realizaron con los actores y usuarios involucrados en el proceso.

Las distintas dependencias de la UCAB actualmente manejan de forma descentralizada sus propios equipos audiovisuales, para lo cual utilizan sus propios métodos y recursos para administrar los mismos. La centralización de la administración de los equipos conllevará a una utilización más efectiva de los equipos así como un resguardo y control del estado más riguroso. Por ello, el DTI considera necesario desarrollar un Sistema de Gestión de Equipos Audiovisuales que apoye el Modelo de Negocio antes descrito así como contar con una base de datos que permita almacenar datos estadísticos que ayuden en la toma de decisiones.

El sistema debe comprender distintos módulos o subsistemas que definan los procesos implicados en la Gestión de Equipos Audiovisuales antes descritos. Estos módulos se pueden agrupar en:

- Módulo de agenda: este módulo se encargará de implantar el proceso de administrar la agenda de los equipos según la dependencia a las que son asignados. Contempla la consulta por rangos amplios de fecha y la disponibilidad de los equipos a ser utilizados.
- Modulo de reservaciones: encargado de administrar las distintas reservas que realicen las escuelas o dependencias a sus usuarios. Las dependencias deben poder establecer el modo de cómo realizan las reservas, es decir, individualmente o por bloques de tiempo periódicos fijos. Además, requiere

controlar qué usuario realiza la reserva y qué equipos son los asignados a una reserva. Así mismo este módulo modelaría el comportamiento del proceso entrega de equipos, ya que por medio de la reserva se identifica al usuario que retira el equipo así como el que lo devuelve y el estado del mismo.

- Módulo para el manejo del inventario: este proceso se encarga de manejar el inventario de los equipos, habilitar o deshabilitar equipos en funcionamiento, así como la diferenciación de los distintos tipos de equipos. A su vez el sistema ha de permitir la asignación de recursos a las escuelas y manejar ellas mismas su propio stock. Modela en parte el comportamiento de los Procesos de Gestión Administrativa.
- Módulo de estadísticas y consulta: Se encarga de implantar el proceso de generar las estadísticas globales y detalladas por dependencias, usuarios, equipos, por rangos de fechas, por uso efectivo, vida útil de los consumibles, estadísticas de reparaciones, tiempos medios entre fallas, tipo de falla, tiempo de reparación, quién lo reparó, entre otros reportes que apoyen la Gestión Administrativa y la toma de decisiones.
- Módulo de mantenimiento: Se encarga de controlar y registrar los daños que puedan sufrir los equipos, cuáles requieren mantenimiento y que tipo de servicio que se les ha efectuado. Manejar la activación, desactivación y desincorporación de los equipos. Este módulo complementa al proceso de Gestión Administrativa.
- Módulo de control de usuarios: este módulo define el proceso de control de usuarios. Su función principal es administrar la incorporación y/o

desincorporación de usuarios del sistema. Administrar las sanciones que pueden aplicarse a los usuarios del sistema. Asignar los distintos roles a los usuarios del sistema (dependencia, operador, administrador).

Restricciones y suposiciones

Para el desarrollo de este sistema es necesario tener presente las siguientes suposiciones y restricciones respecto del mismo:

- El Sistema de Gestión de Equipos Audiovisuales a desarrollar es necesario que se acople al el núcleo de la Intranet de la UCAB (BioUcab[7]) de forma no dependiente, es decir, que pueda funcionar tanto autónomamente como dependiente del manejo de usuario/roles de la Intranet.
- Deben contemplarse las implicaciones de los siguientes puntos críticos:
    - Manejo de múltiples interfaces según roles de usuario.
    - Arquitectura LAMP[8]
    - Gestión de flujos de trabajo, seguridad de transacciones e intercambio de información.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, como parte del Control del Alcance del Proyecto.

---

[7] BioUCAB es el nombre como se conoce al sistema de información de la Intranet de la UCAB.
[8] Linux-Apache-MySQL-Perl/PHP/Phyton por sus siglas en inglés, el término describe a las aplicaciones desarrolladas en ambientes Web creadas utilizando esa combinación de herramientas: Linux, como sistema operativo; Apache, como servidor Web; MySQL, como gestor de bases de datos; Perl, PHP, o Phyton como lenguajes de programación

Modelo de Casos de Uso del Negocio

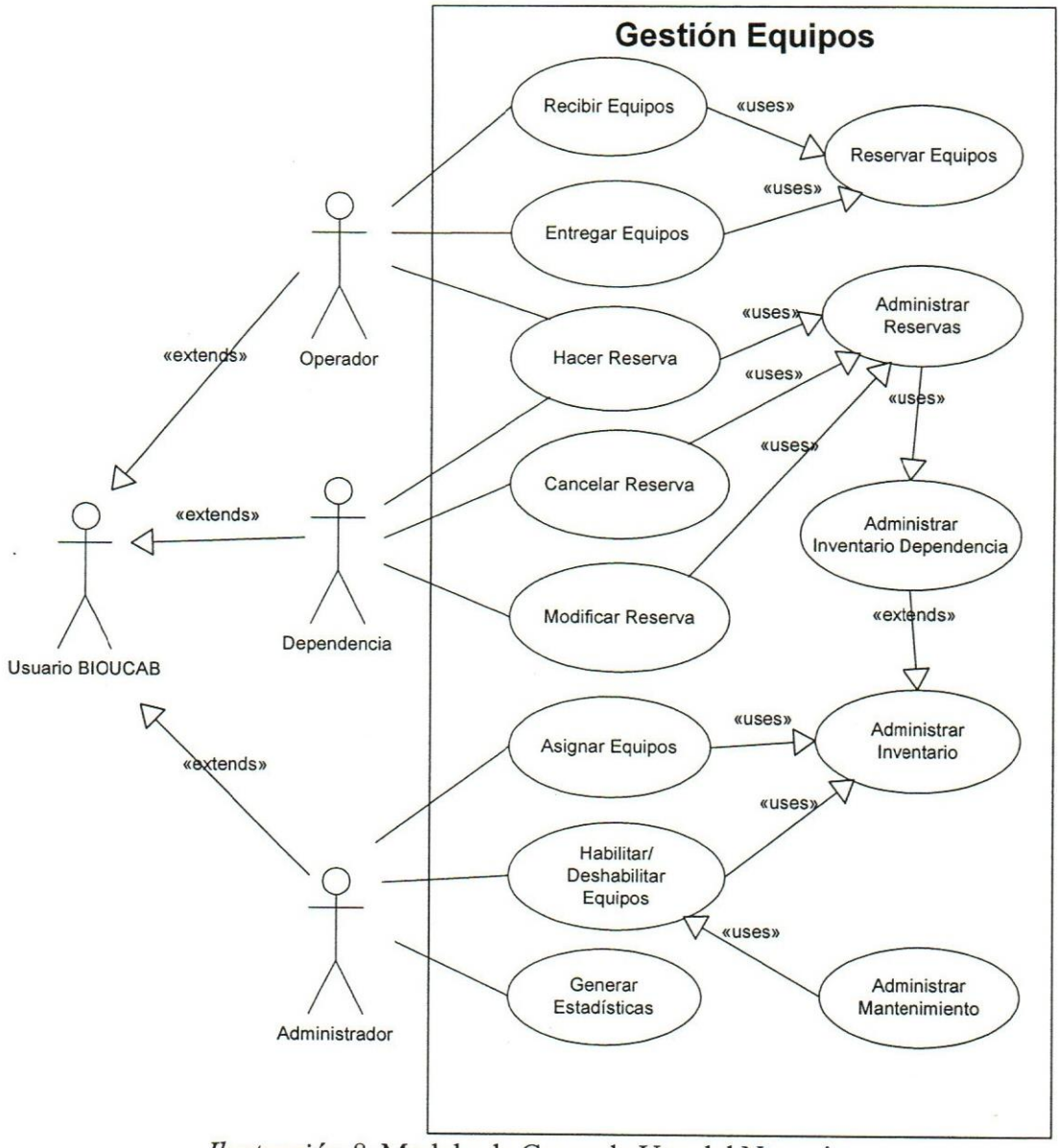


Ilustración 8. Modelo de Casos de Uso del Negocio

Fuente: Elaboración propia.

Actores:

- Operador: representa al personal encargado de la custodia, entrega, revisión y recepción de los equipos. Estos pueden también reservar equipos a usuarios de distintas dependencias.

- Dependencia: representa al personal administrativo de las escuelas o dependencias encargado de procesar las reservas de equipos por parte de sus usuarios.
- Administrador: ejerce funciones de administración del sistema. Se encarga de asignar los usuarios y especificar los roles de cada uno, asignar y manipular el estado de los equipos, manipular el inventario.

Todos los actores tienen una relación de dependencia con el actor Usuario BioUcab, ya que por restricción el sistema debe acoplarle a la Intranet, por ende todos estos actores representa a su vez a el rol de usuario Intranet.

Descripción contextual de los principales Casos de Uso.

Especificación de Caso de Uso: Recibir Equipos

Descripción

Este Caso de Uso lo inicia el operador encargado de turno. Su función es recibir los equipos y modificar el estado actual por recibido, además de verificar tanto la integridad como la identidad del equipo.

Flujo de Eventos

- El sistema muestra la lista con los equipos en tránsito o préstamo circulante.
- El operador puede cerrar la reserva al recibir el equipo o registrar una observación para el administrador producto de la revisión del equipo.
- Si elige cerrar la reserva, el operador debe llenar los datos de la persona que entrega el equipo.

- Si registra una observación deberá introducir los datos de la misma y el sistema genera una alerta para el administrador

Precondiciones

El operador deber haber realizado y registrado la entrega de los equipos en el sistema

Poscondiciones

En caso de haberse registrado o modificado una observación los datos de la misma quedan almacenados en la base de datos.

Especificación de Caso de Uso: Entregar Equipos

Descripción

Este Caso de Uso lo inicia el operador encargado de turno. Su función es revisar los equipos, entregarlos y modificar el estado en el sistema

Flujo de Eventos

- El sistema muestra la lista con los equipos reservados para ese día.
- El operador puede entregar los equipos asignados o cancelar la reserva
- Si elige cerrar la reserva, el operador debe llenar los datos y el motivo de la misma.
- Si la reserva y préstamo se ejecutarán el operador deberá introducir los datos de la persona que retira los equipos (si difieren del que tiene asignado los equipos) y verificar que los datos del equipo a entregar corresponden con los datos del equipo asignado a la dependencia.

Precondiciones

- El sistema debe tener registrada una reservación ya sea por el mismo o por el usuario de la dependencia a la que pertenece el equipo.
- El equipo debe de estar en perfecto estado y con todos sus cables accesorio para su utilización (video, corriente, datos, control remoto si aplica).

Poscondiciones

- El equipo quede registrado en la base de datos con un estado de "en tránsito".

Especificación de Caso de Uso: Hacer Reserva

Descripción

Este Caso de Uso puede ser iniciado tanto por el operador como por el usuario de la dependencia. Su función es registrar una reserva para un día específico para un usuario de la comunidad.

Flujo de Eventos

- El sistema muestra la agenda y la disponibilidad según los días y horas.
- El operador puede entregar los equipos asignados a la dependencia y la agenda organizada por mes o semana con los días y horas disponibles para asignación.
- Si elige reservar para un día específico, el operador o la dependencia debe llenar los datos del miembro (estudiante o profesor) que solicita el equipo, motivo de la solicitud, lugar y fecha donde se utilizarán los equipos y otros datos específicos de la materia.
- Si la reserva se requiere por períodos recurrente el sistema verificará que la disponibilidad sea total, de lo contrario informará al operador o dependencia que días no pueden ser asignados por que se encuentran ocupados.

Precondiciones

- El Usuario quien hace la reserva debería ser miembro y tener una cuenta activa en la Intranet.
- El usuario debe de tener aula asignada por el sistema administrativo de la UCAB. En caso de no ser así debe acudir con la coordinación de la materia para solicitar la misma.

Poscondiciones

- La reserva queda registrada en la base de datos. El administrador de reserva gestiona la transacción con el manejador de base de datos.

Especificación de Caso de Uso: Cancelar Reserva

Descripción

Este Caso de Uso puede ser iniciado tanto por el operador como por el usuario de la dependencia. Su función es cancelar una reserva para un día específico a futuro para un usuario de la comunidad.

Flujo de Eventos

- El sistema muestra la agenda y las reservas según los días y horas.
- El operador puede cancelar la reserva de los equipos asignados a la dependencia según la agenda sólo a futuro. Si elige cancelar la reserva para un día específico, el operador o la dependencia debe llenar el motivo de la cancelación (si aplica).

Precondiciones

- Debe de existir una reserva a futuro para un día específico.

Poscondiciones

- La reserva queda registrada en la base de datos como cancelada. .
- El equipo queda liberado para su uso para el momento que estaba reservado.
- El administrador de reserva gestiona la transacción con el manejador de base de datos.

Especificación de Caso de Uso: Modificar Reserva

Descripción

Este Caso de Uso puede ser iniciado tanto por el operador como por el usuario de la dependencia. Su función es modificar una reserva para un día específico a futuro para un usuario de la comunidad.

Flujo de Eventos

- El sistema muestra la agenda y las reservas según los días y horas.
- El operador puede modificar la reserva de los equipos asignados a la dependencia según la agenda y la disponibilidad.

Precondiciones

- Debe de existir una reserva a futuro para un día específico.

Poscondiciones

- La reserva queda registrada en la base de datos con la nueva fecha.
- El administrador de reserva gestiona la transacción con el manejador de base de datos.

Especificación de Caso de Uso: Asignar Equipos

Descripción

Este Caso de Uso lo inicia el administrador general del sistema. Su función consiste en asignar los equipos disponibles al conjunto de dependencia de la UCAB.

Flujo de Eventos

- El sistema muestra la lista detallada con los equipos disponibles en el sistema con todos sus atributos (seriales, modelos, etc.).
- El administrador debe asignar los equipos actuales según la dependencia a la cual pertenecen. Debe especificar todos los atributos que identifican al equipo así como etiquetar con la numeración de control de activos de la UCAB (en caso de no poseerla).
- Si se registra un equipo nuevo debe de venir acompañado de la correspondiente orden de compra.

Precondiciones

- Si el equipo no pertenece a ninguna de las categorías disponibles en el sistema el administrador debe clasificar y registrar la nueva categoría antes de asignar el equipo.
- Es necesario que la dependencia esté registrada en la lista de dependencias. Si la dependencia no está registrada el administrador debe de registrar la dependencia con todos los atributos específicos.
- El equipo debe de estar en perfecto estado y con todos sus cables accesorio para su asignación.

Poscondiciones

- El equipo quede asignado a la dependencia e inmediatamente esta disponible para hacer reservas. El administrador de inventario gestiona la transacción con el manejador de base de datos.

Especificación de Caso de Habilitar / Inhabilitar Equipos

Descripción

Este Caso de Uso lo inicia el administrador general del sistema. Su función consiste en desincorporar e incorporar los equipos que se encuentra en mantenimiento.

Flujo de Eventos

- El sistema muestra la lista detallada con los equipos en disponibles en el sistema.
- El administrador selecciona el equipo que desea habilitar o el que desea inhabilitar.
- Si se habilita un equipo el administrador debe registrar el mantenimiento al que fue sujeto el equipo.
- Si se deshabilita un equipo el administrador consulta las observaciones
- realizadas por el operador (si las hubiese) y procede a realizar los trámites correspondientes para su revisión o reparación.

Precondiciones

- Equipos disponibles para habilitar / deshabilitar.
- Equipos disponibles para asignación temporal a dependencias para no afectar el servicio.

Poscondiciones

- Si el equipo tiene asignadas reservas a futuro debe de asignarse un equipo de préstamo temporal a la dependencia
- El equipo quede registrado y en estado deshabilitado para su reparación.
- El sistema debe registrar el mantenimiento al cual fue sometido el equipo en caso de procesar una habilitación.
- El administrador de mantenimiento gestiona la transacción con el manejador de base de datos.

Especificación de Caso de Uso: Generar estadísticas

Descripción

Este Caso de Uso lo inicia el administrador general del sistema. Su función consiste en proporcionar una serie de reportes de gestión para ser utilizados y analizados por la coordinación general y la unidad de operaciones y servicios.

Flujo de Eventos

- El sistema muestra los distintos reportes estadísticos disponibles para consulta.
- El administrador selecciona el reporte de su elección. Debe especificar el o los criterio por lo cual desea que sea filtrada o consulta la información.
- El administrador puede exportar la información en formato separado por comas para facilitar su análisis e interpretación.

Precondiciones

- Información disponible en el almacén de datos.

Poscondiciones

- Ninguna

Estructura Desagregada del Trabajo (EDT)

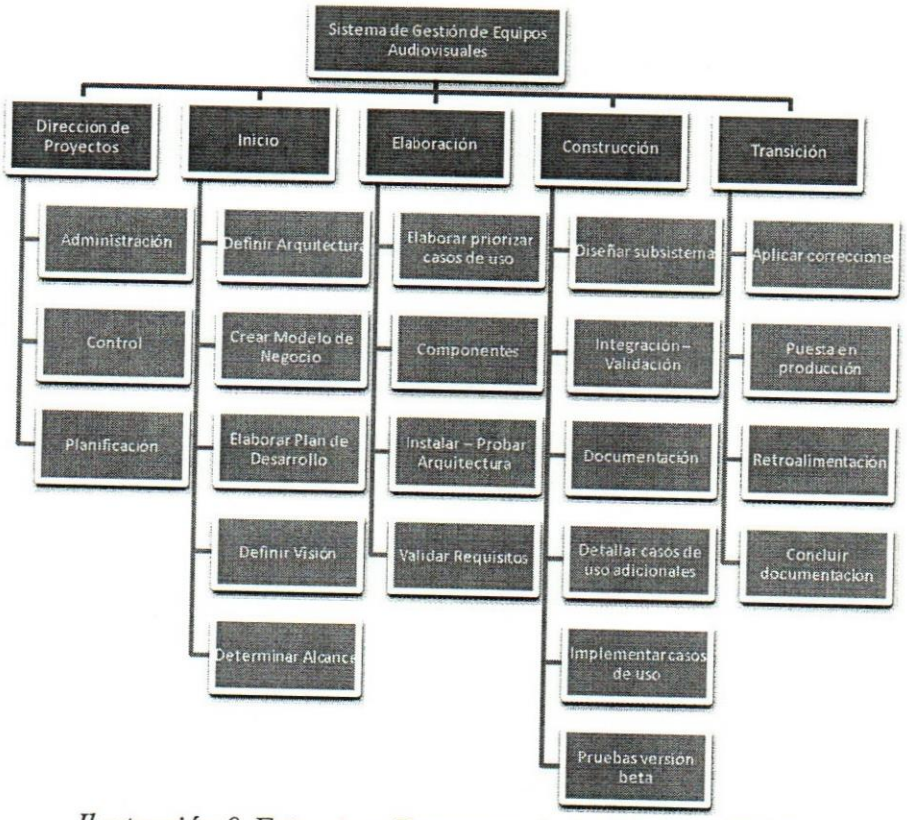


Ilustración 9. Estructura Desagregada del Trabajo (EDT)

Fuente: Elaboración propia.

Verificar Alcance

La coordinación general junto con la unidad de desarrollo se encargará de definir la plantilla de aceptación formal que describa los requisitos y criterios de aceptación del producto. La misma puede expresar el trabajo en términos de producto o actividades a cumplir dentro de una iteración.

Esta forma se modificará a través de un proceso de inspección que permita medir, explorar y comprobar si se están cumpliendo satisfactoriamente los requisitos planteados.

Control del Alcance

Los requisitos del sistema antes especificados tendrán una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una solicitud de cambio, las cual será evaluada por la coordinación conjuntamente con el cliente para determinar su factibilidad o no, luego será distribuido y documentado para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Manejo de los costos.

Estimación de los costos.

Dado que lo que se quiere es tener una primera aproximación de los costos asociados al proyecto, se utilizó el modelo COCOMO para lograr la estimación de costos, ya que el mismo que permite realizar estimaciones y planificaciones de proyectos de sistemas de información.

Para calcular el esfuerzo y la duración del proyecto se utilizó una herramienta desarrollada por el Departamento de Computación y Ciencias de la Información de la

Universidad de Michigan (CIS Department at the University of Michigan)[9] basada en el modelo de básico de estimación COCOMO.

El modelo del CIS está divido en 4 simples pasos. A continuación se presenta los distintos valores utilizados en las 4 fases para realizar las estimaciones:

- Fase 1. Se define la complejidad de los valores de dominio que rigen el desarrollo. Estos vienen dados por el número de entradas, salidas, consultas, archivos e interfaces estimadas.

Information Domain Values

Measurement Parameter	Count		Simple	Average	Complex		Total
Number of user inputs	10	X	3	4	6	=	30.00
Number of user outputs	10	X	4	5	7	=	40.00
Number of user inquiries	20	X	3	4	6	=	60.00
Number of files	2	X	7	10	15	=	14.00
Number of external interfaces	0	X	5	7	10	=	.0
Count=Total							144.00

Ilustración 10. Indicadores de la fase 1 del modelo COCOMO

Fuente: Elaboración propia.

- Fase 2. Utilizando juicio de expertos y el conocimiento que se tiene en el desarrollo de este tipo de aplicaciones, se procedió a determinar los pesos de los factores de complejidad respondiendo el siguiente cuestionario.

[9] Disponible en http://www.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/gamel/cocomo.html. Consultado el 27-10-2009

Complexity Weighting Factors
// heading of the second table Rate each factor on a scale of 0 to 5:
(0 = No influence, 1 = Incidental, 2 = Moderate, 3 = Average, 4 = Significant, 5 = Essential):

Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	●	○	○	○	○	○
2. Are data communications required?	○	○	●	○	○	○
3. Are there distributed processing functions?	●	○	○	○	○	○
4. Is performance critical?	○	○	●	○	○	○
5. Will the system run in an existing, heavily utilized operational environment?	●	○	○	○	○	○
6. Does the system require on-line data entry?	○	○	○	○	○	●
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	○	●	○	○	○
8. Are the master file updated on-line?	●	○	○	○	○	○
9. Are the inputs, outputs, files, or inquiries complex?	○	○	○	●	○	○
10. Is the internal processing complex?	○	○	○	●	○	○
11. In the code designed to be reusable?	○	○	○	○	●	○
12. Are conversion and installation included in the design?	○	○	●	○	○	○
13. Is the system designed for multiple installations in different organizations?	●	○	○	○	○	○
14. Is the application designed to facilitate change and ease of use by the user?	○	○	○	○	●	○
Total 27.00						

Ilustración 11. Pesos de los factores de complejidad, Modelo COCOMO

Fuente: Elaboración propia.

Fase 3. Se obtiene el factor LOC/FP (Líneas de Código) según el lenguaje de programación a utilizar. Para este caso se tiene la restricción de utilizar una solución LAMP lo que implica que el lenguaje de programación a utilizar es PHP el cual pertenece a la categoría de lenguajes orientados a objeto.

Programming Language	LOC/FP (average)	Select
Assembly Language	320	○
C	128	○
COBOL	105	○
Fortran	105	○
Pascal	90	○
Ada	70	○
Object-Oriented Languages	30	◉
Fourth Generation Languages (4GLs)	20	○
Code Generators	15	○
Spreadsheets	6	○
Graphical Languages (icons)	4	○

LOC/F P: [Show LOC/FP] 3974.40

Ilustración 12. Cálculo del factor LOC/FP del modelo COCOMO

Fuente: Elaboración propia.

Fase 4. Finalmente se seleccionó la complejidad del proyecto, que para este caso es del tipo orgánico ya que el proyecto puede variar según la experiencia del equipo de unas cuantas miles a pocas decenas de miles de líneas de código, lo que implica un desarrollo de mediano alcance y baja complejidad.

Software Project	a_b	b_b	c_b	d_b	Select
Organic	2.4	1.05	2.5	0.38	◉
Semi-detached	3.0	1.12	2.5	0.35	○
Embedded	3.6	1.20	2.5	0.32	○

[Calculate Effort and Duration]

Effort (E) = $a_b(KLOC)^{b_b}$ = 10.22 **Duration (D) = $c_b(E)^{d_b}$ =** 6.05

Ilustración 13. Cálculo de la complejidad del desarrollo

Fuente: Elaboración propia.

Todo estos cálculos arrojaron como resultado un esfuerzo de 10.22 personas mes con una duración de 6.05 meses lo que equivale a necesitar por lo menos de 1.68 ~ 2 personas para completar el proyecto, para el caso estudiado un desarrollador experimentado es suficiente, más un arquitecto que ayude a modelar el software y las pruebas unitarias.

Control de los Costos

Como los costos están expresados en términos de personas necesarias para completar el proyecto no es tan trivial el llevar el control de los costos, ya que no se puede comparar esta unidad contra un presupuesto base. El control debe llevarse por medio de la supervisión del trabajo, evaluación de los reportes de progreso a fin de determinar si es necesario un ajuste de los valores y factores que determinaron el esfuerzo necesario para actualizar el tiempo total.

Manejo del tiempo

Cronograma de Desarrollo

A continuación se presente el cronograma general de las actividades, expresado en meses y agrupado por las actividades expresadas en el EDT.

	Task Name	Duration
1	⊟ **Dirección de Proyectos**	**140 days**
2	Administrración	140 days
3	Control	120 days
4	Planificación	15 days
5	⊟ **Inicio**	**15 days**
6	Definir Arquitectura	5 days
7	Crear modelo de negocios	5 days
8	Elaborar plan de desarrollo	5 days
9	Definir visión	2 days
10	Determinar alcance	3 days
11	⊟ **Elaboración**	**17 days**
12	Elaborar Casos de Uso	5 days
13	Componentes	12 days
14	Instalar arquitrectura	5 days
15	Validar requisitos	13 days
16	⊟ **Construcción**	**72 days**
17	Diseñar subsistemas	30 days
18	Integración / Validación	14 days
19	Documentación	20 days
20	Detallar casos de uso adicionales	4 days
21	Implementar casos de uso	14 days
22	Pruebas versión beta	10 days
23	⊟ **Transición**	**17 days**
24	Aplicar correcciones	10 days
25	Puesta en producción	4 days
26	Retroalimentación	16 days
27	Documentación Final	6 days

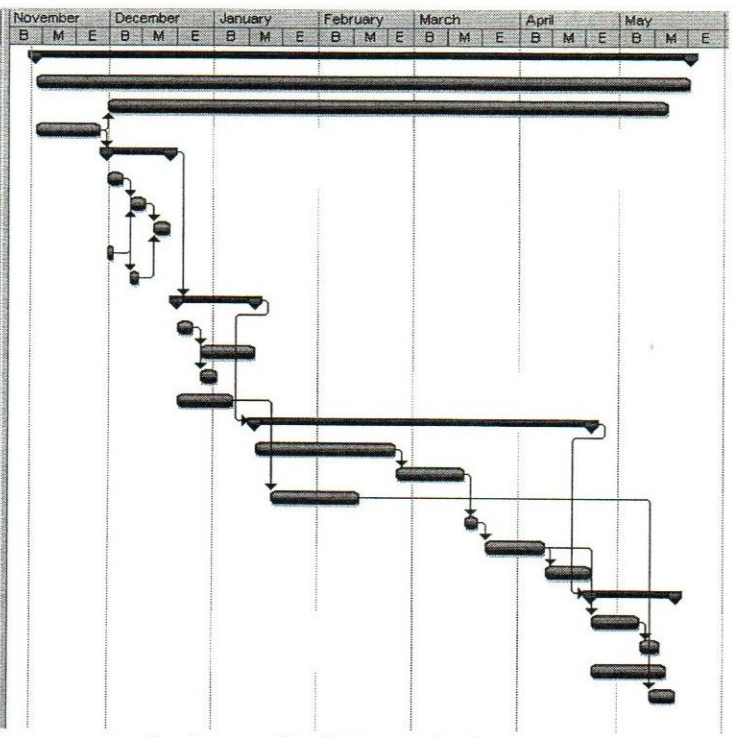


Ilustración 14. Cronograma de desarrollo de las actividades

Fuente: Elaboración propia.

La estimación que ofrece el modelo COCOMO es muy similar en tiempo a la estimación producto de las actividades del EDT.

Manejo de la calidad

Planificación de la Calidad

Se utilizarán herramientas de aseguramiento de la calidad propuestas por el W3C (World Wide Web Consortium) para la validación de los estándares a seguir en todo el desarrollo del sistema. El W3C, es una asociación internacional formada por organizaciones miembro del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web.

Aseguramiento de la Calidad

Para lograr el aseguramiento de la calidad el arquitecto de software coordina y lleva a cabo las siguientes actividades específicas:

- Inspeccionar el código fuente, tanto a nivel de programación como de interfase de usuario (HTML) y compararlo contra los estándares propuestos por el W3C.
- Realizar las pruebas funcionales y no funcionales.
- Inspeccionar la documentación generada.
- Crear los informes de avance.
- Reuniones de planificación de nuevas iteraciones.

Lo que se persigue es garantizar la calidad de los procesos y productos del sistema propuesto.

Control de la Calidad

En Control de Calidad se coordinará a través de las solicitudes de cambio de los defectos o inconsistencias de estándares detectados en las revisiones y formalizados por medio de una solicitud de cambio. Los mismos tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias.

Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y listas de verificación incluidas en RUP.

Manejo de los Riesgos

Para manejar los riesgos asociados al proyecto hay que tener en cuenta que los mismos influyen directamente en la planificación de un nuevo sistema, por tal motivo el

contenido de cada iteración debe basarse en los riesgos que han de ser identificados, transformados a Caso de Uso y mitigarlos en el propio momento. A medida que se desarrolle el trabajo individual se irá anexando los riesgos críticos a la lista de riesgos con el fin de atenuarlos.

Entre los riesgos generales más comunes identificados tenemos:

Descripción	Prioridad	Impacto	Monitor	Responsabilidad	Contingencia
Cronograma optimista	Significativo	Medio	LP — AS, D	GP	Acción 1
Desarrollar sin metodología	Rutinario	Bajo	AS - LP	AS	Acción 2
Poca retroalimentación del usuario final	Crítico	Alto- afecta al equipo de proyecto	LP	LP — D	Acción 3
Nuevos requerimientos a lo largo del proyecto	Crítico	Alto- afecta al equipo de proyecto	LP	LP — D	Acción 2
Proyecto con ciclo largos de desarrollo	Crítico	Medio	LP	LP	Acción 4

Ilustración 15. Lista de Riegos.

Leyenda: GP: Gerente de Proyecto, LP:Lider de Proyecto, D:Desarrollador, AS: Analista de Sistemas
Fuente: Elaboración propia.

Donde la descripción se refiere al riesgo en si identificado, la prioridad indica cuáles son los más importantes a tener en cuenta, el impacto indica cuan profundo pudiera verse afectado el proyecto si sucede el riesgo, monitor y responsabilidad indica quien monitorea y quien es el responsable de estar pendiente de el riesgo y la contingencia se refiere a que acción debe tomarse en caso de que se presente el riesgo.

Las acciones a seguir en caso de que se presente los riesgos:

- Acción 1: Se verificará por medio de reuniones con los miembros del equipo de proyecto así como con el cliente el avance parcial de las actividades cada 2 semanas, con el fin de monitorear el avance general del proyecto así como

tomar acciones correctivas en caso de atraso o replanificación de las actividades.

- Acción 2: Se estudiará el caso para determinar cuáles son las acciones correctivas más pertinentes (si son factibles) para mitigar el problema sin que afecte el Cronograma General del Proyecto.
- Acción 3: Se redactará un informe a la coordinación general que indique el estado actual de las relaciones de contacto con las dependencias involucradas en el proyecto.
- Acción 4: Se crearán mini proyectos esto con el fin de hacer los objetivos de cada iteración más medibles y de menor alcance.

Hay que tener presente que cada proyecto de Desarrollo de Software tiene sus propios riesgos dependiendo de su tamaño, complejidad, usuarios y entorno.

Manejo de las Comunicaciones

El Proceso Unificado agrupa a todo el equipo de Desarrollo de Software y mejora la comunicación del equipo al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado único y un punto de vista de cómo desarrollar el software, dado que la especificación de requerimientos representa una forma de consenso entre los clientes y el equipo o unidad de desarrollo. Como el Proceso Unificado es iterativo e incremental, en cada iteración se culmina con un reporte de actividades que indica los detalles, progreso general del proyecto y las proyecciones

actualizadas. Si es necesario, en esta etapa se actualizan los Casos de Uso y la documentación detallada.

En cuanto a las comunicaciones con el cliente, se elaborará un reporte de progreso quincenalmente con el fin de mantener al día los avances, presentarlos en su debido momento, así como también especificar si las proyecciones en costo tiempo siguen siendo válidas o requieren de alguna discusión / actualización.

Además debe de tenerse en cuenta que debe formar parte del plan de comunicaciones las siguientes actividades:

- Reuniones semanales de tecnología donde se platean temas relacionados con software, hardware y procesos y la integración entre estas áreas.
- Reuniones internas del equipo de desarrollo para los cierres e inicios de iteración de manera que todos conozcan los objetivos a lograr primero, y los resultados del trabajo del período anterior después.
- Reuniones internas con el responsable del aseguramiento de la calidad para realizar la coordinación y seguimiento de actividades.

Manejo de la Integración

Dado que RUP está dirigido por los Casos de Uso, todas las fases se integran por medio de los Casos de Uso. El simple hecho de ser un proceso iterativo permite un entendimiento incremental del problema a través de refinamientos sucesivos, por lo que cada iteración en si conlleva a una mejor integración ya que por un lado permite una fácil retroalimentación además de que el progreso es medido conforme avanzan la implementación.

Manejo del Recurso Humano.

El recurso humano involucrado dentro del proyecto forma parte del DTI de forma permanente por lo que no se hace necesario la contratación de nuevo personal para este proyecto.

Según el cálculo del modelo COCOMO se requieren de dos programadores para realizar el trabajo planificado. Se seleccionarán dos miembros del stock de programadores y se le asignarán las responsabilidades individuales y como equipo.

La gestión del equipo determinará el rendimiento individual y del grupo en general según las actividades programadas, y a su vez permitirá utilizar esta información como parte de la evaluación de desempeño anual a las que son sometidos todos los miembros del departamento. Se definirá en coordinación con la gerencia un formulario de reporte diario de actividades como mecanismo de control, lo que permitirá recibir una retroalimentación para establecer acciones en caso de posibles cambios que tenga que introducirse dentro de la planificación.

Manejo de las Adquisiciones

Dado la naturaleza del proyecto (Desarrollo de Software en ambiente Web) y las condiciones del ambiente en donde se aplicará el modelo, es decir, dentro de una unidad de un departamento perteneciente a una estructura organizativa bien definida y delimitada en cuanto a sus funciones, no se requiere de ningún tipo de adquisición para el desarrollo. Todas las herramientas a ser utilizadas por restricción deben operar bajo el enfoque de arquitectura LAMP, la cual se componen de tecnología de software libre en

su totalidad. Por otro lado se aprovecha toda la infraestructura tecnológica que tiene la UCAB para la ejecución del proyecto.

Capítulo – 5. Conclusiones y Recomendaciones

El siguiente capítulo expone las reflexiones y conclusiones producto del trabajo de investigación, así las sugerencias o recomendaciones por parte del autor para futuros proyectos.

Conclusiones

El presente trabajo de investigación logró esquematizar los elementos constitutivos de un Plan de Ejecución práctico según la Guía de Fundamentos para la Gerencia de Proyectos aplicada al Desarrollo de un Sistema de Software. Así mismo se definieron los modelos que define el Lenguaje Unificado de Modelado a ser aplicados según la metodología del Proceso Unificado a ser implantados para el desarrollo de un sistema en su fase de inicio.

Por otro lado, esta investigación permitió crear un modelo práctico de un Plan de Ejecución que incluye procesos y herramientas, tanto de la Gerencia de Proyectos como del Proceso Unificado de Desarrollo de Software, los cuales definen el "Modelo de Negocio" para el desarrollo de un Sistema de Gestión de Equipos Audiovisuales. Los procesos y las herramientas pudieron ser mezclados de manera tal que se logró la definición del sistema propuesto desde un punto de vista más gerencial, lo cual permite al Gerente de Proyecto guiar la ejecución a través del ciclo de vida para lograr cumplir las metas propuestas.

Se pudo comprobar que en ambas metodologías se hace difícil estimar tiempo y costo dada la naturaleza misma de los proyectos de Desarrollo de Software, por lo que se utilizó un modelo de estimación propuesto por el CIS que permitió comprobar que las

estimaciones de actividades según la Estructura Desagregada del Trabajo (EDT) y las estimaciones de tiempo del modelo COCOMO arrojan cifras similares, por lo que se utilizó dicho modelo para las estimaciones necesarias.

Es importante señalar, que dicho modelo es de utilidad siempre y cuando el estimador tenga cierto grado de conocimiento en el Desarrollo de Software, ya que los índices y variables necesarios para el cálculo de las estimaciones requieren que se conozca ciertos aspectos tecnológicos así como aspectos muy técnicos sobre el Desarrollo de Software, tanto a nivel de teoría como práctica.

Finalmente, y como conclusión general, la creación de un Plan de Ejecución en la Gerencia de Proyecto, está muy ligado a la naturaleza del proyecto en sí y al área específica de aplicación. Sin embargo, en la investigación se pudo comprobar que existen Planes de Ejecución muy similares en áreas de aplicaciones distintas.

Recomendaciones

Se sugiere que para futuros proyectos basados en la metodología RUP se adopte más de una iteración en todas las fases esto con la finalidad de que se pueda describir la ingeniería de detalle de una manera más profunda y desarrollar todos los modelos que permiten describir el sistema desde todos sus puntos de vista.

Al mismo tiempo es importante sugerir para futuros proyectos pos su utilidad implementar en paralelo un Sistema o Módulo de Control de Confiabilidad del Mantenimiento y de los Equipos, que ayude a predecir las fallas antes de que ocurran, solicitar los repuestos justo cuando serán requeridos, entre otras funciones, esto con miras a garantizar la máxima confiabilidad y el mayor ahorro de costos posibles.

Referencias Bibliográficas

American Psychological Association. (2002). *Publication Manual of the American Psychological Association* [Manual de Publicación de la Asociación Americana de Psicología] (5th ed.). Washington, DC: Autor.

Balestrini Acuña, M. (2001). *Como se elabora el proyecto de investigación.* Caracas: BL Consultores Asociados.

Boehm, B., Abts, C., & Winsor, A. (2000). *Software Cost Estimation with Cocomo II* [Estimación del Costo del Software con COCOMOII] (1st ed.). Upper Saddle River,NJ: Prentice-Hall, Inc.

Departament of Premier and Cabinet, Tasmania. (2008). *PM005 Project execution Plan, template & guide*.[Plan de Ejecución de Proyecto, Plantilla y Guía] Recuperado el 27 de Octubre de 2009, de http://www.peacecorps.ro/Resources/Categories_Topic/Business_Resource_Center/Project Management/Project Execution Templates/Project Executive Plan PM 005.doc

El Modelo de COCOMO. (16 de Enero de 2009). Recuperado el 27 de Octubre de 2009, de UNAM CAMPUS ACATLAN: http://www.galeon.com/analisis_de_sistemas/

Hernández, R., Fernández, C., & Batista, P. (2003). *Metodología de la Investigación.* México: McGraw Hill Interamericana.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software.* Madrid: Pearson Educación, S.A.

Montilva, B. (2006). *Modelado de Sistemas Usando UML 2.0.* Mérida: CEISOFT.

Palacios, L. E. (2004). *Principios Esenciales para realizar proyectos. Un enfoque latino* (3ra. ed.). Caracas: Publicaciones UCAB.

Project Management Institute Inc. (2008). *Project Managment Body of Knowledge (PMBOK)* [Fundamentos para la Dirección de Proyetos] (4th ed.). Newtown Square: Autor.

Proyecto de Desarrollo de Software. (s.f.). Recuperado el 27 de Octubre de 2009, de Ejemplo de desarrollo software utilizando la metodología RUP: http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/index.html

Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid, España: Pearson Educación, S.A.

Santalla Peñaloza, Z. d. (2003). *Guía para la elaboración de reportes de investigación.* Caracas: Publicaciones UCAB.

Universidad Pedagógica Experimental Libertador. (2005). *Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales* (3ra. ed.). Caracas: Autor.