

**PROPUESTA GERENCIAL PARA LA ADMINISTRACIÓN DE PROYECTOS DE
DESARROLLO DE SOFTWARE EN INSTITUCIONES DE EDUCACIÓN
SUPERIOR**

INGENIERO GUILLERMO AUGUSTO NARVÁEZ BURBANO

**UNIVERSIDAD DE NARIÑO
FACULTAD DE CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS
ESPECIALIZACIÓN EN ALTA GERENCIA
SAN JUAN DE PASTO
AGOSTO DE 2008**

**PROPUESTA GERENCIAL PARA LA ADMINISTRACIÓN DE PROYECTOS DE
DESARROLLO DE SOFTWARE EN INSTITUCIONES DE EDUCACIÓN
SUPERIOR**

GUILLERMO AUGUSTO NARVÁEZ BURBANO

**Trabajo de grado presentado como requisito parcial para optar el título de
Especialista en Alta Gerencia**

**Mg. Guillermo A. Narváez Ramírez
Asesor:**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS
ESPECIALIZACIÓN EN ALTA GERENCIA
SAN JUAN DE PASTO
AGOSTO DE 2008**

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en este trabajo son de responsabilidad exclusiva del autor”

Artículo 1 del acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN

Jurado Presidente

Jurado

Jurado

San Juan de Pasto, Septiembre de 2008

CONTENIDO

	Pág.
INTRODUCCIÓN	11
1. MARCO GENERAL DE REFERENCIA.....	13
1.1 EL PROBLEMA DE INVESTIGACIÓN.....	13
1.1.1 Planteamiento	13
1.1.2 Formulación del problema.....	14
1.2 OBJETIVOS DEL TRABAJO DE GRADO	14
1.2.1 Objetivo general.....	14
1.2.2 Objetivos específicos:	14
1.3 MARCO TEÓRICO	15
1.3.1 Gerencia, organizaciones y proyectos	15
1.3.2 Generalidades de la gerencia de proyectos.....	17
1.3.3 El desarrollo de software.....	20
1.3.4 El software educativo	26
1.3.5 La gerencia en los dos contextos.....	30
1.4 MARCO CONCEPTUAL	33
1.5 MARCO CONTEXTUAL.....	37
1.6 ASPECTOS METODOLÓGICOS	37
2. COMPONENTES FUNDAMENTALES DE LA PROPUESTA GERENCIAL	38
2.1 EL MARCO DE LA GERENCIA DE PROYECTOS.....	38
2.1.1 Los elementos básicos.....	38
2.1.2 Sumario de las áreas básicas de la administración de proyectos.....	39
2.2 ASPECTOS PRIMORDIALES	43
2.2.1 Fases y ciclo de vida del proyecto	43
2.2.2 Características de las fases del proyecto.....	44
2.2.3 Características del ciclo de vida del proyecto	44
2.3 EL DESARROLLO DE PROYECTOS DE SOFTWARE	45

	Pág.
2.3.1 Punto de partida: la WBS.....	45
2.3.2 Esquema de una propuesta de modelo.	47
2.3.3 Esquemas esenciales del proceso.....	48
2.3.4 Partidos interesados en el proyecto.....	56
2.3.5 Un caso típico universitario: el software educativo	56
2.3.6 Propuesta básica de un software educativo.....	60
2.4 LA DIRECCIÓN INTEGRADA DE PROYECTOS EN RELACIÓN A LA CALIDAD CON EL USO DEL SOPORTE INFORMÁTICO PROJECT 2002.	66
2.4.1 Conceptos necesarios.....	66
2.4.2 Diseño de la calidad.....	67
2.4.3 Control de ejecución de la calidad.	69
2.4.4 El proceso de capacitación de los especialistas	72
2.5 DESARROLLO DE LA PROGRAMACIÓN	73
2.5.1 Elementos claves.....	73
2.5.2 Herramientas y técnicas para el desarrollo de la programación	74
2.5.3 Salidas del desarrollo de la programación	75
3. PROPUESTA DE MODELOS DE DESARROLLO DE SOFTWARE	77
3.1 ELEMENTOS DE GERENCIA Y ADMINISTRACIÓN DEL PROYECTO DE SOFTWARE.....	77
3.1.1 El sistema de información.....	77
3.1.2 Planificación del proyecto de software.....	79
3.1.3 Estimación del proyecto.....	80
3.1.4 Diferentes modelos de estimación	80
3.2 ANÁLISIS DE LOS SISTEMAS DE COMPUTACIÓN.....	82
3.2.1 Premisas.....	82
3.2.2 Objetivos del análisis.....	83
3.3 DISEÑO DE LOS SISTEMAS DE COMPUTACIÓN	84
3.3.1 Conceptos y principios.....	84

	Pág.
3.3.2 Diseño de la salida.	85
3.3.3 Diseño de archivos	85
3.3.4 Implantación, evaluación y pruebas	86
3.4 ANÁLISIS DE REQUERIMIENTOS	86
3.4.1 Fundamentos.....	86
3.4.2 Particularidades	88
3.4.3 Tareas del análisis	88
4. PROPUESTA DE UN MODELO DE PROYECTO DE SOFTWARE	91
4.1 PRESENTACIÓN.....	91
4.2 EL MODELO GERENCIAL BÁSICO	91
4.3 MODELO DE UN SOFTWARE EDUCATIVO	107
5. CONCLUSIONES Y RECOMENDACIONES	113
5.1 CONCLUSIONES	113
5.2 RECOMENDACIONES	114
BIBLIOGRAFÍA.....	115

LISTA DE FIGURAS

	Pág.
Figura 1. Fases de la Gerencia de proyectos	43
Figura 2. Ciclo de vida de desarrollo de software	48
Figura 3. Vista general de la administración de la integración del proyecto	49
Figura 4. Vista general de la administración del tiempo del proyecto	50
Figura 5. Vista general de la administración de costos del proyecto	51
Figura 6. Vista general de la administración de calidad del proyecto	52
Figura 7. Vista general de la administración de las comunicaciones del	53
Figura 8. Vista general de la administración de riesgo del proyecto	54
Figura 9. Vista general de la administración del procuramiento del proyecto	55
Figura 10.....	78
Figura 11. Esquema del análisis de requerimientos	87
Figura 12. Esquema de la propuesta básica:.....	92
Figura 13.....	109

RESUMEN

El trabajo de grado se dedicó al estudio de la gerencia en relación a proyectos de desarrollo de software, que puedan generar las instituciones de Educación superior. Sin embargo la propuesta elaborada no se limita a ellas, y bien puede extenderse a cualquier tipo de empresa. Como punto de partida se tuvo en cuenta que a pesar de la creciente participación del software en el mundo actual y de los avances producidos, su proceso aun no es adecuado, de tal manera que gran cantidad de proyectos de software no llegan a cumplir sus objetivos, y como consecuencia de esto, los altos porcentajes de rechazo entre ellos.

Las principales causas de estos problemas son la administración insuficiente de requisitos, comunicación ambigua e imprecisa, inconsistencias no detectadas entre requerimientos, diseño y programación, validaciones tardías de los requisitos, enfrentamiento tardío de riesgos y propagación de cambios sin control. Frente a esto el estudio realizado prueba que es posible hallar respuesta a la dicotomía que existe entre el desarrollo de un proyecto de software en su parte técnica, su ingeniería, y la gerencia que se debe ejercer en la empresa para hacerlo real, de ejecutarlo, de administrarlo eficientemente, garantizando calidad, permanencia, competitividad y éxito en todos los campos. En el mismo sentido el trabajo comprueba que existen caminos gerenciales para administrar óptimamente el desarrollo de los proyectos de software en organizaciones que actualmente, junto a las empresas comerciales, están especialmente dedicadas a ese propósito, como son las Universidades y demás Instituciones de Educación Superior. Su base es demostrar que las metodologías de desarrollo tecnológico y los modelos de gestión de proyectos no son excluyentes, sino, todo lo contrario, complementarios.

El trabajo concreto finaliza por tanto con el diseño de los procesos componentes de un modelo gerencial para la eficiente administración de proyectos de desarrollo de software.

ABSTRACT

The work of degree was dedicated to the study of the management in relation to projects of development of software, which can generate the institutions of Superior Education. Nevertheless the made proposal is not limited to that, and it can well extend to any type of company. As departure point considered that even though of the growing participation of software in the present world and of the produced advances, its process not yet is adapted, in such a way that great amount of software projects does not get to fulfill their objectives, and as a result of this, the high percentage by reject among them.

The main causes of these problems are the insufficient administration of requirements, ambiguous and vague communication, inconsistencies nondetected between requirements, design and programming, delayed validations of the requirements, delayed confrontation of risks and propagation of changes without control. Against this the realised study proves that is possible to find answer to the dichotomy that exists between the development of a project of software in its technical part, its engineering, and the management that is due to exert in the company to make it real, to execute it, to administer it efficiently, guaranteeing quality, permanence, competitiveness and success in all the fields. In the same sense the work verifies that managerial ways exist to optimally administer the development of the projects of software in organizations who at the moment, next to the commercial companies, especially are dedicated to that intention, as they are the Universities and other Institutions of Education Superior. Its base is to demonstrate that the methodologies of technological development and the models of project management are not excluding, but, quite the opposite, complementary.

The concrete work finalizes therefore with the design of the component processes of a managerial model for the efficient project management of development of software.

INTRODUCCIÓN

Este trabajo de grado se dedicó al estudio de la gerencia en relación a proyectos de desarrollo de software. Se refiere así a temas administrativos y gerenciales, a programas y desarrollos informáticos, en el marco de entidades que deberían contextualizarlos en forma adecuada, como son las instituciones de Educación superior. Sin embargo la propuesta elaborada no se limita a ellas, y bien puede extenderse a cualquier tipo de empresa, solo que las Universidades son el campo de acción de quien realizó el trabajo.

Al respecto es importante destacar que en la organización moderna lo fundamental es el sistema de administración. Este constituye el conjunto de conceptos y técnicas que aplica la misma para planificar sus objetivos, asignar recursos y actividades, y controlar resultados. La organización y la administración son términos interrelacionados aunque conceptual y prácticamente denotan objetivos diferentes. Las organizaciones son entes sociales que constituyen una porción bastante significativa del objeto sobre el cual se aplica la administración. Así mismo la gestión empresarial es la principal actividad que marca una diferencia en el grado en que las organizaciones sirven a las personas que afectan. Y el éxito que puede tener la organización al alcanzar sus objetivos y también al satisfacer sus obligaciones sociales depende, en gran medida, del desempeño gerencial de la organización. Se puede afirmar, entonces, que la gerencia es responsable del éxito o fracaso e indica por qué es necesaria la gestión, sobre la base de la medida de la eficiencia y la eficacia que ésta tenga para lograr las metas de la organización. En torno a ello se recalca que ningún grado de eficiencia puede compensar la falta de eficacia, de manera tal que, un gerente ineficaz no puede alcanzar las metas de la organización. La eficacia es la clave del éxito de las organizaciones. Pero el ser gerente no sólo es dirigir actividades; también implica ser un buen líder; es saber el proceso de cómo penetrar en las actividades que realizan los miembros del grupo con el cual se trabaja. El gerente en el logro de sus objetivos debe saber como usar las diferentes formas del poder para influir en la conducta de sus seguidores, en distintas formas, sin olvidar que es lo que se quiere lograr y hacia donde va. Esto se aplica a la vez a los grupos directivos de las Instituciones de Educación Superior.

Lo anteriormente expuesto es especialmente aplicable en la actualidad al caso de la Gerencia de Proyectos, si tiene en cuenta que todo proyecto se puede a la vez definir como una empresa. Y en este campo, los proyectos de desarrollo de software, los más modernos y los que están moviendo el universo técnico, deben ser necesariamente abordados desde el ámbito gerencial.

En ese escenario es importante caer en cuenta que necesariamente desde el ámbito técnico de la Ingeniería de software, los proyectos deben cumplir los requerimientos universales, ser eficientes. Sin embargo, su talón de Aquiles está en el campo de la Administración y la Gerencia, porque no se aplican los procedimientos y sistemas organizacionales modernos, por lo cual no necesariamente son exitosos en torno a la relación Costo/beneficio. Pero es más preocupante que tengan corta existencia o que fracasen. Para que ello no se de el camino adecuado puede ser el combinar las pautas de una moderna gerencia de proyectos, de acuerdo a los modelos de gestión, con los paradigmas de desarrollo de software.

Sobre esta base, este trabajo de grado desarrolla una propuesta gerencial para la administración de proyectos de desarrollo de software, especialmente enfocados desde organizaciones como las Universidades, pero que se pueden aplicar a cualquier tipo de empresa.

1. MARCO GENERAL DE REFERENCIA

1.1 EL PROBLEMA DE INVESTIGACIÓN

1.1.1 Planteamiento. Desarrollar el software para un banco destinado a que un cliente efectúe una transferencia de dinero por Internet, tiene la misma complejidad en Estados Unidos, Francia, Corea, Japón o Colombia. La diferencia Radica en la relación Costo / Beneficio. Mientras que el costo en los primeros podría distribuirse entre varios millones de posibles clientes del software, titulares por supuesto de miles de millones de dólares, en los países en vías de desarrollo el mismo costo deberá distribuirse entre menos usuarios con menos dinero.

Por otra parte, el avance tecnológico en hardware, software de base y paradigmas de desarrollo, también provienen de los países industrializados lo cual presiona al resto del mundo a desarrollar software bajo herramientas y modelos que para ser creados no han tenido en cuenta las características económicas, académicas y tecnológicas de países como el nuestro, obligándonos a destinar recursos desproporcionadamente elevados continuamente o arriesgarse a perder el tren tecnológico que cada día va mas rápido.

La globalización también presiona desde el ámbito de los negocios para utilizar modelos y procesos *mas evolucionados*. En este contexto, el peor error que podría cometerse es dejarse encandilar por las luces tecnológicas y tratar de aplicar recetas seguramente exitosas en otras latitudes, pero que no necesariamente garantizan el mismo resultado en un entorno de recursos limitados y exigencias impostergables de calidad. Encontrar una solución no es sencillo. Por ejemplo no es posible implementar una tienda virtual si no se está en condiciones de contar con un software con el necesario nivel de calidad, y mucho menos sin estar seguros que tan viable es como producto comercial o si los posibles clientes están en capacidad, tanto tecnológica como cognitiva, de utilizar dicha herramienta.

El camino adecuado a seguir parece ser alternativas intermedias, basadas en la evolución tecnológica que nos llega de los países muy avanzados en el desarrollo de proyectos de software, pero complementada con modelos de gestión; paradigmas de desarrollo de software, diseños y desarrollos que respondan a las reales necesidades tecnológicas, administrativas y económicas del país, las empresas desarrolladoras de software y todos quienes necesiten una guía para como llevar a cabo un proyecto tecnológico.

Pensar entonces en fusionar algunas técnicas o características de la administración de proyectos con las pautas de la ingeniería de software, para crear una guía o modelo gestión de proyectos de software no sonaba

descabellado. La idea no fue desechar los paradigmas ni lo planteado en cuanto a las diferentes técnicas de desarrollo de software, ya que la elección de estas depende del tipo de proyecto, empresa, requerimientos, etc. Lo que se pretendió fue envolver todo este desarrollo técnico en un modelo gerencial que maneje el desarrollo del software como un producto en el cual se tenga en cuenta su viabilidad, la gente que trabaja en su creación, el dinero invertido, el tiempo de cada etapa, etc. Es decir que administre el proceso del desarrollo de un proyecto tecnológico como lo es la creación de software.

En ese contexto, el problema de este trabajo reside en la búsqueda de respuestas a la dicotomía que existe entre el desarrollo de un proyecto de software en su parte técnica, su ingeniería, y la gerencia que se debe ejercer en la empresa de hacerlo real, de ejecutarlo, de administrarlo eficientemente, garantizando calidad, permanencia, competitividad y éxito en todos los campos. Se trató así de buscar un camino gerencial para administrar óptimamente el desarrollo de los proyectos de software en organizaciones que actualmente, junto a las empresas comerciales, están especialmente dedicadas a ese propósito, como son las Universidades y demás Instituciones de Educación Superior.

1.1.2 Formulación del problema. ¿Como puede ser la actividad gerencial para la lograr una eficiente administración de proyectos de desarrollo de software en Instituciones de Educación Superior?

1.2 OBJETIVOS DEL TRABAJO DE GRADO

1.2.1 Objetivo general. Elaborar una propuesta gerencial para la eficiente administración de proyectos de desarrollo de software en Instituciones de Educación Superior.

1.2.2 Objetivos específicos:

- ✓ Describir el ámbito de los proyectos de desarrollo de software.
- ✓ Desde el campo de la Gerencia de proyectos enlazar las teorías administrativas y el componente técnico de la Ingeniería del software.
- ✓ Diseñar los procesos componentes de un modelo gerencial para la eficiente administración de proyectos de desarrollo de software en Instituciones de Educación Superior y otras entidades.

1.3 MARCO TEÓRICO

1.3.1 Gerencia, organizaciones y proyectos. Las organizaciones trabajan. El trabajo generalmente involucra operaciones o proyectos, aunque los dos se pueden traslapar. Las operaciones y los proyectos comparten muchas características; por ejemplo, ellas son desarrolladas por personas, limitadas por recursos escasos y son planeadas, ejecutadas, y controladas. Por otra parte “el Proceso Administrativo o de Dirección es un conjunto de funciones básicas que deben ser realizadas por una administración en el funcionamiento, manejo y control de organizaciones o empresas”.¹

Desde este ámbito Crosby² define a la gerencia como “el arte de hacer que las cosas ocurran”. Por su parte Krygier³ la define como un cuerpo de conocimientos aplicables a la dirección efectiva de una organización. Al respecto, en la actualidad, existe consenso, entre muchos autores, al señalar que el término gerencia se puede definir como un proceso que implica la coordinación de todos los recursos disponibles en una organización (humanos, físicos, tecnológicos, financieros), para que a través de los procesos de: planificación, organización, dirección y control se logren objetivos previamente establecidos.

Chung y Megginson señalan que hay dos maneras de analizar el trabajo gerencial, desde dos puntos de vista. *Normativo*, porque especifica lo que se espera que haga el gerente y que se ha vinculado a las clásicas funciones de la administración señaladas por Farol, a saber: planificación, organización, coordinación y control. *Descriptivo*, pues desde este punto de vista el trabajo gerencial se centra en las actividades que un gerente ejecuta, las cuales se pueden resumir en cuatro categorías: “(a) personales: distribuir su tiempo, desarrollar su propia carrera, manejar sus asuntos; (b) de interacción, llamadas también directivas; (c) actividades administrativas como el procesamiento de papeles y documentos, evaluación de políticas y procedimientos y la preparación y administración del presupuesto; (d) técnicas, que envuelven el uso de herramientas y la ejecución de habilidades técnicas para la solución de problemas”.⁴

¹ DE MARTÍNEZ, JULIÁN. *Teoría General de la Administración*, ESAP, Modelo UAD, Bogotá, 1990, Pág. 110.

² CROSBY, P. *Dinámica gerencial*. México. McGraw Hill, 1988.

³ KRYGIER, A. Consultores de Gerencia. *¿Terapeutas de las organizaciones?* Revista Gerente. Mayo, 1988.

⁴ CHUNG, K. y MEGGINSON, L. *Organizational behavior*. New York: Harper and Row, publishers, 1991.

El proceso de dirección se lleva a cabo por la Gerencia y en la época actual existen diversos paradigmas para ejecutar una exitosa administración de organizaciones, empresas y proyectos. En general los modelos gerenciales giran alrededor de ocho grandes modelos, como son: Dirección por Objetivos, por Resultados, por Sistemas, por Excepción, de Dirección Dinámica, Contingencia, por Tareas y por Funciones. “Pero la alta gerencia, estudiada y diseñada en la época contemporánea, compendia y supera tales modelos a través de un estilo de gerencia estratégica, que parte de la Planeación Estratégica.”⁵

En ese contexto se debe tener en cuenta que “la administración estratégica es un concepto, una mentalidad que puede aplicarse en cualquier medio y en cualquier etapa de desarrollo si la institución está preparada para abordar asuntos fundamentales concernientes a la finalidad, objetivos, orientación, recursos, competencia, desempeño y eficacia. “Si existe la determinación para afrontar estos problemas cruciales, siempre es posible encontrar la oportunidad y el método apropiado aun en instituciones jóvenes y con menos experiencia”, según afirma Kubr Milán”.⁶

Especificando la temática se debe anotar que las operaciones y los proyectos difieren principalmente en que aquellas son sucesivas y repetitivas mientras que estos son temporales y únicos. Un proyecto, por lo tanto, puede ser definido en término de sus características distintivas. Es así una tarea temporal desarrollada para crear un producto o servicio único. Temporal quiere decir que cada proyecto tiene un comienzo definitivo y una terminación definitiva; único que el producto o servicio es diferente de alguna manera distintiva de todos los proyectos o servicios similares. Adicionalmente, el término temporal no se aplica generalmente al producto o servicio creado por el producto. Muchos proyectos son desarrollados para crear un resultado duradero. Por ejemplo, un proyecto para crear un monumento nacional creará un resultado que se espera dure por varios siglos.

Los proyectos son desarrollados en todos los niveles de la organización. Estos pueden involucrar a una sola persona o a muchas miles. Y pueden requerir menos de 100 horas para completarse o más de 10.000.000. Los proyectos pueden involucrar a una sola unidad de una organización o cruzar muchas fronteras organizacionales como en consorcios o sociedades de hecho. Los proyectos son muchas veces componentes críticos de la estrategia de negocios de la organización que los desarrolla. Ejemplos de proyectos pueden incluir el desarrollar un nuevo producto o servicio; efectuar un cambio de estructura, de personal, o de estilo en una organización; desarrollar un nuevo vehículo de transporte; desarrollar o adquirir un nuevo sistema de información; construir o

⁵ STONER, James. *Administración*. Editorial Prentice Hall, México, 1992, p. p.15-28

⁶ EDUCAR. *Clásicos de la Administración*. Bogotá, 1980, p. p 57 – 60

desarrollar una construcción; administrar una campaña electoral; desarrollar un software.

Los proyectos involucran hacer algo que no se ha hecho antes; por lo tanto, es único. Un producto o un servicio puede ser único aunque la categoría a la que pertenezca sea grande. Por ejemplo, muchos miles de edificios de oficina han sido desarrollados, pero cada edificio en si es único — de distinto dueño, de distinto diseño, diferente locación, y diferentes contratistas, y así etc. La presencia de elementos repetitivos no cambia fundamentalmente la característica de ser único. Debido a que el producto de cada proyecto es único, las características que distinguen el producto o servicio deben ser elaboradas progresivamente. Esto significa “Procedimientos en pasos; avance continuo por incrementos”, mientras que elaborados quiere decir “trabajado con cuidado al detalle; desarrollado enteramente”⁷.

1.3.2 Generalidades de la gerencia de proyectos. En la gerencia de todo proyecto se existen elementos diferenciales que se especifican a continuación.

Administración del Alcance del Proyecto: Describe el proceso requerido para asegurar que el proyecto incluye todo trabajo requerido, y sólo el trabajo requerido, para completar el proyecto de manera exitosa. Consiste de la iniciación, planeación del alcance, definición del alcance, verificación del alcance, y control de cambio al alcance.

Administración del Tiempo del Proyecto: Describe los procesos requeridos para asegurar la terminación a tiempo del proyecto. Consiste en la definición de las actividades, secuencia de las actividades, estimación de duración de las actividades, desarrollo del cronograma y control de la programación.

Administración de los Costos del Proyecto: Describe los procesos requeridos para asegurar que el proyecto es completado dentro del presupuesto aprobado. Consiste en la planificación de recursos, estimación de costos, presupuestación de costos, y control de costos.

Administración de la Calidad del Proyecto: Describe los procesos requeridos para asegurar que el proyecto satisfará las necesidades para lo cual fue desarrollado. Consiste en la planeación de la calidad, aseguramiento de la calidad, y control de calidad.

Administración de los Recursos Humanos del Proyecto: Describe los procesos requeridos para hacer el uso más eficiente de las personas involucradas en el

⁷The American Heritage Dictionary of the English Language, Third Edition. Boston, Mass.:Houghton Mifflin Company. 1992.

proyecto. Consiste en la planeación organizacional, adquisición de staff, y desarrollo del equipo.

Administración de las Comunicaciones del Proyecto: Describe los procesos requeridos para asegurar la generación apropiada y a tiempo, colección, diseminación, almacenamiento, y la disposición final de la información del proyecto. Consiste en la planeación de la comunicación, distribución de la información, reportes de desempeño, y el cierre administrativo.

Administración de Riesgo del Proyecto: Describe los procesos concernientes con la identificación, análisis, y respuesta a el riesgo del proyecto. Consiste en la identificación del riesgo, cuantificación del riesgo, desarrollo de la respuesta al riesgo, y en el control de la respuesta al riesgo.

Administración de la Procuración del Proyecto: Describe los procesos requeridos para adquirir bienes y servicios de fuera de la organización ejecutora. Consiste en la planeación de la gestión de la procuración, planear la solicitud, la solicitud, selección de proveedores, administración de contratos, y cierre de contratos.

Por otra parte, la Gerencia de proyectos (Project management) es la disciplina de organizar y administrar los recursos, de forma tal que un proyecto dado sea terminado completamente dentro de las restricciones de alcance, tiempo y coste planteados a su inicio. “Dada la naturaleza única de un proyecto, en contraste con los procesos u operaciones de una organización, administrar un proyecto requiere de una filosofía distinta, así como de habilidades y competencias específicas. De allí la necesidad de la disciplina Gerencia de Proyectos”⁸. Así mismo, la Dirección Integrada de Proyectos o Project Management, tiene como objetivo principal la obtención de los resultados programados, acortando los plazos de tiempo, en el marco del presupuesto y con la calidad requerida. De estos tres elementos, el tiempo es el que con más frecuencia se controla en la programación. El costo generalmente se tiene en cuenta en la primera etapa con la elaboración del presupuesto y en la etapa de ejecución se debe controlar a través de un sistema de centro de costos por proyecto eficiente y una contabilidad confiable en los cortes. La calidad en el proceso de ejecución, tiene una incidencia determinante tanto en el costo como en el tiempo,

Sobre estas bases se comprende que un proyecto debe gerenciarse desde dos enfoques congruentes, máxime cuando se trata de desarrollos de software. Por una parte el área administrativa propiamente dicha, con los elementos de las teorías organizacionales; por otra, la del Project Management.

Sobre el tema de la gerencia de proyectos, como una realidad actual de la administración, la guía que sigue este proyecto tiene su base en la opinión Alan

⁸ LEVINE, Harvey. *Gerencia práctica de proyectos*. Mac Graw Hill, 2006

Randolph y Barry Z. Posner, quienes plantean que “para hacer frente a este nuevo reto gerencial, debemos empezar por vernos como líderes de proyectos, de equipos de trabajo, y no como los tradicionales gerentes funcionales”. Desde el empleado de producción hasta el analista financiero; desde el banquero hasta quien desarrolla programas de computación; desde el ingeniero hasta el administrador, todos debemos empezar a pensar y a trabajar con un enfoque centrado en la innovación y en los equipos de gente organizados (aunque a veces sea temporalmente) para un fin específico. Debemos empezar a pensar en el trabajo como la acumulación de proyectos con características comunes, pero cuyo enfoque particular es único.”⁹

Por otra parte, la Dirección Estratégica Integrada de la empresa, en su proceso de perfeccionamiento, requiere de la integración de la Dirección Integrada de Proyectos (DIP) y la Dirección Integrada de la Calidad (DIC) con el objetivo de lograr el éxito de los proyectos. La necesidad de integración de la DIP y la DIC en un sistema informático único se logra a partir de la programación estructurada de tareas y subproyectos que brinda el MS Project con el objetivo de garantizar la evaluación por cortes de los proyectos en función de la toma de decisiones de la gerencia empresarial haciendo uso del tablero de comandos.

“La Dirección Integrada de la Calidad (DIC) y su vinculación con la ejecución de proyectos garantizan el cumplimiento de los objetivos. La ISO 9000 brinda un conjunto de conceptos sobre el tema”¹⁰.

“Pero el desarrollo de un sistema de Dirección Integrada de la Calidad debe tener un soporte informático del proyecto que permita desarrollar la DIP. El sistema propuesto es el Project 2002 que permite desarrollar su programación estructurada y establecer los vínculos con la base normalizativa para detallar los criterios de medidas por tareas e hitos”.¹¹

Para lograr la relación entre la calidad y el proyecto es necesario precisar algunos conceptos fundamentales de calidad asociados al proyecto, definir su sistema de dirección asociado a la dirección integrada de proyectos y su programación estructurada, apoyada por un sistema informático profesional que permita, a través de los hitos y los criterios de medida consolidar estos dos conceptos, para ejercer

⁹ RANDOLPH, W. Arnold/ POSNER, Barry Z. *Gerencia de proyectos. Como dirigir exitosamente equipos de trabajo* Mc Graw Hill, Bogotá, 2000.

¹⁰ ISO 9000. *Sistemas de gestión de la calidad. Fundamentos y vocabulario.*

¹¹ DELGADO R.; MONTES M. y TORRES M. La Dirección Integrada de Proyectos apoyada por las bases de datos. Ponencia. Forum de Ciencia y Técnica. Junio/2003

posteriormente un control de la ejecución integrando el costo, el tiempo y la calidad”.¹²

“Así mismo, en la Dirección Integrada de la Calidad el control garantiza el cumplimiento de los objetivos a partir del control cualitativo recogido en las tareas e integrado en los hitos”.¹³

Por otra parte, los sistemas de costo y la logística del proyecto requieren de la integración con la calidad en función de las evaluaciones en el proceso de seguimiento de los proyectos por cortes. Y en la misma dimensión la capacitación continua de los especialistas en función de una estrategia de formación del personal, atendiendo a los objetivos propuestos en el proceso de cambio en la empresa, requiere del uso de las tecnologías informáticas para garantizar su formación discreta, de acuerdo con las oportunidades de tiempo disponible por los especialistas, accediendo a la red y tomando la información necesaria a través de un sistema informático, apoyado por un modelo pedagógico que permita su superación y evaluación del impacto, a través del progreso en los proyectos en los que participa y la evaluación favorable en su desempeño.

1.3.3 El desarrollo de software. Con la globalización, los mercados se pusieron más competitivos y la demanda para los productos y servicios de buena calidad aparecen como el factor decisivo en el momento de escoger un compañero de negocios por los días de hoy. “En el ámbito del desarrollo de Software no es diferente, el cliente, cada vez más exigente, demanda por un sistema dentro del la marca pedida y que atienda sus necesidades, desarrollado en el tiempo más pequeño posible”.¹⁴

A pesar de la creciente participación del software en el mundo actual y de los avances producidos, su proceso aun no es adecuado. El desarrollo de software aun no responde a las exigencias de estos tiempos. Las necesidades y expectativas de los clientes y usuarios no son captadas satisfactoriamente. De ahí que gran cantidad de proyectos de software que no llegan a cumplir sus objetivos, y como consecuencia de esto, los altos por cientos de rechazo entre ellos. Es otra problemática importante la incapacidad de absorber cambios en esos requisitos.

¹²DELGADO R.; VEREZ M.; ROEHRIC K. U. Humboldt. *La Dirección Integrada de Proyectos (Project Management) haciendo uso de las Nuevas Tecnologías de la Informática y las comunicaciones aplicadas al CGLALE*. Kologische Hefte der wirtschafft. Helf 1 b / 2 001. p. p. 99 –108.

¹³ DELGADO, R. VEREZ, M.. *La Dirección Integrada de Proyectos haciendo uso de las tecnologías de la informática y las comunicaciones en el marco del perfeccionamiento empresarial*. Monografías.com Internet. 2003

¹⁴ PRESSMAN, Roger S. *Ingeniería del Software: Un enfoque práctico*. Segunda edición, Editorial McGraw Hill, 1990

Las principales causas de estos problemas son la administración insuficiente de requisitos, comunicación ambigua e imprecisa, inconsistencias no detectadas entre requerimientos, diseño y programación, validaciones tardías de los requisitos, enfrentamiento tardío de riesgos y propagación de cambios sin control. En este sentido, es necesario recordar que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen, se deben a una inadecuada Ingeniería de Requisitos (IR). Actividades propias de esta área, como la especificación de requisitos o la gestión de requisitos del usuario, son algunas de las consideradas más críticas en el desarrollo y la producción del software.¹⁵

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto.

Todos aquellos que trabajamos en informática en algún momento nos hemos preguntado ¿Qué metodología debo usar para el desarrollo de un Software? cuando hemos tenido que desarrollar una aplicación. Y de hecho esta pregunta se torna muy importante, pues como arquitectos de software, debemos tener un plano en que apoyarnos. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están

¹⁵ GARCÍA ÁVILA, Lourdes. Modelo para la evaluación de la calidad del análisis y diseño orientados a objetos de sistemas informáticos (CADOOSI). Tesis de doctorado. Universidad Central de Las Villas, 2000.

revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

En febrero de 2001, tras una reunión celebrada en Utah - EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó *The Agile Alliance*, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos.

El punto de partida es el Manifiesto Ágil, un documento que resume la filosofía “ágil” y que enfatiza el valorar: Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas, desarrollar software que funciona más que conseguir una buena documentación, la colaboración con el cliente más que la negociación de un contrato, y en responder a los cambios más que seguir estrictamente un plan.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo. Los principios son:

- a.** La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- b.** Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- c.** Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- d.** La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- e.** Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.

- f.** El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- g.** El software que funciona es la medida principal de progreso.
- h.** Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- i.** La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- j.** La simplicidad es esencial.
- k.** Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- l.** En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí sí toma sentido el basarnos en una metodología de desarrollo, y empezamos a buscar cual sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no encontramos la más adecuada y terminamos por hacer o diseñar nuestra propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo. Muchas veces realizamos el diseño de nuestro software de manera rígida, con los requerimientos que el cliente nos solicitó, de tal manera que cuando el cliente en la etapa final (etapa de prueba), solicita un cambio se nos hace muy difícil realizarlo, pues si lo hacemos, altera muchas cosas que no habíamos previsto, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido.

Obviamente para evitar estos incidentes debemos haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo. Por experiencia, muchas veces los usuarios finales, se dan cuenta de las cosas que dejaron de mencionar, recién en la etapa final del proyecto, pese a que se les mostró un prototipo del software en la etapa inicial del proyecto. Los proyectos en problemas son los que salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente. Para dar una idea de qué metodología podemos utilizar y cual se adapta más a nuestro medio, se mencionarán tres de ellas que actualmente se consideran las más utilizadas, tal como: RUP, XP y MSF.

RUP (Rational Unified Process) es un proceso de desarrollo propuesto por "Rational Software Corporation", resultado del esfuerzo de las tres últimas

décadas en desarrollo de software y de la experiencia de sus creadores Ivan Jacobson, Grady Booch y James Rumbaugh.

La metodología RUP se divide en 4 fases de desarrollo de software: Inicio, elaboración, construcción y transmisión. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevado entendiendo las necesidades del negocio, trasladando las necesidades y los requerimientos del negocio a un sistema automatizado, realizando un análisis y diseño trasladando así los requerimientos dentro de la arquitectura de software, creando una aplicación que se ajuste a la arquitectura, al negocio y que tenga el comportamiento deseado. Y por último asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

Los elementos del RUP son: Actividades, Trabajadores y Artefactos. Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Extreme Programming o XP es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los roles XP de acuerdo con la propuesta original de Beck son: Programador, Cliente, Tester (encargado de pruebas), Tracker (encargado de seguimiento), Coach (entrenador), Consultor y Big Boss (gestor).

Proceso XP presenta un ciclo de desarrollo que consiste a grandes rasgos en los siguientes pasos:

- a.** El cliente define el valor de negocio a implementar.

- b.** El programador estima el esfuerzo necesario para su implementación.
- c.** El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- d.** El programador construye ese valor de negocio.
- e.** Vuelve al paso a.

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

Microsoft Solution Framework o MSF, es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de gestión, proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Lo que ha hecho de MSF una de las metodologías más utilizadas son sus características, ya que es: adaptable, fácilmente escalable, muy flexible a la hora de requerimientos y clientes, y puede ser usada para desarrollar soluciones basadas sobre cualquier tipo de tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

“En este punto es importante recordar que la relación no lineal entre la ingeniería de requisitos y el resto del ciclo de vida del desarrollo del software ha sido detectada desde antaño y propuestas metodológicas como el Modelo en Espiral¹⁶ y el Proceso Unificado de Racional¹⁷, incorporan estrategias iterativas dentro de sus procesos de desarrollo para facilitar la ejecución de actividades propias de la ingeniería de requisitos, una vez iniciado el resto del proceso de desarrollo, al detectarse en éste la necesidad de renegociar algunos requisitos de difícil implementación o porque aparecen nuevos requisitos durante el proceso de

¹⁶ BOEHM, B. A Spiral Model of Software Development and Enhancement". IEEE Computer. Vol. 21, # 5. 1988.

¹⁷ JACOBSON, I., BOOCH, G., RUMBAUGH, J. *The Unified Software Process*. Addison- Wesley. 1998.

desarrollo, entre otros. Debe tenerse en cuenta que la IR continúa durante todo el proceso de desarrollo”.¹⁸

Pero sobre todo esto hay que reconocer claramente que en el mundo altamente tecnológico como es en el que actualmente vivimos, debemos ser productivos. En tal medida se debe admitir que incluso el mejor modelo de desarrollo e investigación en ingeniería puede fracasar fácilmente, al no tenerse en cuenta que ahora todo es un negocio y todo proyecto debe generar una ganancia económica. En otras palabras hasta la más básica investigación científica debe tener una manera de ser administrada o gestionada. Por eso los desarrolladores de software y soluciones informáticas deben conjugar dos visiones para realizar su trabajo: La visión de la investigación y desarrollo tecnológico que, si se escoge correctamente la metodología, puede brindar muy buenos resultados en lo que al producto final se refiere; y la visión empresarial con la cual se busca que el proyecto de desarrollo de software, así como con cualquier otro, sea gerenciado para ser productivo, sea de calidad, altamente funcional y siempre pensando en la satisfacción del cliente. Por eso se hace necesario no solo contar con una metodología adecuada para el desarrollo tecnológico sino también con un sistema administración para el proceso de gestión de creación de un sistema o software, el cual encierra un conjunto de actividades.

1.3.4 El software educativo. El manejo y rápido tratamiento de la información se plantea como el nuevo paradigma de las sociedades modernas; independientemente del enfoque ideológico-político que el término "globalización" implique. Estos avances han generado que los sistemas educativos se revisen y actualicen para responder a las exigencias que se avecinan, de acuerdo con las obvias tendencias. Bajo este panorama, se requiere mejorar el rendimiento académico, mediante la ayuda permanente de un proceso instruccional interactivo con variedad de medios y técnicas adecuadas que permitan proveer soluciones de aprendizaje integral. Se necesita así una metodología para el desarrollo de software educativo que permita optimizar su producción, y determinar el lenguaje de autor más apropiado y los requerimientos correspondientes.

El desarrollo de un software educativo permite generar nuevas estrategias de aprendizaje, donde los profesores encuentran un aliado en la prosecución de actividades que permitan a un grupo de alumnos mejorar el rendimiento académico, de tal manera que logre ser un ciudadano próspero y valioso para el país.

Actualmente son pocos los profesores que pueden tener un software educativo como herramienta de apoyo a su quehacer diario y muchos menos los que puedan desarrollarlos. Esto, aunado a la poca receptividad que el uso del computador ha

¹⁸ SAWYER, P. Y KONTOYA, G. *Software Requirements Engineering Knowledge Area Description*. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.

tenido en las escuelas, impide el desarrollo y la aplicación de la nueva tecnología en el proceso educativo, aunque hay algunas que han hecho esfuerzos por lograr que sea una actividad común dentro de las mismas. La sola dotación de laboratorios de computación equipados no es suficiente, ya que los docentes necesitan tener preparación para la nueva tecnología dentro del aula.

La informática educativa como la ciencia que estudia las formas y maneras de utilizar el computador y las nuevas tecnologías de la comunicación y la información en educación, da al docente herramientas imprescindibles que le permiten aprovechar estos avances tecnológicos. Esta función relevante de la informática educativa ha llevado a que algunas universidades del país desarrollen cursos de capacitación tanto a docentes en servicio como a futuros docentes, intentando llenar la necesidad de personal vinculado a la educación con conocimientos en esta área.

Ahora bien, la diversidad tecnológica actual, ha hecho surgir formas de integrar distintas tecnologías tales como la informática y las telecomunicaciones, dando origen a la mayor revolución tecnológica de los últimos tiempos, la Internet, en especial la World Wide Web (WWW).

Esto ha permitido la existencia de software educativos de tecnología multimedia y ambientes virtuales, pero sobre todo la existencia de metodologías para determinar la forma cómo se tiene acceso a ellos, cómo interactúan y evolucionan desde contextos locales a contextos totalmente distribuidos, accediéndose a la premisa de: "lo que quiera, dónde quiera y cuándo quiera".

Teniendo estas tecnologías como contexto, las bondades de un software educativo unido con las facilidades de masificación de la WWW permiten que personas preocupadas por mejorar la calidad de la educación puedan desde su hogar, trabajo o lugares apropiados, capacitarse y/o especializarse en el área en que se desempeña, y que de otra manera estarían imposibilitados a realizar.

De tal manera que el desarrollo de software educativo bajo plataforma web, como apoyo didáctico a sistemas tanto presenciales como a distancia, se hace necesario para lograr el mejor uso posible de las nuevas tecnologías de la comunicación y la información.

Hoy día, la humanidad como consecuencia de la dinámica de cambios acelerados, se ha colocado en una posición vanguardista creativa, con iniciativa para innovar, convirtiéndose esto en retos constantes que se deben enfrentar con perseverancia y esfuerzo.

De lo que se trata es de modernizar, crecer y desarrollar un país sobre la base de un ciudadano competitivo con capacidad adecuada para reducir costos operativos

y disminuir el tiempo de respuesta en la ejecución de cualquier actividad a desarrollar.

Sobre este aspecto, opinan especialistas de reconocida profesionalidad en el área económica, que la creciente concepción y aceptación de la llamada globalización, impone al hombre nuevos aprendizajes, nuevas tecnologías y nuevos conocimientos. De allí la necesidad de una posición educativa que busque sustituir sus paradigmas actuales; convirtiéndose esta búsqueda en retos constantes.

Tratando de vincular esta corriente de opinión, al respecto señalan Mardones y Ruiz¹⁹ que, “la educación y la producción del conocimiento constituyen en la actualidad la base del proceso de desarrollo nacional”.

Para afianzar su afirmación agregan los citados autores que, la globalización de la Economía, Industria y creciente integración que hoy se da entre países ha sido posible, en buena medida, por el acelerado desarrollo tecnológico donde se han apoyado, y entre los cuales destacan los sistemas de Telemática e Informática. Es decir, es la revolución de la comunicación tecnológica.

Planteado así como está, se puede inferir que este tipo de desarrollo tecnológico ha revolucionado el mundo de tal forma que el uso de las computadoras se ha hecho casi imprescindible en la mayoría de las actividades que se desarrollan en la sociedad actual. Un ejemplo de ello es la manera cómo el manejo y rápido tratamiento de la información se plantea como el nuevo paradigma de las sociedades modernas y hasta se habla de la sociedad globalizada relacionada por la telemática; independientemente del enfoque ideológico-político que el término globalización implique.

Al respecto, Sequera destaca que las nuevas tecnologías juegan un papel significativo en los planteamientos educativos, alterando algunos parámetros clásicos de la enseñanza presencial, y permiten a la educación a distancia alcanzar un alto valor dentro del sistema educativo, incorporando en su modelo los recursos que ofrecen ellas”.²⁰

Todo esto propone retos a los métodos de enseñanza actuales, donde la experiencia de cursos virtuales motiva la reflexión en torno a la Universidad que se desea tener en el milenio que estamos viviendo.

¹⁹ MARDONES E. y RUIZ M. *La Informática Educativa, un Factor de Cambio para la Educación Venezolana del Siglo XXI*. Revista Encuentro Educativo. Vol. 2 No. 2. Universidad del Zulia. Facultad de Humanidades y Educación. Centro de Documentación e Investigación Pedagógica. 1996.

²⁰ SEQUERA, M. *Diseño de una Propuesta Curricular para una Maestría en Informática Educativa*. Trabajo Especial de Grado para optar al título de Magíster en Educación Área de Planificación y Administración Educativa. Maracaibo. Universidad del Zulia, 1999.

Son múltiples las propuestas que se encuentran para la producción de software educativo, la mayoría de ellas dirigidas al desarrollo de software prototípico, elaborado por grupos especializados de profesionales.

Entre estas propuestas se encuentran aquellas de tipo estructurado, como por ejemplo las de modelo sistemático propuesto por Dick, Carey y Gagné²¹, constituido por cinco fases lineales como el análisis, diseño, desarrollo, evaluación e implementación de software y la de tipo no lineal, propuesto por Tripp y Bichelmayer, el cual consiste en formular los objetivos del software, buscar un diseño que consiga los objetivos propuestos, formular soluciones desarrolladas en un prototipo, a partir del cual se revisan los objetivos y las soluciones propuestas. En este tipo de propuestas, se encuentran un tanto alejadas del contexto pedagógico en el cual se van a aplicar y parten en general, de un análisis o planteamiento de objetivos de carácter técnico.

Posteriormente, se encuentran modelos encaminados al desarrollo de software multimedial, como por ejemplo el modelo basado en la ingeniería de software desarrollada por Alvaro Gálvis²² (1992), la que comienza el análisis de las situaciones, seguida de una etapa de diseño educativo y comunicacional, el desarrollo del software educativo y posteriormente la prueba piloto y de campo, de donde puede volver al momento de análisis para mejora y retroalimentación. Otros autores describieron un modelo basado en procesos técnicos de cinco fases: concepción del software educativo, conceptualización, pre producción o desarrollo del guión, producción y post producción, o entrega. Por su parte, Zambrano²³ (1995) inicia con el momento pedagógico para definir objetivos y contenidos, estudio de probabilidad de realización, la etapa de desarrollo gráfico y programación y el estudio del producto, o evaluación de calidad.

Se menciona también el modelo de Liu, orientado hacia el desarrollo de negocios, que propone cinco etapas: consolidación o de obtención del contrato, planificación de presupuestos y el alcance del proyecto, diseño, producción, prueba del software multimedia y finalmente, el mercadeo, la distribución y el apoyo del producto.

La metodología propuesta para el desarrollo de un Software Educativo Hipermédial realizada por Ovalle y Padilla, presenta cinco etapas: análisis y estudio de factibilidad, el diseño y esquematización pedagógica de la aplicación, el

²¹ GAGNE, R. *Instructional technology*. En Fundatios. Hillsdale, Lawrence Erlbaum Associates, 1987.

²² GALVIS, Alvaro. *Ingeniería de software educativo*. Ediciones Uniandes. Bogotá, 1992.

²³ ZAMBRANO, J. *Enseñanza asistida por computador y producción de software educativo (PROSDOS)*. Caracas, Venezuela. Imprenta Universitaria Universidad Central de Venezuela, 1995.

desarrollo y programación y la distribución para su aplicación, mientras que Chacón, planteó la necesidad de cuatro etapas: diseño, producción, evaluación y entrega, Ruffini, propone siete etapas: el análisis de la audiencia, selección del tópico a tratar; elaboración de objetivos; definición del tipo de proyecto, por el autor, el hiperlibro, la investigación, la presentación y el viaje interactivo; diseño de los contenidos; diseño de los hipervínculos de navegación, y la evaluación del proyecto.

Para el planteamiento del modelo del proceso de desarrollo de software educativo, es necesario tener en consideración los siguientes principios fundamentales:

Un modelo es necesariamente la simplificación de la realidad, si no sería la realidad misma y no contribuiría en nada a la construcción del conocimiento sobre el proceso de desarrollo de software educativo. Así, es una representación abstracta, lo cual significa que la aproximación -e incluso el error- es inherente al modelo.

El de desarrollo de software educativo es un proceso complejo, en el que entran en juego múltiples factores de carácter cultural, social, educativo, técnico y comunicativo, que lo convierten en un proceso dinámico, es decir, en construcción y evolución permanente.

No se puede afirmar que exista un solo tipo de software educativo, sino que este ha venido evolucionando de acuerdo con las innovaciones tecnológicas, que han redundado en la existencia de una variada gama de software educativo, que van desde los software de tipo transmisionista, pasando por el software activo, hasta llegar al interactivo.

En el proceso de desarrollo de software educativo es necesario tener en cuenta los procesos educativos que apoyan y que se fundamentan en enfoques pedagógicos variados, de acuerdo con las diferentes tendencias utilizadas, las cuales van desde enfoques instruccionalistas, tradicionales, pasando por enfoques activos, críticos, constructivistas e interaccionistas.

El desarrollo de software educativo se apoya en procesos técnicos para su producción, los cuales, al igual que los enfoques pedagógicos, son variados y dependen de las plataformas utilizadas, el tipo de tecnología disponible y aquella que se pueda utilizar en el sector educativo.

1.3.5 La gerencia en los dos contextos. De acuerdo a lo expresado todo debe comenzar con una planificación del proyecto de Software pensando no solo en tecnología sino también en negocio. Su objetivo es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo

limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

En cuanto a la determinación de recursos, se debe considerar que uno de los resultados más valiosos del planteamiento es la integración de los recursos y actos de la institución en torno a metas generales. En este orden de ideas, como explica Marshal Dimock, "Planear es poner claros los objetivos propios y luego determinar qué acciones habrían de tomarse, por quién, cuándo, mediante qué método y a qué costo, al fin de alcanzar las metas deseadas".²⁴

Al estimar tomamos en cuenta no solo el procedimiento técnico a utilizar en el proyecto, sino que se toma en cuenta los recursos, costos y planificación. El Tamaño del proyecto es otro factor importante que puede afectar la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del Software. En el principio el costo del Software constituía un pequeño porcentaje del costo total de los sistemas basados en Computadoras. Hoy en día el Software es el elemento más caro de la mayoría de los sistemas informáticos. Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo. La estimación basada en procesos es la técnica más común para estimar un proyecto tecnológico. Esta basa la estimación en el proceso que se va a utilizar, es decir, el proceso se descompone en un conjunto relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para llevar a cabo la estimación de cada tarea. Al igual que las técnicas basadas en problemas, la estimación basada en el proceso comienza en una delineación de las funciones del software obtenidas a partir del ámbito del proyecto. Se mezclan las funciones del problema y las actividades del proceso. Como ultimo paso se calculan los costos y el esfuerzo de cada función y la actividad del proceso de software.

Por supuesto una parte muy importante de la planificación del desarrollo de Software es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de Software, esto simula a una pirámide donde las Herramientas (hardware y Software), son la base proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los Componentes reutilizables. Y en la parte mas alta de la pirámide se encuentra el recurso primario, las personas (el recurso humano). La cantidad de personas requeridas para el desarrollo de un proyecto de software solo puede ser

²⁴ DIMOCK, Marshal. *Principios y Normas de Administración*. Ed. Libreros Mejicanos Unidos, México, 1965. p. 137

determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años), y seleccionar la posición dentro de la organización y la especialidad que desempeñara cada profesional. Y ahora que se debe pensar empresarialmente aun en este tipo de proyectos, la importancia que se le de al manejo de las personas que hacen parte del mismo puede ser la diferencia entre simplemente un buen software y un buen producto comercial que ofrece soluciones informáticas y ganancias para sus desarrolladores.

Cualquier estudio sobre recursos de software en una empresa desarrolladora y que piensa con una visión de negocio estaría incompleto sin estudiar la reutilización, esto es la creación y la reutilización de bloques de construcción de Software. Tales bloques se deben establecer en catálogos para una consulta más fácil, estandarizarse para una fácil aplicación y validarse para la también fácil integración.

Por otro lado, muchas veces cuando se emprende el desarrollo de un proyecto de Sistemas los recursos y el tiempo no son realistas para su materialización sin tener pérdidas económicas y frustración profesional. La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta tres áreas principales de interés: La viabilidad económica, la viabilidad técnica, viabilidad Legal.

El análisis económico incluye lo que llamamos, el análisis de costos – beneficios, significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad del producto o sistema. Muchas veces en el desarrollo de Sistemas de Computación estos son intangibles y resulta un poco dificultoso evaluarlo, esto varía de acuerdo a las características del Sistema. El análisis de costos – beneficios es una fase muy importante de ella depende la posibilidad de desarrollo del Proyecto.

En el Análisis Técnico, el Analista evalúa los principios técnicos del Sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad. Los resultados obtenidos del análisis técnico son la base para determinar sobre si continuar o abandonar el proyecto, si hay riesgos de que no funcione, no tenga el rendimiento deseado, o si las piezas no encajan perfectamente unas con otras.

El estudio de la viabilidad puede documentarse como un informe aparte para la alta gerencia.

Por ultimo en la planeación de un proyecto de este tipo se debe tener en cuenta el entorno donde se apoya el proyecto de Software, llamado a menudo entorno de Ingeniería de Software, constituido por Hardware y Software. El primero proporciona una plataforma con las herramientas (Software) requeridas para

producir los productos que son el resultado de la buena práctica de la Ingeniería del Software, un planificador de proyectos debe determinar la ventana temporal requerida para el Hardware y el Software, y verificar que estos recursos estén disponibles. Muchas veces el desarrollo de las pruebas de validación de un proyecto de software para la composición automatizada puede necesitar un compositor de fotografías en algún punto durante el desarrollo. Cada elemento de hardware debe ser especificado por el planificador del Proyecto de Software.

Todo lo planteado anteriormente permite observar muy claramente que las metodologías de desarrollo tecnológico y los modelos de gestión de proyectos no son excluyentes, por el contrario en los países más desarrollados y algunos otros que no lo son tanto como La India, Uruguay e Israel las empresas desarrolladoras de software han entendido que al manejar el desarrollo de software, no solo como un logro intelectual creado de una manera casi artesanal, sino como un negocio que incluye productos que prestan servicios y que con una buena administración al desarrollar los proyectos se puede obtener una empresa rentable y con un amplio espectro de posibilidades. El presente trabajo busca poder crear un modelo útil y viable para administrar los diversos proyectos de desarrollo de software en nuestro entorno, teniendo en cuenta las metodologías más atractivas y óptimas en cuanto al desarrollo tecnológico se refiere, varias de las cuales ya han sido expuestas de manera resumida en este marco teórico conceptual, así como los principios y modelos gerenciales que pueden transformar a una organización desarrolladora de software en un verdadero negocio.

1.4 MARCO CONCEPTUAL

En el trabajo de grado se utilizan conceptos técnicos diversos. Los fundamentales se especifican a continuación.

Actividad. Un elemento de trabajo desarrollado durante el curso de un proyecto. Una actividad normalmente tiene una duración esperada, un costo esperado, y unos requerimientos esperados de recursos. Las actividades generalmente se subdividen en tareas.

Administración de calidad del proyecto. Es un subconjunto de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto va a satisfacer las necesidades para las cuales fue encomendado. Y consiste de planeación de la calidad, aseguranza de la calidad, y control de calidad.

Administración de proyectos. Es la aplicación de conocimientos, habilidades, herramientas, y técnicas a las actividades del proyecto de manera que se cumplan o excedan las necesidades y expectativas que los partidos interesados tengan en el proyecto.

Charter del proyecto. Es un documento emitido por la alta administración que provee al administrador del proyecto con la autoridad de aplicar recursos de la organización a las actividades del proyecto.

Ciclo de vida del proyecto. Es una colección de fases de proyecto generalmente secuenciales cuyos nombres y números están determinadas por las necesidades de control de organización u organizaciones involucradas en el proyecto.

Control de calidad (QC). Es el proceso de monitorear resultados específicos del proyecto para determinar si estos cumplen los estándares relevantes de calidad e identificar maneras de eliminar causantes de desempeño no satisfactorios. (2) Es la unidad organizacional a la que se le asigna la responsabilidad por el control de la calidad.

Control. Es el proceso de comparar el rendimiento real con el planeado, analizar varianzas, evaluar posibles alternativas, y tomar la acción correctiva apropiada en la medida que se necesite.

Convergencia de Rutas. En el análisis matemático, es la tendencia de caminos paralelos de aproximadamente igual duración a retrasar la terminación de los hitos donde convergen.

Cuerpo de Conocimientos de la Administración de Proyectos (PMBOK). Es un término inclusivo que describe la suma de conocimientos dentro de la profesión de la administración de proyectos. Como en otras profesiones tales como abogacía, medicina, y contabilidad, el cuerpo de conocimiento descansa en los practicantes y académicos que la aplican y avanzan. El PMBOK incluye prácticas tradicionales probadas que son de uso generalizado, así como prácticas innovadoras y avanzadas que han visto un uso más limitado.

Desarrollo de la Programación. Es analizar la secuencia de actividades, duración de actividades, y los requerimientos de recursos para crear la programación del proyecto.

Desarrollo del Plan de Proyecto. Es tomar los resultados de los otros procesos de planeación y colocarlos en un solo documento consistente y coherente.

Equipo Administrativo de Proyectos. Son los miembros del equipo de proyecto que están directamente involucrados en las actividades de la administración de proyectos. En proyectos más pequeños, el equipo administrativo de proyectos puede virtualmente incluir a todos los miembros del equipo de proyecto.

Fast Tracking. Es comprimir la programación de proyecto al traslapar actividades que normalmente se harían en secuencia, tales como diseño y construcción. Algunas veces se confunde con ingeniería concurrente.

Flotación. Es la cantidad de tiempo que una actividad se puede retrasar desde su comienzo temprano sin atrasar la fecha de terminación del proyecto. La flotación es un calculo matemático y puede cambiar a medida que el proyecto progresa y se efectúan cambios al plan del proyecto. También se le conoce

Hanger. Es una discontinuidad o quiebre no intencionado en la ruta de la red .Los hangers generalmente son causados por actividades faltantes o por relaciones lógicas faltantes.

Matriz de Asignación de Responsabilidades (RAM). Es una estructura que relaciona la estructura organizativa a la estructura de desglose de trabajo para ayudar a asegurar que cada elemento de trabajo del alcance del proyecto se a asignado a un individuo responsable.

Método de Diagramación de Precedencias (PDM). Es una técnica de diagramación de redes en la que las actividades se representan con cajas (o nodos). Las actividades están ligadas por medio de relaciones de precedencia para mostrar la secuencia en las que las actividades deberán ser ejecutadas.

Método de la Ruta Crítica (CPM). Es una técnica de análisis de red usada para predecir la duración del proyecto al analizar que secuencia de actividades (que ruta) tiene la menor cantidad de flexibilidad de programación (la menor cantidad de flotación) Las fechas tempranas se calculan por medio de un pase hacia delante usando una fecha especificada de comienzo. Las fechas tardías se calculan por medio de un pase hacia atrás comenzando desde una fecha especificada de terminación (usualmente la fecha temprana de terminación del proyecto calculado por el pase hacia adelante).

Organización Ejecutora. Es la empresa cuyos empleados están más directamente involucrados en realizar el trabajo de proyecto.

Organización Funcional. Es una estructura organizacional en la cual el staff esta agrupado de manera jerárquica por especialidad (e.g., producción, mercadeo, ingeniería, y contabilidad en el nivel superior; con la ingeniería, subdividida en mecánica, eléctrica, y otras).

Organización Matricial. Es cualquier estructura organizacional en la que el administrador de proyectos comparte responsabilidad con los administradores funcionales para la asignación de prioridades y por la dirección del trabajo de individuos asignados al proyecto.

Organización Projectizada. Es cualquier estructura organizacional en la que el administrador tiene total autoridad para asignar prioridades, y de dirigir el trabajo de individuos asignados al proyecto.

Paquete de Trabajo. Es una entrega al nivel más bajo de la estructura de desglose de trabajo. Un paquete de trabajo se puede dividir en actividades.

Patrones para el desglose del trabajo (WBS). Una estructura de desglose de trabajo de un proyecto previo puede ser usado como un patrón para un nuevo proyecto. Aunque cada proyecto es único, un WBS puede ser muchas veces “reutilizado” ya que muchos proyectos se parecen a otro proyecto en algún grado.

Plan del Proyecto. Es un documento formal, aprobado usado para guiar tanto la ejecución como el control del proyecto. Los usos primarios del plan de proyecto son documentar las suposiciones de la planeación y toma de decisiones, de facilitar la comunicación entre los partidos interesados del proyecto, y de documentar los cambios aprobados a la línea de base del alcance, costos, y programación. Un plan de proyecto puede ser detallado o concatenado.

Reserva Administrativa. Es una cantidad planeada por separado que se usa para situaciones futuras que son imposibles de predecir (a veces llamadas “desconocido conocido”). Las reservas administrativas pueden incluir los costos o la programación.

Sistemas de información de administración de proyectos (PMIS). Un sistema de información para administración de proyectos consiste de las herramientas y técnicas usadas para recoger, integrar, y diseminar las salidas de los otros procesos de administración de proyectos. Se usa para darle soporte a todos los aspectos del proyecto desde su iniciación hasta su finalización y generalmente incluye tanto sistemas automáticos como manuales.

Software de Administración de Proyectos. Es una categoría de aplicaciones para computadoras diseñadas especialmente para asistir con la planeación y control de la programación y costos de los proyectos.

Técnica de Revisión y Evaluación de Programas (PERT). Es una técnica de análisis de red orientada hacia eventos usada para estimar la duración del proyecto cuando existe un alto grado de incertidumbre dentro de los estimados individuales de las duraciones de las actividades. PERT aplica el método de la ruta crítica a un estimado de duración ponderado promedio.

Técnica de Revisión y Evaluación Gráfica (GERT). Es una técnica de análisis de red que permite el tratamiento condicional y probabilístico de las relaciones lógicas (i.e., algunas actividades pueden no ejecutarse).

Workaround. Es una respuesta a un evento negativo de riesgo. Se debe distinguir de plan de contingencia en que un workaround no es planeado en anticipación de la ocurrencia del evento de riesgo.

1.5 MARCO CONTEXTUAL

El trabajo de grado realizado tuvo dos frentes. Por una parte se refiere al ámbito gerencial de los proyectos, en las Instituciones de Educación Superior, consideradas como empresas. Por otra toma el campo del desarrollo de proyectos de Software, que requiere la participación administrativa desde el contenido gerencial.

Específicamente, el trabajo de grado elabora una propuesta gerencial para la administración de proyectos de desarrollo de software en instituciones de educación superior. Al respecto se aclara que en términos concretos se dirige a los proyectos de desarrollo de software que actualmente están en curso en La Universidad Mariana en la ciudad de Pasto, en la cual labora, en La Universidad virtual, el autor del trabajo. Por tales razones, los elementos generales de administración de proyectos sobre la base de lo aprendido en la especialización, ya se han puesto en práctica.

1.6 ASPECTOS METODOLÓGICOS

El trabajo de grado realizado, como tipo de estudio, es primordialmente *descriptivo* porque se ocupa de la descripción de las características que identifican los diferentes elementos y componentes de un modelo gerencial, dentro del ámbito de las ciencias administrativas, y su correspondiente interrelación. Es en ese sentido es a la vez un punto de partida para investigaciones posteriores.

Por otra parte, en términos del paradigma investigativo, el trabajo se inscribe en el ámbito del *explicativo*. Según este paradigma, en primer lugar, se plantea la posibilidad de reducir la subjetividad individual, reconociendo una objetividad científica basada en la intersubjetividad de los diversos miembros de la comunidad científica que evalúan la validez de los resultados obtenidos por el análisis realizado por un estudio como el que se pretende llevar a cabo. En este sentido la propuesta pretende generalizar sus elementos a las Instituciones de Educación superior que trabajen proyectos de software. En segundo lugar, dentro de la concepción del paradigma explicativo, el trabajo de grado se define en el ámbito de las investigaciones cualitativas explicativas, pero que utiliza elementos de análisis cuantitativo.

En lo que respecta al método el trabajo se inscribe en el campo *inductivo*. Por esto estudia fenómenos particulares, como son los proyectos de desarrollo de software, con el propósito de llegar a conclusiones y premisas generales que pueden ser aplicadas en situaciones similares a la estudiada. Teniendo en cuenta esta posición, se puede plantear que a partir de este conocimiento inductivo se puede llegar a establecer leyes generales de comportamiento de los fenómenos, en este caso los proyectos de software, a partir del análisis de los hechos empíricos.

2. COMPONENTES FUNDAMENTALES DE LA PROPUESTA GERENCIAL

2.1 EL MARCO DE LA GERENCIA DE PROYECTOS

2.1.1 Los elementos básicos. El Marco de la Gerencia de Proyectos, provee la estructura básica para entender la administración de proyectos. En términos generales se deben contemplar 12 fases para el documento respectivo, que se especifican a continuación.

A. Introducción, que define los elementos claves y provee una vista del resto del documento.

B. El Contexto de la administración de proyectos. Que describe el ambiente en el cual los proyectos operan. El equipo de administración de proyectos debe comprender este contexto más amplio — administrar las actividades día a día de un proyecto es necesario para su éxito pero no es suficiente.

C. Los Procesos de administración de proyectos. Fase en la cual se da una vista generalizada de como los procesos varios de la administración de proyectos interactúan comúnmente.

D. Administración de la integración de proyectos. En la cual se plantean los procesos requeridos para asegurar que los elementos varios de un proyecto están coordinados apropiadamente. Consiste del desarrollo de un plan de proyecto, ejecución del plan de proyecto, y el control de cambios en general.

E. Administración del alcance del proyecto. Que describe el proceso requerido para asegurar que el proyecto incluye todo trabajo requerido, y sólo el trabajo requerido, para completar el proyecto de manera exitosa. Consiste de la iniciación, planeación del alcance, definición del alcance, verificación del alcance, y control de cambio al alcance.

F. Administración del tiempo del proyecto. Fase que sirve para presentar los procesos requeridos para asegurar la terminación a tiempo del proyecto. Consiste en la definición de las actividades, secuencia de las actividades, estimación de duración de las actividades, desarrollo del cronograma y control de la programación.

G. Administración de los costos del proyecto. Parte en la cual se presentan los procesos requeridos para asegurar que el proyecto es completado dentro del presupuesto aprobado. Consiste en la planificación de recursos, estimación de costos, presupuestación de costos, y control de costos.

H. Administración de la calidad del proyecto. Donde se plantean los procesos requeridos para asegurar que el proyecto satisfará las necesidades para lo cual fue desarrollado.

I. Administración de los recursos humanos del proyecto. Que describe los procesos requeridos para hacer el uso más eficiente de las personas involucradas en el proyecto. Consiste en la planeación organizacional, adquisición de staff, y desarrollo del equipo.

J. Administración de las comunicaciones del proyecto. Describe los procesos requeridos para asegurar la generación apropiada y a tiempo, colección, disseminación, almacenamiento, y la disposición final de la de la información del proyecto. Consiste en la planeación de la comunicación, distribución de la información, reportes de desempeño, y el cierre administrativo.

K. Administración de riesgo del proyecto. Que describe los procesos concernientes con la identificación, análisis, y respuesta al riesgo del proyecto. Consiste en la identificación del riesgo, cuantificación del riesgo, desarrollo de la respuesta al riesgo, y en el control de la respuesta al riesgo.

L. Administración de la procuración del proyecto. Presenta los procesos requeridos para adquirir bienes y servicios de fuera de la organización ejecutora. Consiste en la planeación de la gestión de la procuración, planear la solicitud, la solicitud, selección de proveedores, administración de contratos, y cierre de contratos.

2.1.2 Sumario de las áreas básicas de la administración de proyectos. Se resaltan nueve componentes, a saber:

A. Administración de la integración del proyecto. Consta de las siguientes partes:

Desarrollo del Plan del Proyecto. Es tomar los resultados de otros procesos de planificación y colocarlos en un solo documento coherente, y lógico.

Ejecución del Plan del Proyecto. Es desarrollar el plan del proyecto, al ejecutar las actividades incluidas en el plan.

Control general de cambios. Consiste en coordinar los cambios a través de todo el proyecto.

B. Administración del alcance del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto incluye todo el trabajo requerido, y solo el trabajo requerido, para completar el proyecto de manera exitosa. Consiste en:

Iniciación, o sea comprometer a la organización a comenzar la siguiente fase del proyecto.

Planeación del alcance: desarrollar un documento del alcance escrito, que sirva como la base para la toma de decisiones futuras del proyecto.

Definición del alcance: subdividir las principales entregas del proyecto, en componentes más pequeñas y manejables.

Verificación del alcance: formalizar la aceptación del alcance del proyecto.

Control de cambios del alcance: controlar los cambios al alcance del proyecto.

C. Administración del tiempo del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto terminara a tiempo. En ella se tiene en cuenta:

La Definición de actividades: identificar las actividades específicas que tienen que ser desarrolladas para poder producir las entregas varias del proyecto.

Secuencia de las actividades: identificar y documentar las interdependencias de las mismas.

Estimación de la duración de las actividades. Consiste en estimar el número de periodos de trabajo que se necesitaran para completar las actividades individuales.

Desarrollo de la programación. Es analizar las secuencias de las actividades, las duraciones de las actividades, y los requerimientos de recursos para poder crear la programación del proyecto.

Control de la programación. Consiste en controlar los cambios a la programación de proyecto.

D. Administración de los costos del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto se competa dentro del presupuesto aprobado. Consiste en lo siguiente:

Planeación de recursos, o sea determinar que recursos (personas, equipo, materiales) y que cantidad de cada uno se debe utilizar para ejecutar las actividades del proyecto.

Estimación de costos. Consiste en desarrollar una aproximación (estimado) de los costos de los recursos necesarios para completar las actividades del proyecto.

Presupuestación de los costos. Es asignar el estimativo general de costos a los ítems individuales de trabajo.

Control de costos, o sea controlar los cambios al presupuesto del proyecto.

E. Administración de la calidad del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para asegurar que el proyecto va a satisfacer las necesidades para las cuales fue acometido. Consta de:

Planeación de la calidad: identificar que estándares de calidad son relevantes al proyecto y determinar como satisfacerlos.

Aseguramiento de la calidad: evaluar el desempeño general del proyecto de manera regular para proveer confianza de que el proyecto va a satisfacer los estándares relevantes de calidad.

Control de calidad: es monitorear resultados específicos del proyecto para determinar si cumplen con los estándares relevantes de calidad e identificar maneras de eliminar las causas de desempeño no satisfactorio.

F. Administración de los recursos humanos del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para hacer el uso más efectivo de las personas involucradas con el proyecto. Consiste en:

La Planeación organizacional, o sea identificar, documentar, y asignar roles de proyecto, responsabilidades, y relaciones de reporte.

Adquisición de staff, que es conseguir los recursos humanos necesarios, asignados para trabajar en el proyecto.

Desarrollo del equipo, o sea desarrollar las habilidades individuales y de grupo para el mejoramiento del desempeño del proyecto.

G. Administración de las comunicaciones del proyecto. Es la parte de la administración de proyectos que incluye los procesos requeridos para asegurar una generación, colección, diseminación, almacenaje, y disposición de la información del proyecto de forma apropiada y a tiempo. Consta de lo siguiente:

Planeación de la comunicación, entendida como el determinar las necesidades de información y comunicación de los partidos interesados del proyecto: quien necesita que información, cuando la necesitaran, y como se les será entregada.

Distribución de la información: es hacer que la información necesaria este disponible para los partidos interesados del proyecto de manera oportuna.

Reporte de desempeño: coleccionar y diseminar la información de desempeño. Esto incluye reportes de status, medición de avance, y pronósticos.

Cierre administrativo: generar, recoger, y diseminar información para formalizar la terminación de una fase o del proyecto.

H. Administración del riesgo del proyecto. Es una parte de la administración de proyectos que incluye los procesos que se ocupan de identificar, analizar, y responder al riesgo del proyecto. Consiste en:

Identificación del riesgo, entendida como el determinar que riesgos posiblemente afecten al proyecto y documentar las características de cada uno.

Cuantificación del riesgo. Es evaluar los riesgos y las interacciones del riesgo para evaluar el rango de posibles resultados del proyecto.

Desarrollo de la respuesta al riesgo. Es definir pasos de mejoramiento para el aprovechamiento de oportunidades, o de respuesta a amenazas.

Control de respuesta al riesgo. Responder a cambios en el riesgo sobre la ejecución de proyecto.

I. Administración de la procuración del proyecto. Es una parte de la administración de proyectos que incluye los procesos requeridos para adquirir bienes y servicios de afuera de la organización ejecutora. Consiste en:

Planeación de adquisiciones: determinar que adquirir y cuando.

Planeación de sollicitación: documentar los requerimientos de producto e identificar las fuentes potenciales.

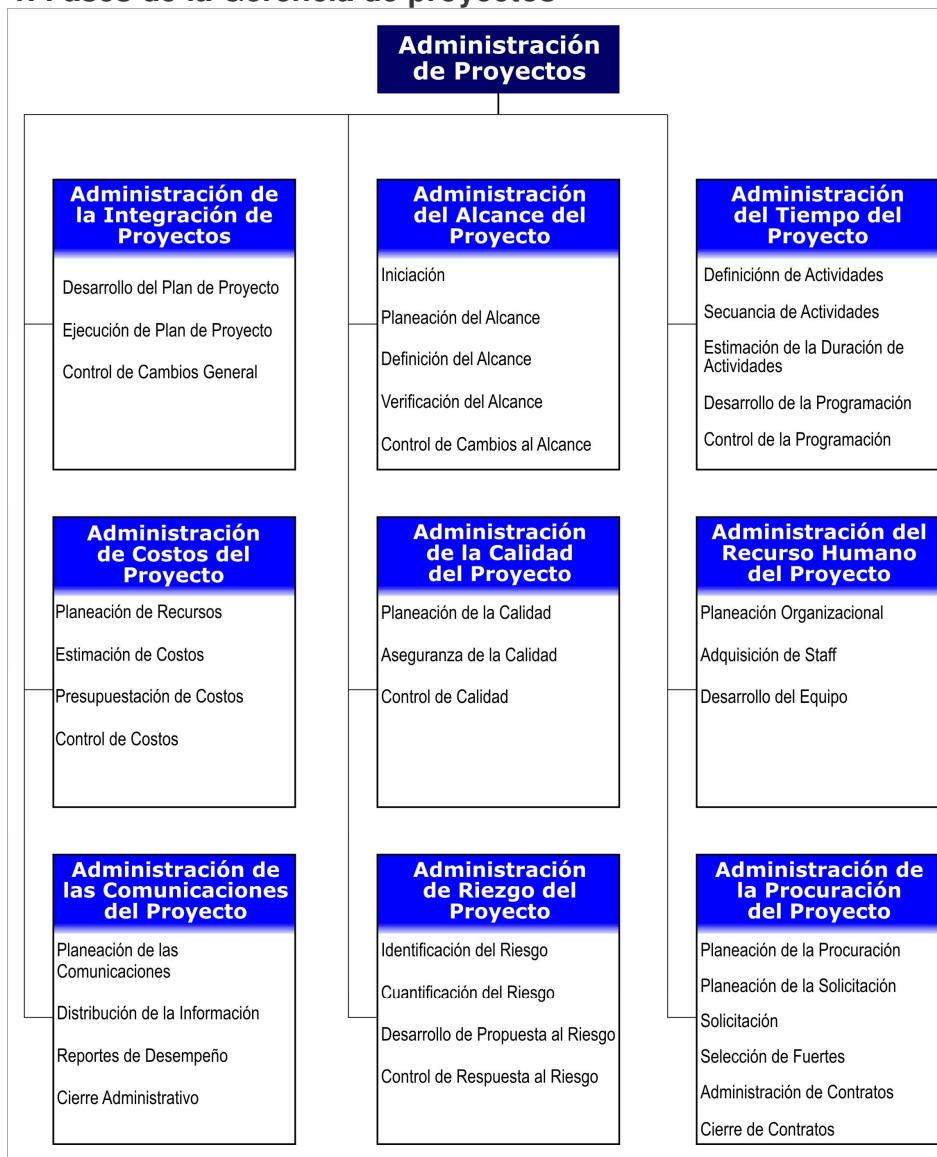
Selección de fuentes: escoger de entre proveedores potenciales.

Administración del contrato: administrar las relaciones con el proveedor.

Cierre del contrato. Es el cierre y negociación del contrato, incluye la resolución de cualquier ítem abierto.

En la figura 1 se esquematizan en resumen las áreas básicas descritas.

Figura 1. Fases de la Gerencia de proyectos



Fuente. Esta investigación

2.2 ASPECTOS PRIMORDIALES

2.2.1 Fases y ciclo de vida del proyecto. Porque los proyectos son tareas únicas, involucran cierto nivel de incertidumbre. Las organizaciones ejecutoras de proyectos generalmente dividen cada proyecto en fases del proyecto para poder administrar mejor los enlaces apropiados con las operaciones de la organización ejecutara. De manera colectiva, estas fases se conocen como el ciclo de vida del proyecto.

2.2.2 Características de las fases del proyecto. Cada fase del proyecto es marcada por la terminación de una o más entregas. Una entrega es un tangible, un producto de trabajo verificable tal como un estudio de factibilidad, un detalle de diseño, o un prototipo que trabaje. Las entregas, y por tanto las fases, son parte generalmente de una secuencia lógica diseñada para asegurar una definición apropiada del producto del proyecto.

La conclusión de una fase de proyecto es generalmente marcada por la revisión de tanto las entregas como del desempeño del proyecto para poder (a) determinar si el proyecto debe continuar a su próxima fase y (b) detectar y corregir errores de manera eficiente. Estas revisiones de final de fase generalmente se llaman salidas de fase, puertas de fase o puntos muertos.

Cada fase de proyecto normalmente incluye una serie definida de productos de trabajo diseñados para establecer el nivel deseado de control administrativo. La mayoría de estos ítems están relacionados con la entrega de la fase primaria, y las fases típicamente toman sus nombres de estos ítems: requerimientos, diseño, construcción, texto, comienzo, entrega, y otros como sea apropiado.

2.2.3 Características del ciclo de vida del proyecto. El ciclo de vida del proyecto sirve para definir el comienzo y el final de un proyecto. Por ejemplo, cuando una organización identifica una oportunidad a la que le gustaría responder, autorizará un estudio de factibilidad para determinar si debe adoptar el proyecto. La definición del ciclo de vida del proyecto determinará si el estudio de factibilidad es tratado como la primera fase de vida del proyecto o como un proyecto independiente.

La definición de ciclo de vida del proyecto determinará también que acciones de transición se incluirán al final del proyecto y cuales no. De esta manera, la determinación del ciclo de vida del proyecto puede ser usado para enlazar el proyecto a operaciones sucesivas de la organización ejecutora.

La secuencia de fase definida por la mayoría de los ciclos de vida del proyecto generalmente involucran algún tipo de transferencia en tecnología o intercambios tales como los requerimientos para diseñar, construcción para operaciones o diseño para manufactura. Entregas de la fase precedente son usualmente aprobadas antes que comience el trabajo en la fase siguiente. Sin embargo, una fase subsiguiente es a veces comenzada antes de la aprobación de las entregas de la fase anterior cuando los riesgos involucrados se tornan aceptables. Esta táctica de traslapo de fases muchas veces es llamada "Fast Tracking".

Las descripciones de los ciclos de vida del proyecto pueden ser o muy generales o muy detallados. Las descripciones altamente detalladas tienen muchas formas, tablas y lista de chequeo para proveer estructura y consistencia. Tales

aproximaciones de detalle son llamadas a veces metodologías de administración de proyectos.

La probabilidad de completar exitosamente el proyecto es más bajo, y por lo tanto el riesgo e incertidumbre son altos, al comienzo del proyecto. La probabilidad de completar exitosamente generalmente se vuelve progresivamente más grande a medida que el proyecto continúa.

La habilidad de los partidos interesados para influenciar las características finales del producto del proyecto y su costo final son más altas al comienzo y se vuelven progresivamente más bajas a medida que el proyecto continúa. La contribución más grande de este fenómeno es que los costos de cambio y de corrección de errores generalmente se incrementan a medida que el proyecto continúa.

Se debe tener cuidado para distinguir entre el ciclo de vida del proyecto y el ciclo de vida del producto.

2.3 EL DESARROLLO DE PROYECTOS DE SOFTWARE

2.3.1 Punto de partida: la WBS. La WBS o Estructura desglosada del Trabajo, es una técnica de planeación mediante la cual podemos definir y cuantificar el trabajo a realizar en todo el proyecto.

Es un proceso de pensamiento mediante el cual se pretende organizar el proyecto. Es similar al organigrama tradicional de una empresa, donde se tiene un Director, Subdirectores, Jefes de Departamento, Jefes de oficinas, etc.

Para lograr establecer la WBS del proyecto, tenemos que organizar nuestras ideas alrededor de lo pretendemos hacer en el proyecto, o sea definir el título del proyecto de software, que esté de acuerdo con la meta que se pretende alcanzar.

Seguidamente, se debe pensar en las grandes áreas de trabajo en que puede ser dividido, lo cual constituirá los paquetes de trabajo a desarrollar para lograr la meta.

Para el logro de este proceso en la elaboración de la WBS, se puede utilizar con éxito, una técnica de tormenta de ideas en un grupo de trabajo.

Posteriormente, cada uno de esos paquetes de trabajo, pueden examinarse también a través de la tormenta de ideas y elaborar un listado de actividades constituyentes de cada uno de los paquetes de trabajo. Tales actividades a su vez, pueden ser subdivididas hasta lograr el desglose necesario. El nivel de desglose requerido por el proyecto, estará determinado por la complejidad y tamaño del

proyecto. Pero se pueden considerar las siguientes recomendaciones para saber si se tiene el nivel de desglose necesario:

- a. Los paquetes de trabajo deben ser independientes unos de otros.
- b. Las actividades en el nivel de mayor desglose, deben ser medibles; esto es, que pueda establecerse un estimado de plazo de ejecución y recursos necesarios para llevarla a cabo, con la mayor precisión por parte del investigador más experimentado.
- c. Es necesario también que cada actividad se refleje en algo “tangible”, como puede ser: elaboración de un dibujo o plano, realización de un experimento, compra de un equipo, diseñar un programa, etc. De tal manera que pueda ser también cuantificable su avance real en la etapa de ejecución, seguimiento y control del proyecto.

La primera parte en la planeación del trabajo y para la elaboración de la WBS del proyecto, es necesario poner por escrito lo que serían las bases iniciales sobre las que se edificará el proyecto.

Un concepto interesante de mencionar ahora, es la CODIFICACIÓN de las actividades, pues es conveniente utilizar un código para identificar las actividades, que indique los diferentes niveles de desglose, por ejemplo:

1. Paquete de trabajo 1
1.1 Actividad 1 del paquete 1 1.1.1 Subactividad 1 de la actividad 1 del paquete 1 1.1.1.1 etc. 1.2 Actividad 2 del paquete 1 1.2.1 Subactividad 1 de la actividad 2 del paquete 1.
2. Paquete de trabajo 2
2.1 Actividad 1 del paquete 2 2.1.1 Subactividad 1 del paquete 2 2.1.1.1 Etc.

Y así sucesivamente. Se sugiere utilizar este método de codificación de los niveles de desglose, ya que este mismo método utiliza automáticamente el MS-Project, que será utilizado más adelante.

Como puede observarse, se pretende que en la WBS se incluyan TODAS las actividades que se van a desarrollar en el proyecto, así como también, se dejarán fuera, aquellas que no sean de su incumbencia.

Una vez establecida la WBS, se tendrá definida la cantidad de trabajo a realizar, totalmente organizado por áreas, paquetes o especialidades. También se podrá determinar un código identificador a través del catálogo de actividades.

En síntesis, como punto de partida, la WBS, permite definir el trabajo de lo general a lo particular en la etapa de planeación y cuantificar avances y recursos de lo particular a lo general, en la etapa de seguimiento y control del proyecto.

2.3.2 Esquema de una propuesta de modelo. La propuesta gerencial del desarrollo de proyectos de software que se elabora en este trabajo se basa en los estudios de Muench²⁵. Este describe un modelo espiral para desarrollo de software con cuatro ciclos y cuatro cuadrantes, que a continuación se explican.

A. Ciclo de prueba de concepto. Aquí se captura los requerimientos de la entidad educativa, empresa o negocio. Se define metas para la prueba del concepto. Se produce diseños conceptuales de sistema, diseño y construcción de la prueba del concepto. Así mismo se produce planos para el ensayo de la aceptación y conduce a análisis de riesgo, haciendo recomendaciones.

B. Ciclo de la primera construcción. En esta parte se define los requerimientos del sistema y las metas para la primera construcción. Se produce diseños de los sistemas lógicos y la construcción del primer modelo. También se elaboran planos para los ensayos del sistema, se evalúa la primera construcción y se hacen recomendaciones.

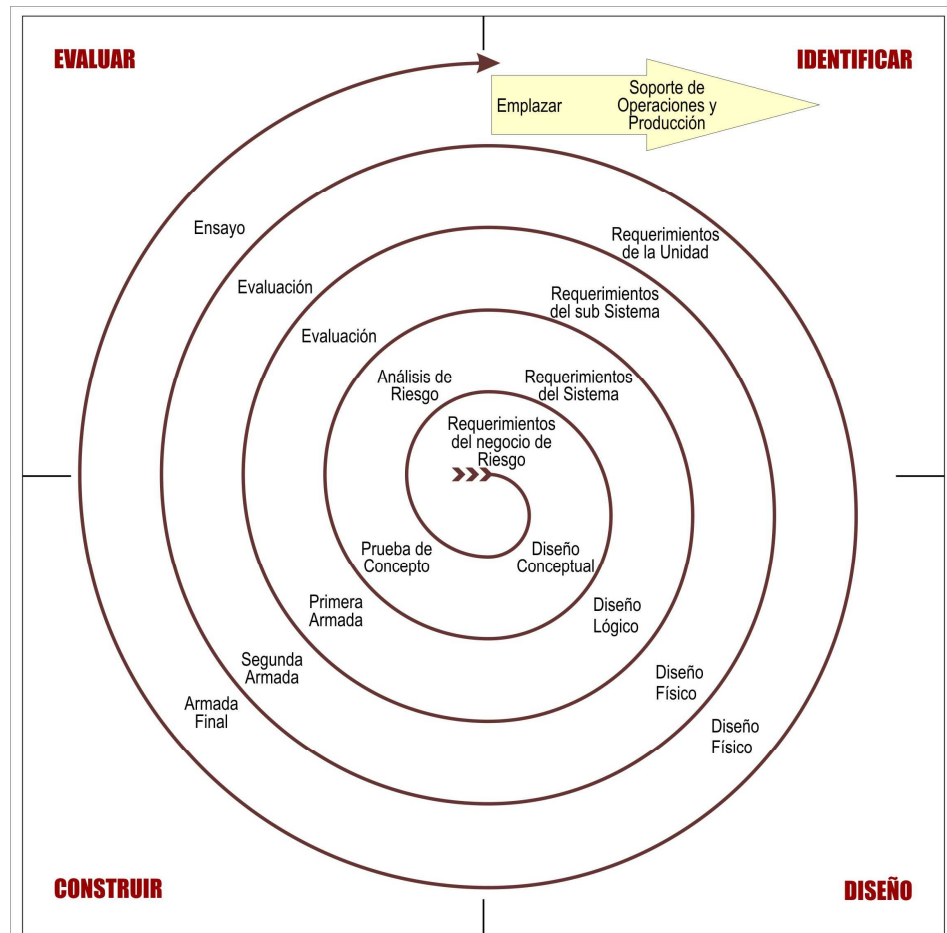
C. Ciclo de la segunda construcción. Implica requerimientos del sistema, definir metas para la segunda construcción, producir diseños físicos, construir el segundo modelo, producir planos para los ensayos del sistema, evaluar la segunda construcción y hacer recomendaciones.

D. Ciclo final. Esta parte completa los requerimientos del proyecto, como son el diseño final, construcción del modelo final, hacer los ensayos de unidad, subsistema, sistema, y se procede a la aceptación.

²⁵ MUENCH, Dean. *The Sybase Development Framework*. Oakland Calif. Sybase Inc., 1994.

El ciclo de vida así explicado, según Muench, se esquematiza en la figura 2.

Figura 2. Ciclo de vida de desarrollo de software

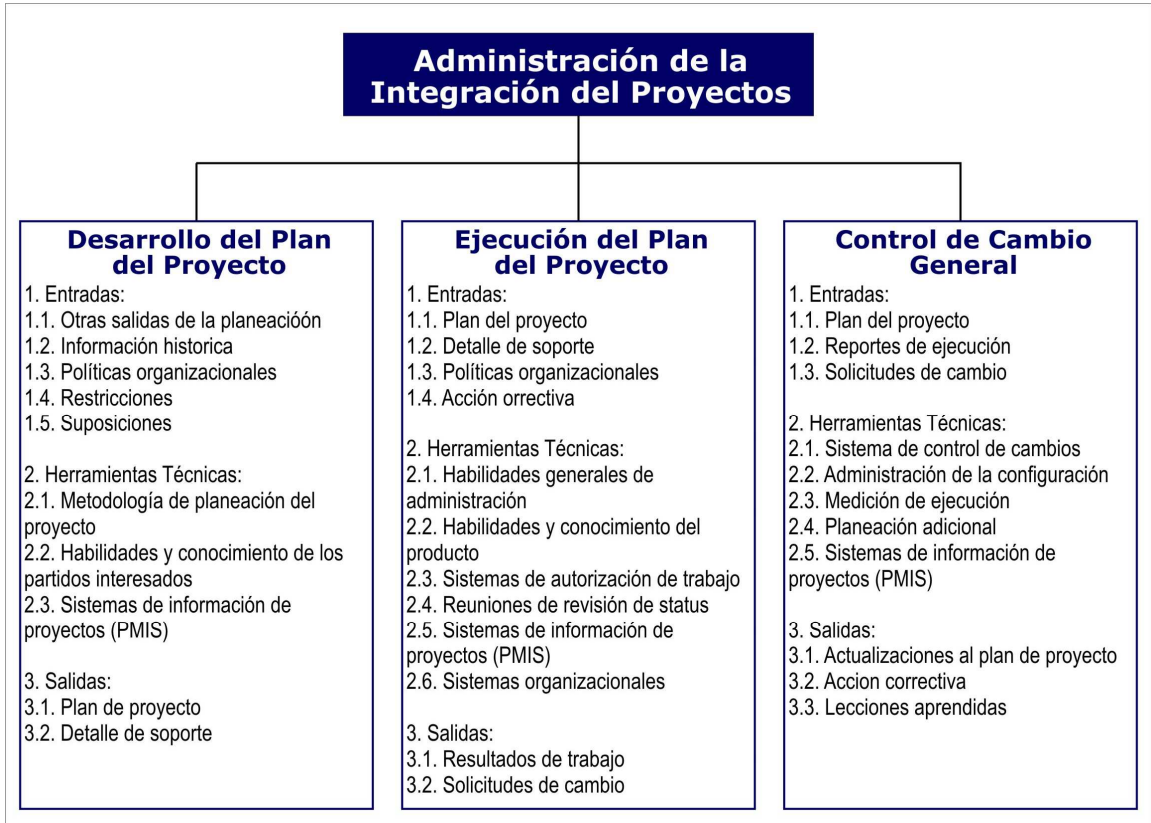


Fuente. Esta investigación

2.3.3 Esquemas esenciales del proceso. Es aconsejable en la construcción de modelos para los proyectos de software, en términos de la excelencia gerencial, recalcar en unos procesos que son fundamentales. Son ellos los que se refieren a la administración de la integración, del tiempo, de los costos, de la calidad, de las comunicaciones, del riesgo y del procuramiento del proyecto.

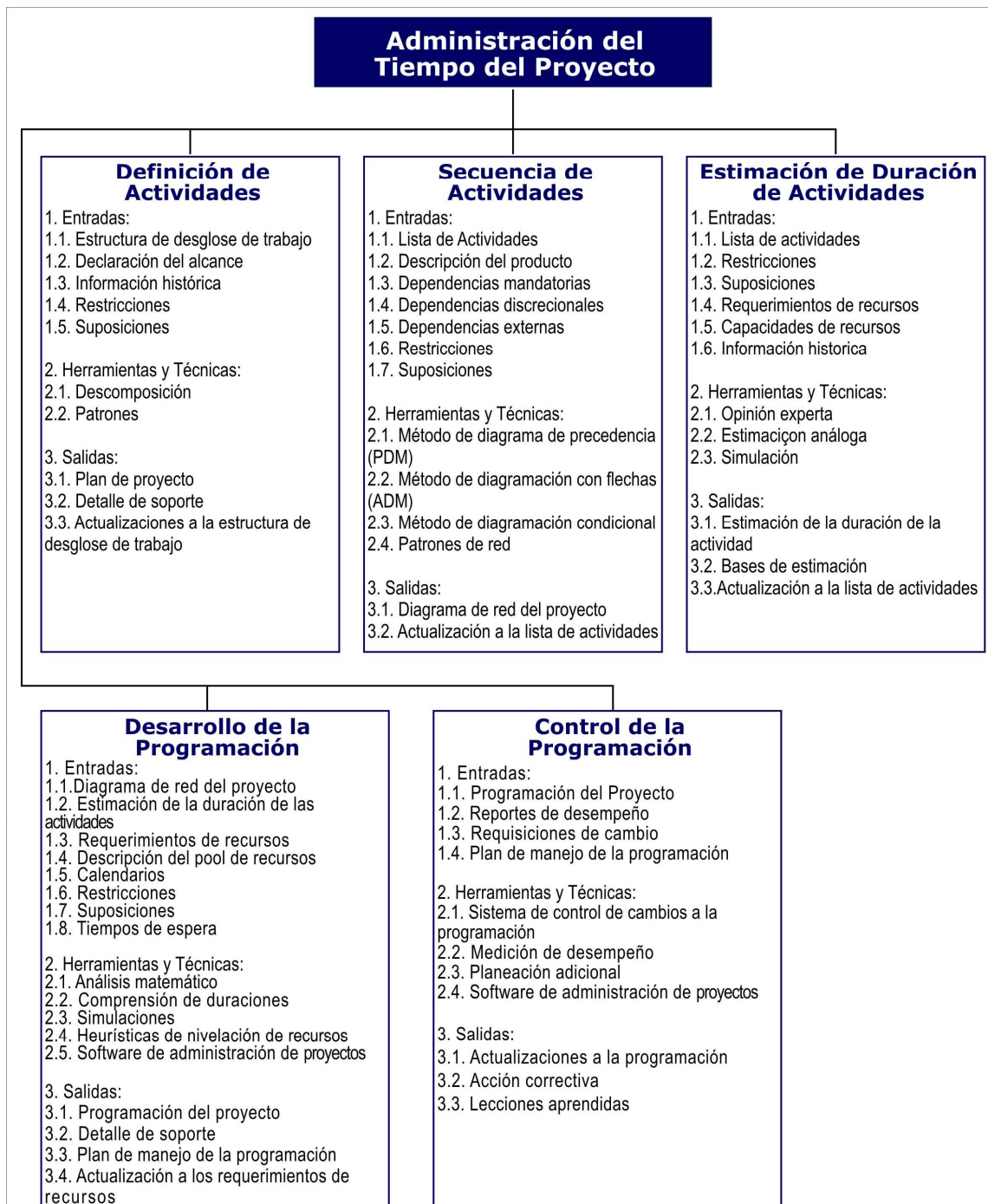
Para ser gráficos se esquematizan estos en procesos en las figuras 3 a 9.

Figura 3. Vista general de la administración de la integración del proyecto



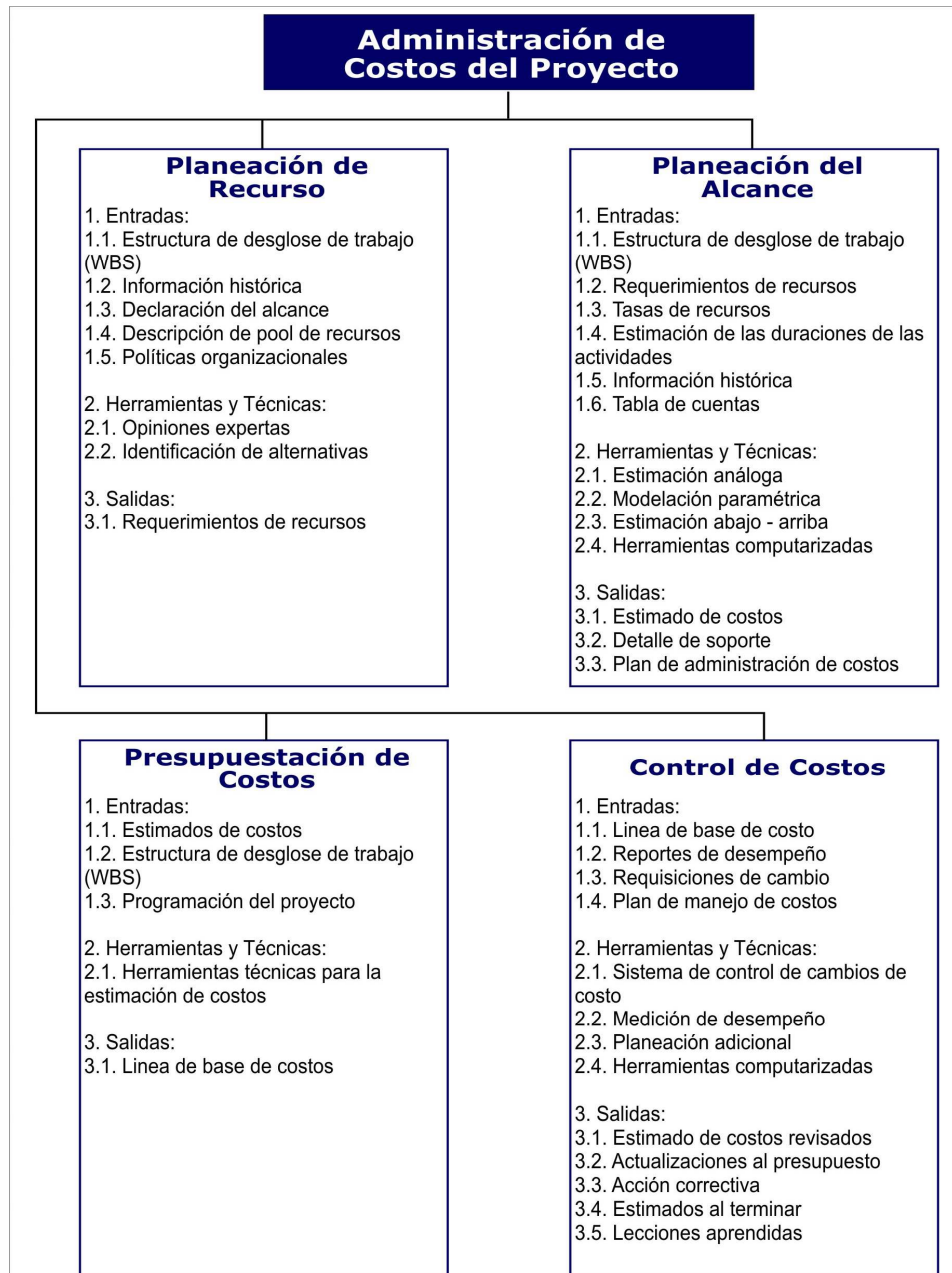
Fuente. Esta investigación

Figura 4. Vista general de la administración del tiempo del proyecto



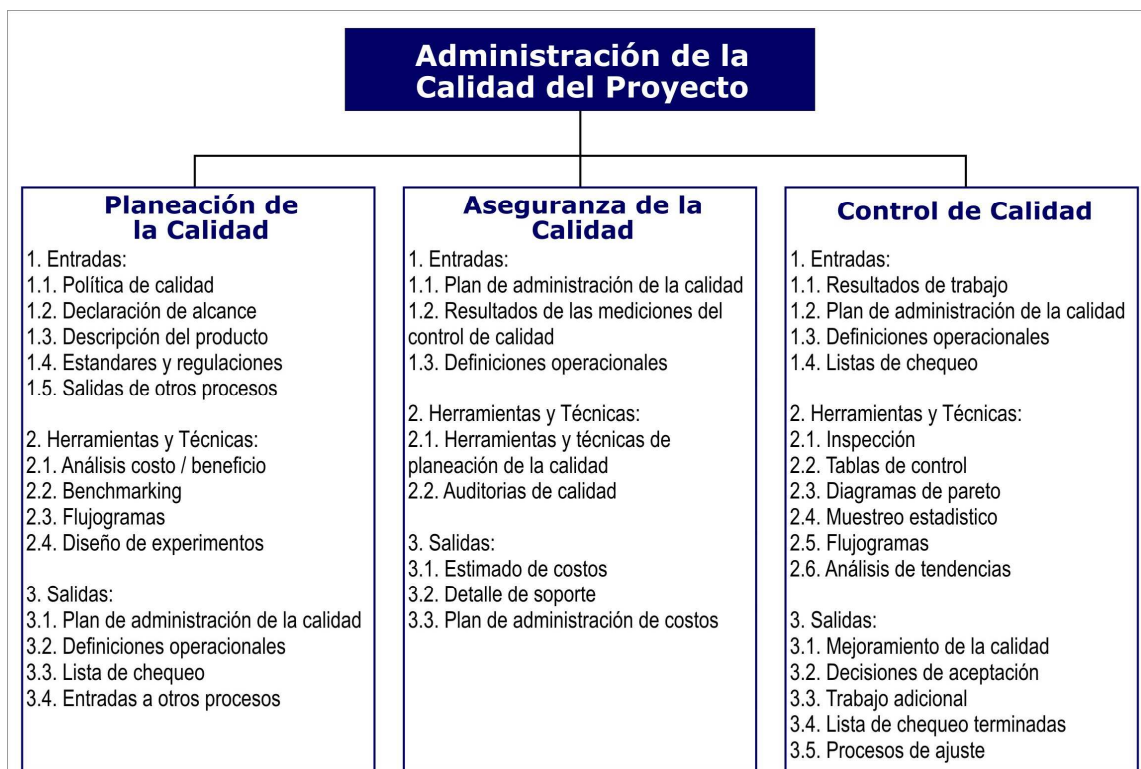
Fuente. Esta investigación

Figura 5. Vista general de la administración de costos del proyecto



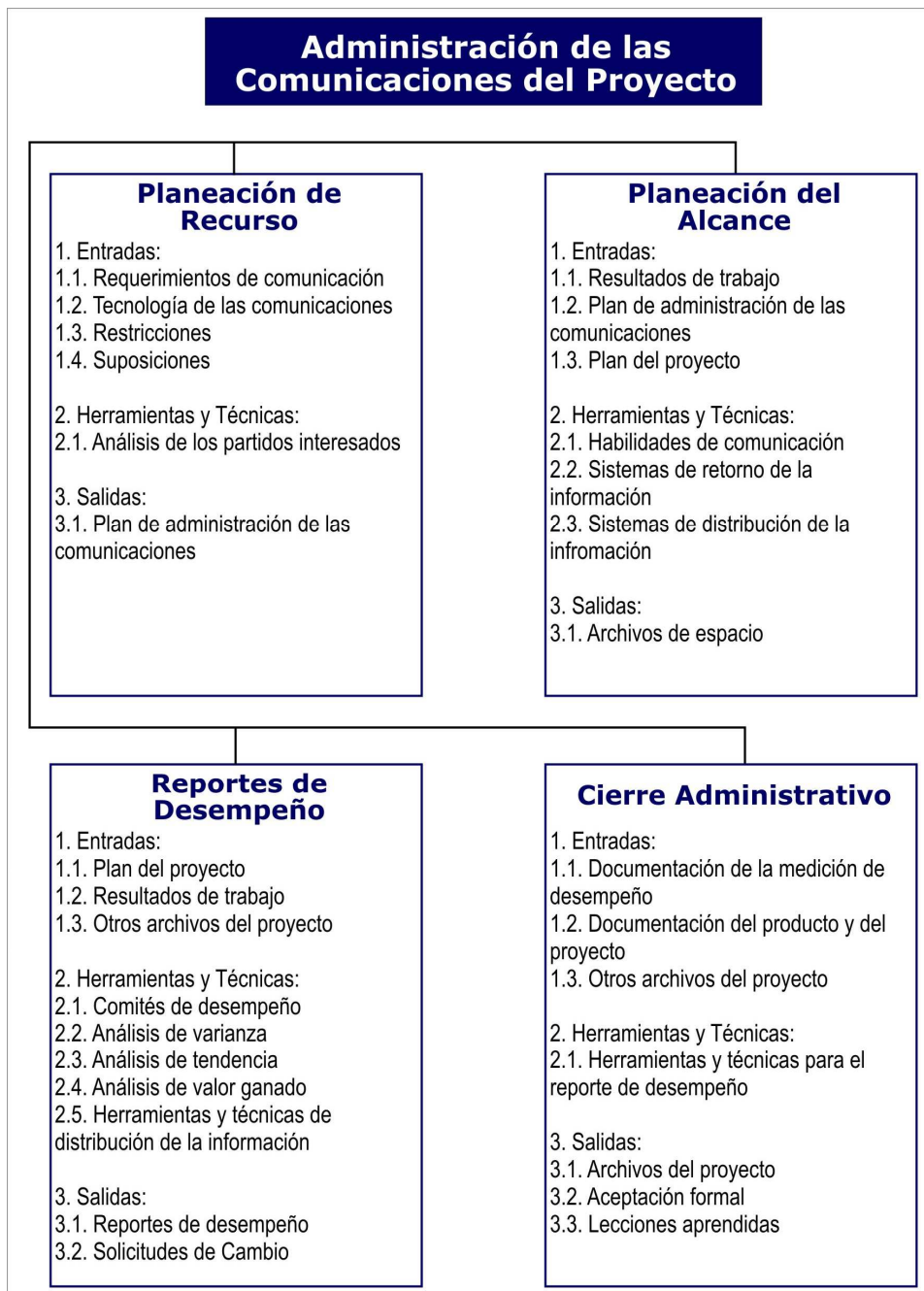
Fuente. Esta investigación

Figura 6. Vista general de la administración de calidad del proyecto



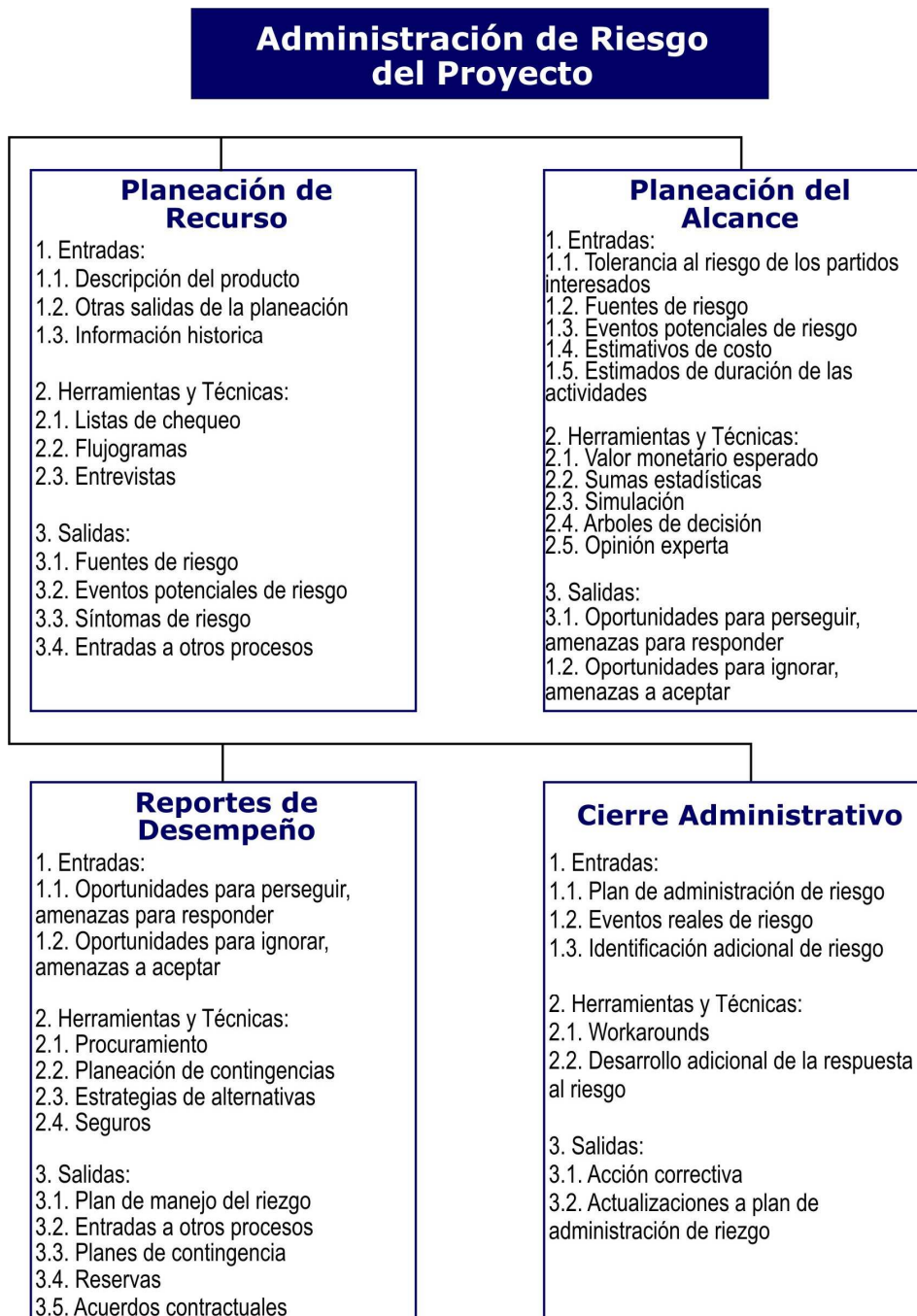
Fuente. Esta investigación

Figura 7. Vista general de la administración de las comunicaciones del proyecto



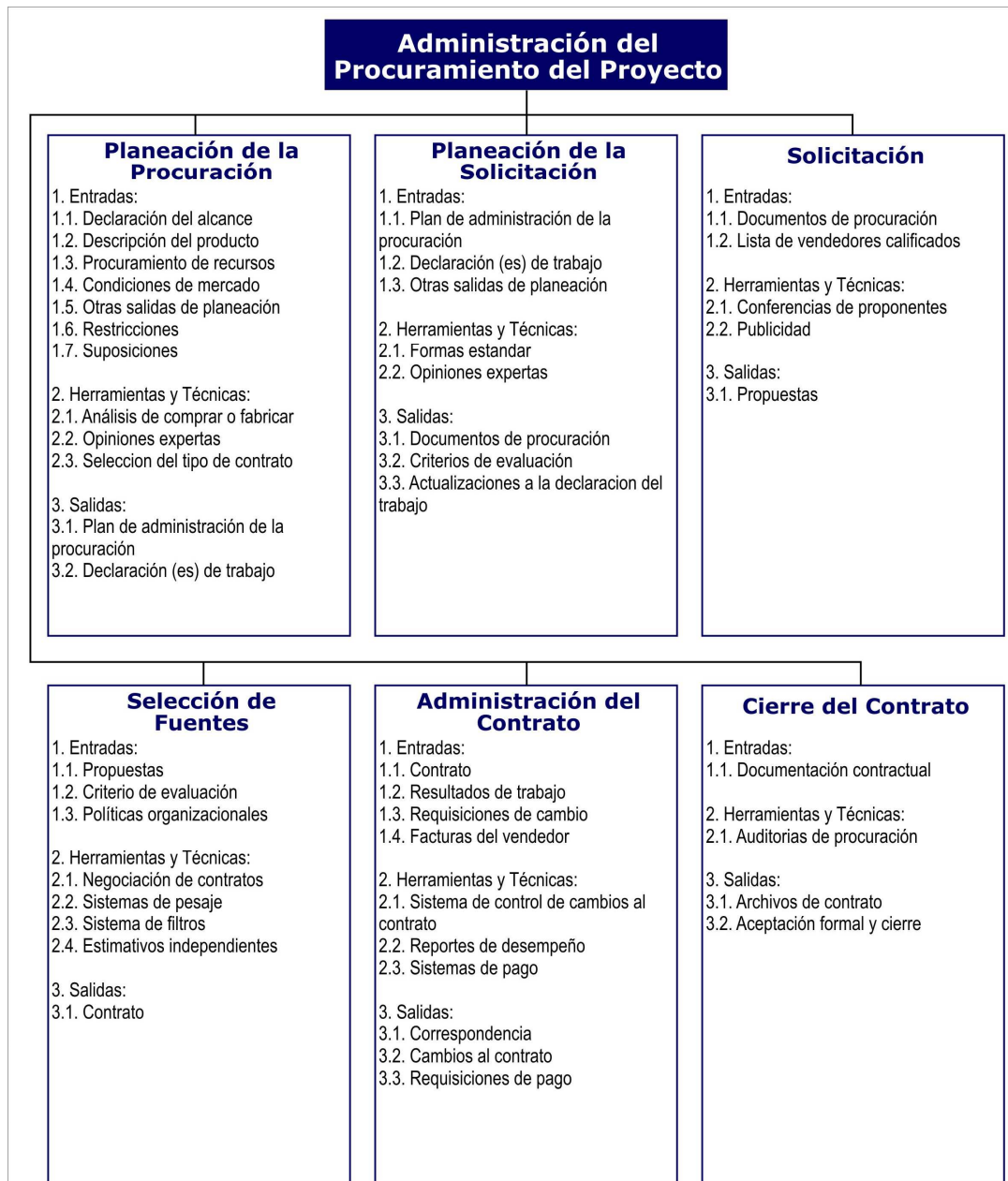
Fuente. Esta investigación

Figura 8. Vista general de la administración de riesgo del proyecto



Fuente. Esta investigación

Figura 9. Vista general de la administración del procuramiento del proyecto



Fuente. Esta investigación

2.3.4 Partidos interesados en el proyecto. Se define como los partidos interesados, a los individuos y organizaciones que están activamente interesados en el proyecto de software, o cuyos intereses pueden ser afectados positiva o negativamente como resultado de la ejecución o de la terminación exitosa del mismo. En este trabajo hacen referencia a las instituciones de Educación Superior.

El equipo de administración del proyecto de software debe identificar a los partidos interesados, determinar cuales son sus necesidades y expectativas, y administrar e influenciar esas expectativas para asegurar un proyecto exitoso. La identificación de los partidos interesados en el proyecto es a veces difícil.

Los partidos claves en cada proyecto incluyen:

Administradores de proyectos: individuos responsables por administrar el proyecto.

Cliente: individuo u organización que usará el producto del proyecto. Puede haber múltiples capas de clientes. En nuestro caso son las Universidades.

La organización ejecutora: la entidad cuyos funcionarios están más directamente en el trabajo del proyecto.

Administrar las expectativas de los partidos interesados puede ser difícil porque estos muchas veces tienen objetivos muy distintos, que pueden entrar en conflicto.

2.3.5 Un caso típico universitario: el software educativo. Como un caso práctico, uno de los más comunes en instituciones de educación, y sobre todo a nivel superior, es interesante hacer unas anotaciones a los denominados software educativos, que son fuente abundante de desarrollo de proyectos.

Para la elaboración de software educativo se debe seguir una metodología de desarrollo que se compone de las siguientes fases: Conceptualización, Pre - Producción, Producción y Post - Producción.

Conceptualización. Durante la fase de conceptualización del software educativo, el equipo humano genera las ideas para el logro de ambientes explorativos y desarrollo de habilidades de pensamiento, que estimulan el fortalecimiento de aptitudes y actitudes frente a la creación, investigación y el trabajo en equipo, de colaboración, alrededor de los valores culturales.

Así mismo, se perfila una adecuada imagen del usuario y su medio, en la cual se reúne una descripción psicológica que plasma sus niveles de desarrollo, aspectos cognitivos y capacidades de abstracción, así como el medio ambiente social y cultural en el cual se desenvuelve.

El contexto construido se toma como base para determinar, de acuerdo con la necesidad planteada, el contenido (temas - subtemas) y su tratamiento (recursos gráficos y lingüísticos, aspectos y estilos motivacionales, características de la interfaz humana - computador, las metáforas a utilizar y las actividades a realizar), concretando así la estructura conceptual.

En esta fase, se crea la propuesta y se estudian las diferentes alternativas de estructura, navegación e interacción teniendo en cuenta los aportes realizados desde la investigación gráfica, como gran elemento de trabajo.

En esta fase fundamental para esta clase de software se recomienda tener presente los siguientes puntos:

A. Condiciones a tener en cuenta en el desarrollo de sistemas gráficos. En este campo se deben considerar aspectos como los siguientes:

- La ergonomía visual.
- La coherencia conceptual.
- Los niveles de interacción.
- La navegación.
- La comunicación.
- Los principios multimediales y pedagógicos.

Estos elementos deben ser claros en las formas de relacionarlos a través de metáforas con el tema a tratar y el público objetivo al cual va dirigido cada software educativo.

B. Creación y representación de metáforas. Se trata de diversificar el proceso, teniendo en cuenta cada uno de los proyectos de software educativo.

Las metáforas con el desarrollo de los software se han materializado a través de una amplia gama de técnicas gráficas. Con ello, se ha buscado propiciar una experiencia altamente interactiva y transformadora para la comunidad educativa, logrando la apropiación y el reconocimiento cultural, dando a la interfaz un sentido de escenario donde tanto el alumno como el maestro se identifican y se convierten en protagonistas.

Pre – producción. Con la información construida, el equipo de trabajo entra a la fase de pre - producción para formalizar el proyecto, materializando la propuesta en elementos preceptuales, metodológicos y técnicos.

Los diseñadores conceptualizan y proponen las diferentes composiciones de las interfaces relacionándolas en el ámbito formal y funcional con el usuario.

Con apoyo de los comunicadores, se escriben las propuestas, narraciones explicativas, diálogos de actores y pantallas de texto para dar mensajes.

Los pedagogos y diseñadores de contenido participan activa y conjuntamente con diseñadores y comunicadores en todo el proceso creativo.

Los ingenieros determinan las herramientas, el porqué de su uso y toman decisiones acerca de lenguajes, formatos, bases de datos y otros elementos de acuerdo con la conceptualización y el perfil del usuario. Sus determinaciones son compartidas con el resto del equipo de trabajo. Como resultado de esta acción se genera el guión (story board) en el cual se formaliza y consolida la idea a materializar.

Producción. Posteriormente, se pasa a un proceso de evaluación y definición de medios para dar paso a la fase de producción de la propuesta. El equipo, entonces, pasa a realizar las respectivas digitalizaciones, ediciones gráficas, la producción y edición de medios, los links entre pantallas, bases de datos, ediciones de texto y audio.

Durante esta etapa se alimenta permanentemente el guión y el mapa de navegación, al igual que la documentación o bitácora de desarrollo del proyecto.

Post – producción. Una vez materializada la idea, durante la fase de post - producción, se realizan una serie de pruebas y revisiones por parte del equipo de trabajo y también con usuarios potenciales, con el fin de realizar los análisis correspondientes al producto vs. el público objetivo, proponiendo y desarrollando los ajustes que, desde todas las disciplinas, se consideran necesarios para obtener el producto deseado.

En la fase final del proceso, se entra a generar las estrategias de capacitación y preparación de materiales para la posterior entrega de los productos.

Es interesante anotar al respecto que la interacción es la palabra clave que diferencia los programas de aprendizaje con las aplicaciones de las presentaciones. El usuario puede controlar por sí mismo el camino que el programa debe seguir naturalmente, dependiendo de las posibilidades que el programa ofrece.

De especial importancia es la interacción en relación con los programas de aprendizaje, de tal forma que la transmisión de conocimientos se produce a través del diálogo con los estudiantes. Una transmisión de conocimientos a través de una presentación convencional, presupone un nivel similar de conocimientos y una misma capacidad de asimilación de información, por parte de los espectadores. Los programas interactivos de aprendizaje están estructurados de tal forma, que su base de partida puede escogerse libremente.

Cada información puede presentarse tanto tiempo y tantas veces sea necesario para asimilar el conocimiento que se transmite. Sólo entonces se pasará a la siguiente página (pantalla o escena). Las preguntas de control y ejercicios controlados por el programa redondean este método. Dependiendo del nivel de respuestas correctas, se repetirá, por ejemplo, el último capítulo (módulo) o se pasará al siguiente. El usuario puede interrumpir el programa en cualquier momento y en cualquier punto y continuar en el mismo lugar posteriormente, en un proceso denominado Análisis de Ancla.

En los programas de aprendizaje no sólo es importante la preparación multimedia de la idea o materia, sino su estructura didáctica. Al usuario le ayuda si los temas están bien estructurados y las preguntas de control resaltan adecuadamente los aspectos fundamentales. De gran interés resulta la utilización de amplios materiales gráficos, así como animaciones y archivos de sonido para presentar la realidad.

También la estructura en forma de evaluaciones, en la que los usuarios puedan ir acumulando puntos a través de la respuesta correcta hace más atractivos los programas de aprendizaje.

La transmisión de información de un programa de aprendizaje mal presentado puede ser compensada con la ayuda de instructores especializados. Pero mientras mayor sea la independencia a que se aspire en el estudio con un programa de ese tipo, más elevadas tendrán que ser sus exigencias de calidad.

Los programas de aprendizaje deben cubrir las necesidades demandadas de información. Con el ahorro que se logra, al eliminar los costosos seminarios de actualización y clasificación, debido a los cuales es necesario desplazar a los profesores, trabajadores de sus puestos de trabajo e incurrir en gastos directos e indirectos, se amortizan con un buen software de aprendizaje. Si además, los computadores están conectados en red, entonces es posible garantizar todas las informaciones más actualizadas de forma global.

Un programa de aprendizaje requiere de un mínimo de tranquilidad y una concentración efectiva en la materia de estudio.

Puntos destacados. En el diseño de un proyecto de software educativo se deben distinguir los siguientes pasos:

Paso 1: Esquematización o análisis de necesidades. En el primer paso se trata, fundamentalmente, de definir el esquema de aprendizaje y cómo estructurarlo. El esquema, por supuesto, tiene que ajustarse a la necesidad o materia a transmitir.

Paso 2 : Objetivos. En los objetivos se definen los requerimientos del solicitante. Aquí deben quedar muy claro cuáles son los conocimientos previos que pueden ser asumidos, qué temas deben incluirse y, no menos importante, cuál es el presupuesto.

Paso 3 : Recolección de materiales. Como el programa debe caracterizarse por un enfoque práctico de la materia o necesidad a estudiar o resolver, debe realizarse una correspondiente recolección de materiales que apoyen este aspecto. En temas específicos de una empresa, instituto, colegio se debe conseguir imágenes y videos de la misma. También puede incluirse informaciones de los manuales y de materiales adicionales. Archivos de sonido que reproduzcan el entorno, descripción o narración del tema tratado.

Paso 4 : Guión. Este paso es fundamental en la fase de diseño, dado que aquí se materializa el concepto didáctico.

Surgen así interrogantes como los siguientes:

- ¿Qué temas deben tratarse primero?
- ¿Cómo se plantearán los temas?
- ¿Hasta dónde se profundizará la materia?
- ¿Cuáles serán las metodologías de evaluación?

2.3.6 Propuesta básica de un software educativo. Las anteriores consideraciones ponen de presente la necesidad de tener en cuenta el contexto de desarrollo del software educativo y por lo tanto, la posibilidad de plantear más de un modelo, de tal manera que se responda a la variedad de tipos de software utilizado, los diferentes enfoques pedagógicos y herramientas y plataformas técnicas que se pueden utilizar. Por ello, a continuación, se plantean los criterios generales y actores a tener en cuenta en los modelos de desarrollo y finalmente, algunas consideraciones puntuales sobre el proceso mismo de desarrollo.

En la propuesta básica se tienen en cuenta los criterios para desarrollo de este tipo de software, supremamente útil para la Universidad del siglo XXI, los componentes para los modelos, las plataformas para el desarrollo, el equipo de apoyo, las formas y procesos de desarrollo, la presentación de información y los actores del modelo.

A. Criterios de desarrollo de software

TIPO DE MODELO CRITERIOS	SOFTWARE PROTOTIPO	SOFTWARE ESTRUCTURADO	SOFTWARE EVOLUTIVO
Necesidad que lo origina	Necesidades específicas de aprendizaje.	Distancias geográficas y de tiempo.	Compartir información para construir conocimiento.
Componentes del modelo	Pedagógico, Tecnológico, Disciplinar, Gráfico.	Pedagógico, Tecnológico, Disciplinar, Comunicativo, Administrativo.	Multicomponente. Sin definir.
Formas de presentación de información	Medios expositivos Medios activos.	Medios expositivos, activos e interactivos.	Medios activos. Medios interactivos.
Plataformas utilizadas	Software propietario. Software libre.	Software propietario. Software libre.	Software libre.
Tipo de producto	CD Multimedial	Objetos y cursos virtuales	Blogs personales, escritos colaborativos.
Forma de desarrollo	Tipo catedral	Tipo catedral	Tipo bazar
Equipo de Desarrollo	Desarrollador independiente Equipo de Producción.	Equipo de Producción.	Equipo abierto.
Proceso de Desarrollo	Por proyecto.	Por proyecto.	Colaborativo
Tipo de financiación	Compromiso individual. Institucional.	Institucional.	Compromiso individual.
Medio de difusión	CD	Internet	Internet
Tipo de contenido	Cerrado (Contenidos protegidos por Derechos de autor y Derechos de comercialización).	Cerrado (Contenidos protegidos por Derechos de autor y Derechos de comercialización). Abierto (Público, multiautor)	Abierto (Público, multiautor)
Usuarios del modelo	Docentes, especialistas, estudiantes de primaria, secundaria, media, universitaria)	Estudiantes de media, universitarios de pregrado y postgrados, educación permanente y no formal.	Todo tipo de persona.
Usos metodológicos.	Grupos formales.	Redes de aprendizaje.	Comunidades virtuales.

Fuente. Esta investigación

El proceso de desarrollo de software educativo, se fundamenta en unos criterios generales que orientan el proceso y permiten establecer diferencias entre, al menos, tres categorías de software educativo. Estos criterios se pueden dividir entre aquellos encaminados al proceso mismo de desarrollo y otros dirigidos a determinar las características de los actores que intervienen en el modelo. Dentro de los primeros se encuentran los siguientes:

B. Componentes de los modelos.

El desarrollo de software Prototipo y estructurado, deben incluir consideraciones acerca del modelo pedagógico, comunicativo y técnico entre otros. La determinación del modelo pedagógico, al igual que la modelación del estudiante, evoluciona a partir de términos como modelo de instrucción o currículo instruccional, utilizados para designar uno de los elementos del diseño de tutores inteligentes. (Kaplan y Rock, 1995) Originalmente era entendido como el conjunto de instrucciones, mensajes y reglas que guiaban y controlaban la interacción del usuario con el tutor.

Sin embargo, en el diseño de software educativo, estos términos se amplían en el sentido de que se alejan de los elementos instruccionales-conductistas y adquieren una connotación más amplia y cercana al proceso pedagógico, el cual se apoya en concepciones de hombre, sociedad, conocimiento, aprendizaje y en elementos de tipo metodológico y didáctico. Bajo esta perspectiva, el modelo pedagógico, como patrón, está apoyado en una corriente pedagógica determinada que se ocupa de involucrar en el diseño aspectos propios de su concepción.

El proceso de producción de software educativo, contiene en su interior una concepción de comunicación a partir de la cual se determinan los criterios de interactividad e interacción del proceso, los aspectos de inducción e integración de contenidos, definición del público, de la estrategia, de los medios, del diseño a utilizar en los procesos pedagógicos y los procedimientos de generación de programas, así como la elaboración de guiones, producción y edición de materiales. De igual manera determina las estrategias de implementación, los indicadores, la utilización de medios y los aspectos fundamentales de la comunicación, como la caracterización de emisores, receptores, mensajes, diseño de medios, tipos de responsabilidad comunicativa e informativa y generación de acciones.

El componente tecnológico contiene los criterios tecnológicos y técnicos que orientaran el desarrollo de software, como por ejemplo la producción, actualización e interactividad de materiales, la administración del sistema, el proceso de programación del software, liberación y realización de posteriores versiones. Este componente se apoya en la elaboración de mapas de navegación entre las partes del software especificando las rutas de enlace entre ellas, bien sea por medios de mapas conceptuales o redes semánticas. De igual manera se utiliza el guión de

pantallas, es decir, un cuadro de resumen guía en la realización del software y que contiene, discriminado por cada una de las pantallas a elaborar, los elementos que se utilizarán en ellas.

Por su parte, en el software de tipo evolutivo, estos componentes se entretajan por parte de los participantes, junto con otras intencionalidades de interactividad, en las cuales se pueden observar elementos culturales y sociales que no se presentan de manera formal.

C. Plataformas de desarrollo.

En las categorías de software prototipo y estructurado, se puede observar el manejo de plataformas propietarias que permiten la producción de software multimedia y en red, que permiten la exposición de información y el desarrollo de acciones por parte del usuario. Sin embargo el software libre empieza a ser utilizado en la programación de software estructurado como una alternativa que permite configurar una plataforma con elementos institucionales y sin costos de adquisición, aunque con costos de manejo y adecuación permanente. Este tipo de plataforma, además permite el uso de herramientas comunicativas para generar procesos de interactividad entre los usuarios, lo cual también es una de las características del software evolutivo.

Por su parte, el software de carácter evolutivo, en este momento está utilizando casi exclusivamente el software libre y los procesos de interactividad entre los participantes del proceso.

D. Equipo de apoyo.

Mientras que el software prototipo, gran parte de la responsabilidad de proceso de desarrollo recae sobre el equipo técnico de producción, con una participación muy puntual de pedagogos y diseñadores, en la producción de software estructurado para distribuir en línea, empieza a cobrar importancia el apoyo de un equipo multidisciplinar encargado tanto del diseño del software como de su desarrollo. Este equipo está conformado por diseñadores instruccionales, encargados de la estructuración de las actividades de aprendizaje a incluir en el software, comunicadores y diseñadores encargados del proceso comunicativo y de diseño de interfaz, además de un equipo técnico de manejo de medios y programación, además de un administrador del proyecto, quien dirige y coordina las acciones del desarrollo.

El proceso de desarrollo por su parte, cuenta con un equipo conformado por personas interesadas en las temáticas de trabajo contenidas en el software, y con unos conocimientos mínimos de programación y diseño, dado que no es necesario un diseño muy elaborado. Esto implica que el equipo de desarrollo es

completamente abierto y las condiciones de participación están dadas fundamentalmente por el interés de sus miembros.

E. Formas y procesos de desarrollo.

Tanto el software de carácter prototipo como estructurado obedecen a formas de producción a un modo similar a la manera como se construye una catedral: con unos objetivos previamente definidos, diseños complejos y con partes cuidadosamente relacionadas entre sí, con materiales y procesos manejados por unas pocas personas altamente especializadas, responsables de procesos particulares, en donde unos pocos tienen las claves para realizar cambios y con actualizaciones espaciadas en el tiempo. Este tipo de producción, responde a la una metodología de producción por proyecto, con objetivos, etapas, responsables, materiales y cronogramas precisos.

Por su parte, el software evolutivo responde a una forma de producción similar a la manera como se construye un bazar: con un grupo de personas, con objetivos particulares que tienen acceso a todos los materiales y pueden hacer cambios y publicaciones dentro del software son ningún tipo de restricción. La autoría es compartida y la participación abierta, lo cual obedece a un tipo de producción colaborativa, en la cual todos los participantes son responsables de la producción y comparten la autoría del mismo.

F. Presentación de información y tipos de contenido.

El software prototipo y estructurado, poseen formas de presentación de información de carácter expositivo, es decir, se presentan la información en una estructura en la que el estudiante puede navegar y acceder a los vínculos temáticos propuestos. El modelo de aprendizaje utilizado para este medio centra su atención en el maestro, pues él quien determina y controla los contenidos, actividades propuestas y niveles de aprendizaje.

De la misma manera, presentan formas de presentación de información interactiva, es decir, que utilizan en un modelo de aprendizaje centrado en el estudiante, quien interpreta la información, enriquece su estructura de conocimiento, sus procesos cognitivos y las posibilidades de aplicación práctica, a partir de las experiencias individuales de aprendizaje que ha diseñado el maestro.

El software estructurado y evolutivo, utilizan medios interactivos que permiten la comunicación entre quienes participan del proceso de forma sincrónica o asincrónica, para hacer posible la conformación de grupos que se dedican a la generación de conocimiento. Estos medios se centran en el aprendizaje colaborativo del grupo.

En consecuencia, el software prototipo y estructurado presentan contenidos Cerrados, con autores reconocidos y protegidos por Derechos de autor y Derechos de comercialización, mientras que el software evolutivo presenta contenidos abiertos, en los cuales es difícil establecer la autoría, debido a las colaboración permanente y la multiplicidad de usuarios que participan en la producción.

G. Actores del modelo de desarrollo.

El estudiante: eje central del proceso. En cualquiera de los tipos de software a desarrollar, es esencial la participación activa del estudiante, bien sea como un elemento que permite modelar la acciones y procesos que debe desarrollar en un prototipo determinado, como actor autónomo del proceso de aprendizaje que determina no solo los tiempos y contenido de aprendizaje en un ambiente estructurado, sino también como evaluador y participante en la configuración de las actividades de aprendizaje y autor de los contenidos en el software de carácter evolutivo.

El papel del docente. Es claro que en esta mirada de desarrollo de software educativo, el docente abandona su papel de usuario del mismo, para convertirse en un diseñador de ambientes de aprendizaje prototípicos o estructurados, y como participante activo y autor de software evolutivo.

El docente ya no es exclusivamente un poseedor de información, su papel en el proceso educativo traspasa la transmisión de conocimiento y llega a los límites de la tutoría de los estudiantes, de la asesoría permanente, de la orientación de procesos y de un papel de par académico que tiene bajo su responsabilidad, e proceso de colaboración en el aprendizaje de los estudiantes y de estos entre si.

Los administradores de salas e ingenieros de soporte. Esta mirada de desarrollo de software educativo, implica que los administradores de sala e ingenieros de soporte abandonan su postura de poder frente al manejo de programas, para convertirse en facilitadores de los procesos de desarrollo de ambientes propicios para el aprendizaje, que participan de forma colaborativa en los procesos de desarrollo de software poniendo sobre el tapete todos los elementos técnicos, de manejo de programas y redes, necesarios para los procesos de aprendizaje de los estudiantes. En cambio de la búsqueda permanente de restricciones, se convierten en posibilitadores de procesos.

Directivos y administración universitaria. Los procesos de desarrollo de software educativo deben responder a las necesidades del contexto universitario y los lineamientos determinados desde el proceso de planeación de las instituciones. Por lo tanto el papel de los directivos y administradores académicos de las instituciones de educación superior es facilitar estos procesos de

producción, ofreciendo todas las posibilidades para el desarrollo de las tres categorías de software descrito.

2.4 LA DIRECCIÓN INTEGRADA DE PROYECTOS EN RELACIÓN A LA CALIDAD CON EL USO DEL SOPORTE INFORMÁTICO PROJECT 2002.

2.4.1 Conceptos necesarios. La Dirección Integrada de Proyectos o Project Management, tiene como objetivo principal la obtención de los resultados programados, acortando los plazos de tiempo, en el marco del presupuesto y con la calidad requerida. De estos tres elementos, el tiempo es el que con más frecuencia se controla en la programación.

El costo generalmente se tiene en cuenta en la primera etapa con la elaboración del presupuesto y en la etapa de ejecución se debe controlar a través de un sistema de centro de costos por proyecto eficiente y una contabilidad confiable en los cortes.

La calidad en el proceso de ejecución, tiene una incidencia determinante tanto en el costo como en el tiempo,

La Dirección Integrada de la Calidad (DIC) y su vinculación con la ejecución de proyectos garantizan el cumplimiento de los objetivos. La ISO 9000 brinda un conjunto de conceptos sobre el tema. Al respecto se deben aclarar algunos.

Calidad total. Diseñar los objetivos del proyecto en función de las necesidades del cliente, garantiza el establecimiento de esta filosofía en el sistema para hacer lo que hay que hacer en función de las necesidades del cliente.

Esta filosofía fue elaborada a partir de los conceptos generados por los profesores occidentales Deming y Juran e impulsada fuertemente por los japoneses con las aportaciones del Profesor Ishikawa.

Enfoque sistémico. La calidad es un sistema que parte de la base normalizadota, se apoya en el control petrológico y evalúa los resultados con el enfoque sistémico que brinda el diseño estructurado del proyecto.

Dirección Integrada de la Calidad. Esta incide en todos los procesos del proyecto desde la concepción, el control de la documentación del mismo, evaluación de alternativas, control de ejecución, toma de decisiones y puesta en explotación aplicando el concepto integrado de la estructura funcional en apoyo al proyecto.

En el desarrollo del proyecto la dirección integral de la calidad en función de garantizar tanto los objetivos fundamentales del proyecto como los objetivos

parciales permite precisar en cada etapa y en cada hito los criterios de medidas en función de la base normalizativa vinculada a las tareas.

En la Dirección Integrada de la Calidad el control garantiza el cumplimiento de los objetivos a partir del control cualitativo recogido en las tareas e integrado en los hitos.²⁶

Estructura de la calidad. La calidad se apoya en la estructura del proyecto y la estructura funcional de la empresa. La dependencia de la estructura funcional de la empresa que atiende la gestión de la calidad no debe estar comprometida con la ejecución de los proyectos y debe responder a la dirección de la empresa. Cada participante de la estructura funcional actualiza en el sistema informático la evaluación de la calidad ponderada a partir de los acuerdos correspondientes de la ponderación de la calidad en función de los acuerdos tomados en las curvas características de la calidad haciendo uso de la ética y los valores. El director de proyecto se apoya en las evaluaciones depositadas en el sistema informático por los ejecutores para la toma de decisiones a partir de las mismas.

Control de la calidad. Es el control sistémico que ejecuta la parte técnica del proyecto. Se diseña en función del control de la ejecución del proyecto por objetivos, acciones y tareas. Se reflejan en los criterios de medidas contenidas en los hitos de la programación estructurada del proyecto.

El control de la calidad abarca desde las materias primas hasta la terminación del proyecto y está asociado a los materiales, mano de obra, los equipos, y los métodos empleados en función de las tecnologías. La ISO 8402 brinda un vocabulario actualizado que relaciona todos los conceptos anteriormente señalados.

El control de la calidad es una tarea de todos.

2.4.2 Diseño de la calidad. El diseño de la calidad debe ser integral, conjugando los factores de costo y las necesidades del cliente a través de los procesos que se desarrollan en el proyecto.

El proyecto se desarrolla en distintos procesos y en cada uno es necesario precisar la calidad, como se especifica a continuación.

Etapa de Diseño: Implica generar la garantía de la calidad en todas las fases de la planificación; evaluar las alternativas en el sistema; garantizar también la calidad

²⁶ DELGADO R. / VEREZ M. *La Dirección Integrada de Proyectos haciendo uso de las tecnologías de la informática y las comunicaciones en el marco del perfeccionamiento empresarial.* Monografías.com Internet. 2003

de la documentación del proyecto y la vinculación de las tareas con las normas y procedimientos.

Etapa de ejecución. En esta se debe garantizar la información prevista en los cortes; elaborar los diagnósticos y pronósticos; evaluar la calidad en las tareas en ejecución en el corte; disponer de una columna para la evaluación de la calidad; hacer uso de las notas y vinculaciones en las tareas con rechazo en los cortes; tomar en cuenta la evaluación de la calidad en la toma de decisiones.

La aceleración del cambio y las exigencias del usuario originan variaciones en el diseño de la calidad con variaciones en el costo. Este balance o ajuste periódico es necesario ejecutarlo dentro del entorno del presupuesto con vista a mantener un resultado competitivo.

Etapa de cierre. Debe permitir la evaluación integral de la calidad del proyecto y de los indicadores de calidad; la correspondencia con la calidad prevista en la concepción del proyecto y la evaluación de la satisfacción del cliente.

Diseño del sistema. Flujo de producción. El diseño del sistema de la calidad se ejecuta sobre un flujo de producción o proceso de organización con enfoque sistémico, que permita analizar la secuencia y definir en cada tarea su entrada y salida, precisar la documentación normalizativa y metrológica a emplear y reflejarla en el sistema informático para facilitar su control posterior.

En cada tarea deben estar recogidos los criterios de medidas que garantizan la misma con los cuales deben estar comprometidos todos los que participan en su ejecución.

En el diseño del proyecto aplicando el enfoque sistémico se definen las tareas y su secuencia de ejecución. En este proceso se define para cada tarea i la información de entrada con los parámetros de calidad que deben garantizar la tarea $(i - 1)$.

En la tarea i se desarrolla un proceso que termina con una salida caracterizada por las condiciones de calidad requeridas para la ejecución de la tarea $(i + 1)$. Este proceso se diseña con acuerdos de los criterios de medidas con todas las partes interesadas y recogidas en las tareas y en los hitos del proyecto en dependencia de sus prioridades y la incidencia del sistema. La capacitación del personal juega un papel importante.

En los sistemas de organización en cadena para proyectos lineales el diseño del sistema de calidad se desarrolla por cada cadena simple.

El desarrollo de un sistema de Dirección Integrada de la Calidad debe tener un soporte informático del proyecto que permita desarrollar la DIP. El sistema propuesto es el Project 2002 que permite desarrollar su programación

estructurada y establecer los vínculos con la base normalizativa para detallar los criterios de medidas por tareas e hitos.

El sistema informático debe ser capaz de asimilar los comentarios, notas y ampliaciones en los casos en que no se cumplen los índices de calidad, explicando las causas y las medidas tomadas para restablecer los índices, con vistas a lograr la aceptación por la tarea siguiente y continuar el flujo de producción. En los casos de rechazo el incremento del costo y el tiempo son elementos importantes a destacar.

Estos criterios serán recogidos en la tarea correspondiente y cuando incide en el cumplimiento de los objetivos parciales es reflejado en el hito de control para ser evaluado por la gerencia con vistas a tomar las decisiones necesarias para restablecer la calidad. Los hitos son programados de acuerdo con los intereses de control del proyecto de acuerdo con la programación del cumplimiento de los objetivos parciales previstos en el proyecto. Los hitos se representan en el cronograma de ejecución en las fechas previstas con duración igual a cero.²⁷

Manual de la calidad del proyecto El manual de calidad del proyecto relaciona la documentación normalizativa necesaria y el apoyo metrológico para garantizar la gestión de la calidad del proyecto a partir del manual de la empresa. Se establecen los procedimientos para el control, apoyado por la estructura funcional, delimitando las responsabilidades en el mismo.

La integración entre la DIP y la DIC es una necesidad de la empresa actual dadas las exigencias del cliente que se desarrolla en el entorno moderno.

Para lograr la integración es necesario disponer en la red informática de una base de datos que contenga las normas y procedimientos con el objetivo de lograr establecer las vinculaciones necesarias entre las tareas del proyecto y sus correspondientes normas.

El uso de las notas para la ubicación de los criterios de medidas es un elemento importante para garantizar la información por la cual será evaluado el cumplimiento de las tareas.

Es indispensable conocer antes del inicio de la tarea la forma en que se debe ejecutar y como se medirán sus resultados.

Este es un elemento importante por su incidencia en la evaluación del desempeño del jefe de la tarea.

2.4.3 Control de ejecución de la calidad. El control de ejecución del proyecto parte de definir las necesidades de cada nivel jerárquico, desde el autocontrol en

²⁷ Idem

cada puesto de trabajo en las tareas, conjunto de tareas y objetivos, que serán evaluados en los cortes del proyecto. La calidad debe controlarse de la misma forma establecida en la estrategia de la empresa para ser consecuente en cada nivel de la misma y evaluar el costo, tiempo y calidad para satisfacer integralmente las necesidades de la alta gerencia, los directivos o directores de proyectos y los ejecutivos o jefes de unidad o tareas apoyadas por un sistema de control informático eficiente.

El control que ejecuta la alta gerencia está asociado al control por objetivos de los proyectos en ejecución y se basa fundamentalmente en los criterios de medida contenidos en los hitos de la programación estructurada, partiendo del concepto que no es posible controlarlo todo, por tanto, tiene que existir un nivel de prioridades para ejercer un control eficaz con alta eficiencia. En el resto de los niveles jerárquicos se controlan también los objetivos o los criterios que garantizan los mismos hasta llegar al control del puesto de trabajo por el nivel correspondiente apoyándose en la Estructura de Desagregación del Proyecto.

El control periódico en función de los hitos y su sistema de información apoyado por el correo electrónico, permite contar con una información actualizada a nivel de red y servidor que facilita el uso de las páginas Web para garantizar la búsqueda de la información necesaria para la toma de decisiones.

En el control se emplean los ensayos donde se miden los parámetros con las especificaciones, índices y tolerancias recogidas en las normas técnicas y las regulaciones que se analizan en los controles, inspecciones y auditorías de calidad.

En ocasiones es necesario ilustrar los problemas de calidad con un croquis, gráfico de ejecución, una foto digital incorporada a las notas o en un archivo vinculado a la tarea.

La documentación del control de la calidad sirve de base para la etapa de ejecución y debe estar prevista en la documentación del proyecto en los cortes de acuerdo con lo registrado en el manual de calidad.

En proyectos de altos presupuestos y complejidades técnicas, se aplica el concepto de certificación de la calidad por módulos en los que intervienen un grupo de expertos que evalúan los índices y certifican la misma.

La calificación de la calidad en cada tarea y por hito se recoge en el sistema de información y se incorpora a la EDP del modelo informático. La actualización de la calidad en el seguimiento del proyecto por cortes se ejecuta por los responsables de las tareas resumen directamente en el Project y el director del proyecto evalúa el proyecto en el corte y toma las decisiones estratégicas.

La DIP brinda un apoyo importante para la definición del nivel de calidad del resultado del proyecto, conjugando los criterios de la visión con las condiciones objetivas en las que se desenvuelve el proyecto y los requerimientos del cliente, lo que permite definir el nivel de calidad exigido para precisar los materiales, equipos, personal y métodos de trabajo que garanticen los objetivos del proyecto.

El director del proyecto requiere de una evaluación en función de un corte en el proyecto donde se conjuguen el costo el tiempo y la calidad. Tomando como base esta información se desarrolla un pronóstico con alternativas para preceder a tomar las decisiones correspondientes.

La vinculación de la evaluación de la calidad en los cortes con el tablero de comandos a través de las bases de datos de los proyectos en ejecución brinda la información estratégica necesaria para la realización del pronóstico en el próximo corte con las metas estratégicas a lograr en función del éxito de los proyectos. Este detalle puede ser recogido en un croquis o gráfico en PowerPoint el cual puede ser incorporado en las notas de la tarea.

El criterio es brindar información adicional en las notas o vinculaciones complementarias en Word. En los casos de rechazo de calidad con incremento del costo y el tiempo o en este caso en el que no hay rechazo pero la evaluación es de regular por estar muy próximo al mismo, es un alerta para el control.

La ISO 9004 Gestión de la calidad, brinda un conjunto de elementos relacionados con la dirección que facilitan la aplicación de estos conceptos.

El volumen de información y la calidad deben ser consecuente con la capacidad de procesamiento con que se cuente, de acuerdo con la pirámide de dirección establecida. Si se pide mas información que la capacidad de procesamiento, se genera un trabajo extra innecesario que puede ser muy incomodo y por tanto el que recibe la información debe estar convencido de la importancia y calidad de la misma para su posterior uso.

Para obtener la información del comportamiento de la calidad en el soporte informático, el sistema se basa en la incorporación de una columna en la tabla de control con los índices de calidad de cada tarea, que son los encargados de dar una evaluación a este nivel con una columna de Calificación de calidad y otras especificaciones del proyecto.

Cada uno de estos índices de calidad representa en cada tarea una de sus características fundamentales, a las cuales por su importancia es necesario valorar. En tareas donde sea necesario valorar mas de un índice para brindar una calificación final es conveniente la ponderación de los mismos para obtener el promedio ponderado. Esta información se brinda por cortes según la línea de progreso y la tabla de seguimiento.

En ocasiones se pide indiscriminadamente máxima calidad sin tener criterio de la significación que representa en el costo. Esta valoración es importante tanto para el proyecto general como para cada una de sus partes. El éxito de un diseño balanceado está en definir la calidad requerida en cada una de sus partes con el objetivo de garantizar la calidad necesaria en el resultado final del proyecto.

El concepto del desarrollo del proyecto, tal y como lo concibe la DIP evalúa el costo integralmente considerando tanto el costo de diseño y ejecución como el de mantenimiento en la etapa de explotación bajo el concepto proyecto – negocio.

Normalmente cuando no se concibe de esta forma, el costo de construcción trata de minimizarse sin tener presente el costo de mantenimiento que en estos casos por lo general es alto.

Este razonamiento induce a que es necesario buscar un índice de calidad del proyecto que integre los costos, buscando una función objetivo de costo total mínimo.

2.4.4 El proceso de capacitación de los especialistas. La integración de la DIP con la DIC en un soporte informático único requiere de una definición precisa acorde con las características de la empresa con el concepto de traje a la medida. Definida la integración de los sistemas es necesario determinar para cada dependencia de la estructura funcional las necesidades de capacitación atendiendo a sus objetivos parciales en la estructura de desagregación del proyecto y la base normalizadora necesaria para el éxito de su trabajo. Con estas definiciones se determina la estrategia de capacitación para cada dependencia funcional de la empresa. El uso de los sistemas personalizados de capacitación montados en sistemas informáticos que brindan la posibilidad de acceso a la información en forma discreta atendiendo a las posibilidades de capacitación del especialista desde su puesto de trabajo permiten la capacitación personalizada con mediciones del impacto en función de la secuencia de los proyectos que ejecuta en el tiempo.

Se cuenta con un CD en formato electrónico con el objetivo de facilitar el estudio a distancia o a través de su montaje en la red informática de la empresa para su uso discreto personalizado en función de la disponibilidad de tiempo de los especialistas para su capacitación.

La DIC toma en cuenta y garantiza la gestión ambiental. En el libro de DIP haciendo uso de las NTIC se hace énfasis en este tema.

Por lo tanto, para concluir estos aspectos, la Dirección Estratégica Integrada de la empresa constituye la base del proceso de perfeccionamiento y requiere de la integración de la Dirección Integrada de Proyectos y la Dirección de la Calidad con el objetivo de lograr el éxito de los proyectos.

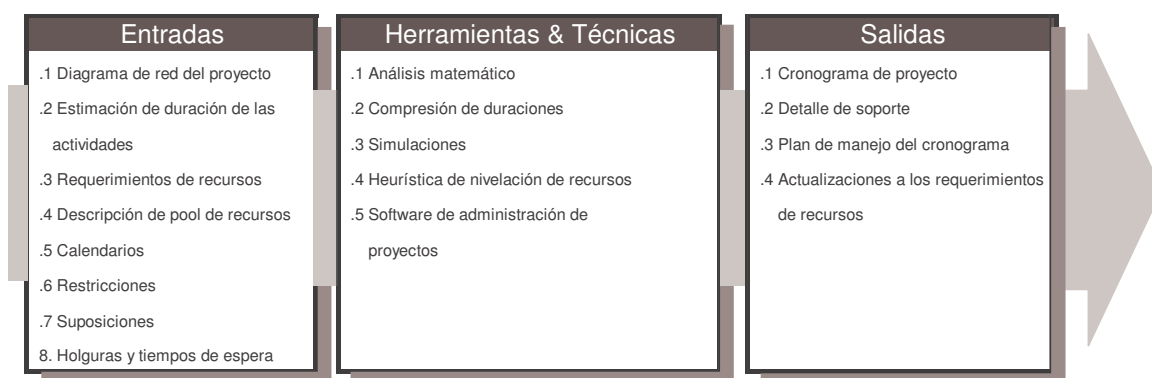
El uso del MS Project 2002 permite la integración de la DIP y la DIC, en un sistema informático único con la evaluación de los criterios de medida asociados a las tolerancias de las magnitudes por cortes, para la toma de decisiones en apoyo de la gerencia empresarial haciendo uso del tablero de comandos.

Para la implantación de un sistema de DIC es imprescindible el análisis conceptual de las normas, procedimientos y la metrología como un sistema que parte de la base normalizativa donde se precisa la forma en que deben desarrollarse las tareas. La metrología determina la forma en que se mide la ejecución de las tareas haciendo uso de los criterios de medida apoyados por los laboratorios y los medios de medición. La gestión de la calidad integra todo el sistema en función de los requerimientos del cliente en el marco del presupuesto previsto, apoyado por las Nuevas Tecnologías de la Informática y las Comunicaciones.

2.5 DESARROLLO DE LA PROGRAMACIÓN

2.5.1 Elementos claves. El desarrollo de la programación requiere determinar fechas de comienzo y finalización para las actividades del proyecto. Si las fechas de comienzo y finalización no son realistas, el proyecto tendrá pocas probabilidades de terminar como programar. El proceso de desarrollo de la programación, muchas veces tendrá que ser iterarte (al mismo tiempo con los procesos que proveen entradas, especialmente la estimación de las duraciones y de costos) antes de la determinación de la programación del proyecto.

El desarrollo el general se esquematiza a continuación



Fuente. Esta investigación

Al respecto se deben tener en cuenta algunos elementos importantes, a saber:

Descripción del pool de recursos. El conocimiento de que recursos estarán disponibles en que tiempos y en que patrones es necesario para el desarrollo de la

programación. Por ejemplo, los recursos compartidos podrán ser especialmente difíciles de programar ya que su disponibilidad puede ser altamente variable.

El grado de detalle y el nivel de especificidad en la descripción del pool de recursos pueden variar. Por ejemplo, para el desarrollo preliminar de la programación de un proyecto de consultoría, la entidad solo necesita saber que dos consultores estarán disponibles en un marco de tiempo específico. La programación final del mismo proyecto sin embargo, debe identificar que consultores específicos estarán disponibles.

Calendarios. Los calendarios de proyecto y de recursos identifican períodos de tiempo donde es permitido trabajar. Los calendarios de proyecto afectan a todos los recursos (e.g., algunos proyectos solo trabajaran durante horas normales de negocio, mientras que otros trabajaran tres turnos diariamente). Los calendarios de recursos afectan a un recurso o categoría de recurso en particular (e.g., un miembro del equipo de proyecto puede estar de vacaciones o en un curso de capacitación, un contrato colectivo de trabajo puede limitar la labor de algunos empleados durante la semana).

Fechas impuestas. La entrega de ciertos productos en una fecha específica puede ser requerida por los usuarios del proyecto, el cliente del proyecto, u otros factores externos.

Eventos claves o hitos de importancia. La entrega de ciertos productos en una fecha específica puede ser solicitada por un patrocinador del proyecto, el cliente de proyecto, u otros partidos interesados. Una vez programados, estas fechas se vuelven formales, y muchas veces sólo se pueden cambiar con gran dificultad.

Holguras y tiempos de espera. Cualquiera de las dependencias puede requerir de una holgura o tiempo de espera (lags y leads) para poder definir de manera correcta la relación (e.g., puede existir un retraso de dos semanas entre la compra de un equipo y su instalación para su uso).

2.5.2 Herramientas y técnicas para el desarrollo de la programación. Se refieren en general al análisis matemático y los diversos métodos que se utilizan para la programación.

Análisis matemático. El análisis matemático requiere calcular las fechas teóricas tempranas y tardías para todas las actividades sin tener en cuenta cualquier limitación del pool de recursos disponibles. Las fechas resultantes no son la programación, sino que mas bien indican los periodos de tiempo el los que las actividades se deberían programar dadas las limitaciones de recursos y de otros tipos conocidas.

Las técnicas más comunes conocidas se relacionan a continuación.

Método de la ruta crítica (CPM). Calcula un solo juego determinístico de fechas tempranas y tardías de comienzo y finalización para cada actividad, basada en una lógica de red secuencial y solo una duración. El foco de CPM es calcular la flotación para poder determinar que actividades tienen la menor flexibilidad de programación. Los algoritmos inherentes a CPM son muchas veces usados en otros tipos de análisis matemáticos.

Método de evaluación y revisión gráfica (GERT). Permite el tratamiento probabilístico de tanto la red de lógica como de la estimación de las duraciones de las actividades (i.e., algunas actividades pueden no ser ejecutadas, algunas pueden ser ejecutadas algunas veces, y otras pueden ser ejecutadas varias veces).

Técnica de evaluación y revisión de programas (PERT). Usa lógica secuencial de red y una distribución por pesos para la duración de las actividades para calcular la duración del proyecto. Aunque existen algunas diferencias superficiales, PERT se diferencia de CPM en que PERT usa la media de la distribución (el valor esperado) en lugar del el valor más probable usado originalmente en CPM. PERT se usa poco hoy día aunque muchas veces se usan estimados que se asemejan a PERT en cálculos de CPM.

Compresión de duraciones. La compresión de duraciones es un caso especial de análisis matemático que busca maneras de acortar la duración del proyecto sin cambiar el alcance de este (por ejemplo: cumplir fechas impuestas o metas de programación).

2.5.3 Salidas del desarrollo de la programación. Comprende lo relativo a la programación, cronograma, diagramas, redes y otros elementos.

Programación del proyecto. Incluye al menos fechas de inicio y de terminación planeadas para cada detalle de actividad.

El cronograma de proyecto. Puede ser presentado de forma resumida (la "programación maestra") o en forma detallada. Aunque puede ser presentado en forma tabular, suele presentarse generalmente de forma gráfica usando uno o más de los formatos como los siguientes:

Diagramas de red de proyecto. Estas gráficas muestran usualmente tanto la lógica del proyecto como las actividades de su ruta crítica.

Gráficas de barras. También se conocen como diagramas de Gant, muestran tanto las fechas de comienzo como de terminación de las actividades y sus duraciones esperadas, pero no muestran sus dependencias. Son fáciles de leer, y son de uso frecuente en presentaciones ejecutivas.

Gráficas de hitos o mojones. Son similares a las gráficas de barras, pero identifican los comienzos o terminaciones programadas de las principales entregas e interfaces externas claves del proyecto.

Diagramas de red de proyectos en escalas de tiempo. Son una mezcla de los diagramas de red del proyecto y de los diagramas de barras de una manera tal que muestran la lógica del proyecto, las duraciones de las actividades, y la información de la programación.

Detalle de soporte. El detalle de soporte para la programación del proyecto incluye al menos documentación de todas las restricciones y suposiciones identificadas. El grado de detalle adicional requerido varía de acuerdo al área de aplicación.

Plan de manejo de la programación. Un plan de manejo de la programación define como se manejarán los cambios a la programación. Puede ser formal o informal, con gran grado de detalle o basado de forma conceptual amplia dependiendo de las necesidades del proyecto. Es un elemento subsidiario del plan general del proyecto.

Actualizaciones a los requerimientos de recursos. Las nivelaciones de recursos y actualizaciones a la lista de actividades pueden tener un efecto significativo sobre las estimaciones preliminares de los requerimientos de recursos.

Control de la programación. El control de la programación se preocupa con (a) influenciar los factores que crean cambios en la programación para asegurar que tales cambios sean beneficiosos, (b) determinar que la programación ha sido cambiada, y (c) administrar los cambios actuales cuando y como ocurren. El control de la programación debe estar íntimamente ligada con los otros procesos de control, tal como se describe en la Sección 4.3, Control de Cambios General.

Software de administración de proyectos. El software de administración de proyectos se refiere a la programación descrita. La habilidad del software de administración de proyectos de hacer un seguimiento de fechas programadas versus fechas reales y de pronosticar los efectos de los cambios de programación, reales o potenciales, hacen de esta herramienta un recurso útil para el control de la programación.

3. PROPUESTA DE MODELOS DE DESARROLLO DE SOFTWARE

3.1 ELEMENTOS DE GERENCIA Y ADMINISTRACIÓN DEL PROYECTO DE SOFTWARE

3.1.1 El sistema de información. El sistema de información es: “Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) desempeñar su actividad de acuerdo a su estrategia de negocio”.

Otra definición de sistemas de información se refiere a “procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas”.

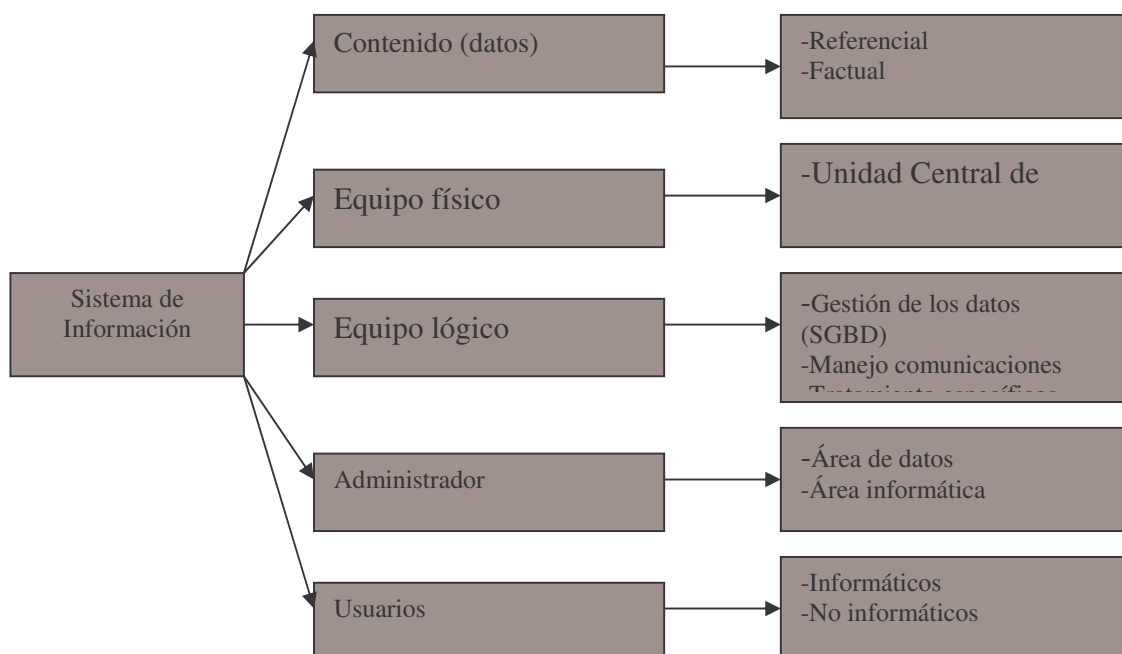
Por lo tanto podemos definir un sistema de información como un conjunto de subsistemas que incluyen hardware, software, medios de almacenamiento de datos ya sea primarios, secundarios y bases de datos relacionadas entre si con el fin de procesar entradas para realizar transformaciones a esas entradas y convertirlas en salidas de información importantes en la toma de decisiones.

El objetivo de un sistema de información es ayudar al desempeño de las actividades que desarrolla la empresa, suministrando la información adecuada, con la calidad requerida, a la persona o departamento que lo solicita, en el momento y lugar especificados con el formato más útil para el receptor.

El sistema de información esta al servicio de los objetivos de la empresa para lograr dichos objetivos la empresa y sus individuos adoptan procedimientos y practicas de trabajo que resultan mas útiles y eficaces.

Todo sistema de información computacional con los componentes que se esquematizan a continuación.

Figura 10.



Fuente. Esta investigación

Los elementos de un sistema de información hacen referencia a los procedimientos y las prácticas habituales del trabajo que los directivos suelen hacer para coordinar los distintos elementos de la empresa para su buen funcionamiento. En ellos se destacan los siguientes:

- **Información.** Este es el elemento fundamental de todo sistema y su razón de ser. Este debe adaptarse a las personas que la manejan y al equipo disponible con el que cuenta la empresa, según los procedimientos de trabajo para que las actividades se realicen de forma eficaz.
- **Personas o usuarios.** Se trata de los individuos o unidades de la organización que introducen manejan o usan la información para realizar sus actividades y operaciones en función de los procedimientos de trabajo establecidos.
- **Equipo de soporte.** El equipo de soporte se ocupa para la comunicación, el procesamiento y el almacenamiento de información, este constituye la parte más visible del sistema de información, su parte tangible o física. Este sistema tangible y físico puede incluir elementos de los mas variados

niveles tecnológicos y pueden ser: papel, maquinas de escribir, archivadores, cintas magnéticas, impresoras, computadoras, etc.

3.1.2 Planificación del proyecto de software. El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

El Objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables.

Las actividades asociadas al proceso se aconseja sean las siguientes:

A. Ámbito del Software. Es la primera actividad de llevada a cabo durante la planificación del proyecto de Software.

En esta etapa se deben evaluar la función y el rendimiento que se asignaron al Software durante la Ingeniería del Sistema de Computación para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible para directivos y técnicos. Describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad, se evalúan las funciones del ámbito y en algunos casos se refinan para dar mas detalles antes del comienzo de la estimación. Las restricciones de rendimiento abarcan los requisitos de tiempo de respuesta y procesamiento, identifican los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.

El requisito esencial es la obtención de la información necesaria para el software. Para esto el analista y el cliente se reúnen sobre las expectativas del proyecto y se ponen de acuerdo en los puntos de interés para su desarrollo.

B. Recursos. La Segunda tarea de la planificación del desarrollo de Software es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de Software. Esto simula a una pirámide donde las Herramientas (hardware y Software), son la base proporciona la infraestructura de soporte al esfuerzo de desarrollo, en segundo nivel de la pirámide se encuentran los Componentes reutilizables. Y en la parte mas alta de la pirámide se encuentra el recurso primario, las personas (el recurso humano).

Cada recurso queda especificado mediante cuatro características: Descripción del Recurso, Informes de disponibilidad, Fecha cronológica en la que se requiere el recurso, Tiempo durante el que será aplicado el recurso.

3.1.3 Estimación del proyecto. En el principio el costo del Software constituía un pequeño porcentaje del costo total de los sistemas basados en Computadoras. Hoy en día el Software es el elemento más caro de la mayoría de los sistemas informáticos.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.

Para realizar estimaciones seguras de costos y esfuerzos tienen varias opciones posibles:

- Dejar la estimación para mas adelante (obviamente podemos realizar una estimación al cien por cien fiable después de haber terminado el proyecto.
- Basar las estimaciones en proyectos similares ya terminados.
- Utilizar técnicas de descomposición relativamente sencillas para generar las estimaciones de costos y esfuerzo del proyecto.
- Desarrollar un modelo empírico para él cálculo de costos y esfuerzos del Software.

Desdichadamente la primera opción, aunque atractiva no es práctica. La Segunda opción puede funcionar razonablemente bien si el proyecto actual es bastante similar a los esfuerzos pasados y si otras influencias del proyecto son similares. Las opciones restantes son métodos viables para la estimación del proyecto de software. Desde el punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas usando cada una de ellas como comprobación de las otras.

Antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su tamaño.

3.1.4 Diferentes modelos de estimación. Existen diferentes modelos y herramientas de estimación como son los que a continuación se especifican.

Los Modelos Empíricos. En ellos los datos que soportan la mayoría de los modelos de estimación obtienen una muestra limitada de proyectos. Por esta razón, el modelo de estimación no es adecuado para todas las clases de software y en todos los entornos de desarrollo. Por lo tanto los resultados obtenidos de dichos modelos se deben utilizar con prudencia.

El Modelo COCOMO. Barry Boehm, en su libro clásico sobre economía de la Ingeniería del Software, introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO, por su nombre en Ingles (Constructive, Cost, Model) modelo constructivo de costos. La jerarquía de modelos de Boehm esta constituida por los siguientes:

- **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.*

Herramientas Automáticas De Estimación. Estas herramientas permiten al planificador estimar costos y esfuerzos, así como llevar a cabo análisis del tipo, que pasa si, con importantes variables del proyecto, tales como la fecha de entrega o la selección del personal. Aunque existen muchas herramientas automáticas de estimación, todas exhiben las mismas características generales y todas requieren de una o más clases de datos.

A partir de estos datos, el modelo implementado por la herramienta automática de estimación proporciona estimaciones del esfuerzo requerido para llevar a cabo el proyecto, los costos, la carga de personal, la duración, y en algunos casos la planificación temporal de desarrollo y riesgos asociados.

En resumen el planificador del Proyecto de Software tiene que estimar tres cosas antes de que comience el proyecto: cuanto durará, cuanto esfuerzo requerirá y cuanta gente estará implicada. Además el planificador debe predecir los recursos de hardware y software que va a requerir y el riesgo implicado. Para obtener estimaciones exactas para un proyecto, generalmente se utilizan al menos dos de las tres técnicas referidas anteriormente. Mediante la comparación y la conciliación de las estimaciones obtenidas con las diferentes técnicas, el planificador puede obtener una estimación más exacta. La estimación del proyecto de software nunca será una ciencia exacta, pero la combinación de buenos datos históricos y técnicas puede mejorar la precisión de la estimación.

3.2 ANÁLISIS DE LOS SISTEMAS DE COMPUTACIÓN

3.2.1 Premisas. El análisis de los sistemas de computación constituye un conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad. Un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. Un método, plan o procedimiento de clasificación para hacer algo. También es un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información.

Lo anterior se lleva a cabo teniendo en cuenta al menos los siguientes principios:

- *Debe presentarse y entenderse el dominio de la información de un problema.*
- *Definir las funciones que debe realizar el Software.*
- *Representar el comportamiento del software a consecuencias de acontecimientos externos.*
- *Dividir en forma jerárquica los modelos que representan la información, funciones y comportamiento.*

El proceso debe partir desde la información esencial hasta el detalle de la implementación.

Un Análisis de Sistema se lleva a cabo teniendo en cuenta los siguientes objetivos en mente:

- *Identificar las necesidades del Cliente.*
- *Evaluar que conceptos tiene el cliente del sistema para establecer su viabilidad.*
- *Realizar un Análisis Técnico y económico.*
- *Asignar funciones al Hardware, Software, personal, base de datos, y otros elementos del Sistema.*
- *Establecer las restricciones de presupuestos y planificación temporal.*
- *Crear una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.*

3.2.2 Objetivos del análisis. Deben referirse al menos a los siguientes aspectos:

a. Identificación de Necesidades. Es el primer paso del análisis del sistema, en este proceso en Analista se reúne con el cliente y/o usuario (un representante institucional, departamental o cliente particular), e identifican las metas globales, se analizan las perspectivas del cliente, sus necesidades y requerimientos, sobre la planificación temporal y presupuestal, líneas de mercadeo y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

Esto se puede llamar también como " Análisis de Requisitos " y se lo puede dividir en cinco partes: Reconocimiento del problema, Evaluación y Síntesis, Modelado, Especificación y Revisión.

b. Estudio de Viabilidad. Muchas veces cuando se emprende el desarrollo de un proyecto de Sistemas los recursos y el tiempo no son realistas para su materialización sin tener pérdidas económicas y frustración profesional. La viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta cuatro áreas principales de interés:

Viabilidad económica: Una evaluación de los costos de desarrollo, comparados con los ingresos netos o beneficios obtenidos del producto o Sistema desarrollado.

Viabilidad Técnica: Un estudio de funciones, rendimiento y restricciones que puedan afectar la realización de un sistema aceptable.

Viabilidad Legal: Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el Sistema.

Alternativas. Una evaluación de los enfoques alternativos del desarrollo del producto o Sistema.

El estudio de la viabilidad puede documentarse como un informe aparte para la alta gerencia.

Análisis Económico y Técnico. El análisis económico incluye lo que llamamos, el análisis de costos – beneficios, y significa una valoración de la inversión económica comparado con los beneficios que se obtendrán en la comercialización y utilidad del producto o sistema.

Muchas veces en el desarrollo de Sistemas de Computación estos son intangibles y resulta un poco dificultoso evaluarlo, esto varía de acuerdo a las características del Sistema. El análisis de costos – beneficios es una fase muy importante de ella depende la posibilidad de desarrollo del Proyecto.

En el Análisis Técnico, el Analista evalúa los principios técnicos del Sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad.

Los resultados obtenidos del análisis técnico son la base para determinar sobre si continuar o abandonar el proyecto, si hay riesgos de que no funcione, no tenga el rendimiento deseado, o si las piezas no encajan perfectamente unas con otras.

Modelado de la arquitectura del Sistema. Cuando se quiere dar a entender mejor lo que vamos a construir en el caso de proyectos, se crea un modelo idéntico, pero en menor escala (mas pequeño). Sin embargo cuando aquello que construiremos es un Software, sobre todo para Universidades, nuestro modelo debe tomar una forma diferente, deben representar todas las funciones y sub-funciones de un Sistema. Los modelos se concentran en lo que debe hacer el sistema no en como lo hace, estos modelos pueden incluir notación gráfica, información y comportamiento del Sistema.

Todos los Sistemas basados en computadoras pueden modelarse como transformación de la información empleando una arquitectura del tipo entrada y salida.

Especificaciones del Sistema. Es un Documento que sirve como fundamento para la Ingeniería Hardware, software, Base de datos, e ingeniería Humana. Describe la función y rendimiento de un Sistema basado en computadoras y las dificultades que estarán presentes durante su desarrollo. Las Especificaciones de los requisitos del software se producen en la terminación de la tarea del análisis.

3.3 DISEÑO DE LOS SISTEMAS DE COMPUTACIÓN

3.3.1 Conceptos y principios. El Diseño de Sistemas se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

La etapa del Diseño del Sistema encierra cuatro etapas:

Etapa 1. El diseño de los datos. Transforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el Software.

Etapa 2. El Diseño Arquitectónico. Define la relación entre cada uno de los elementos estructurales del programa.

Etapa 3. El Diseño de la Interfaz. Describe como se comunica el Software

consigo mismo, con los sistemas que operan junto con el y con los operadores y usuarios que lo emplean.

Etapa 4. El Diseño de procedimientos. Transforma elementos estructurales de la arquitectura del programa. La importancia del Diseño del Software se puede definir en una sola palabra **Calidad**, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El Diseño del Software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación. Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el Software.

3.3.2 Diseño de la salida. En este caso salida se refiere a los resultados e informaciones generadas por el Sistema, Para la mayoría de los usuarios la salida es la única razón para el desarrollo de un Sistema y la base de evaluación de su utilidad. Sin embargo cuando se realiza un sistema, como analistas deben realizar lo siguiente:

- *Determinar que información presentar. Decidir si la información será presentada en forma visual, verbal o impresora y seleccionar el medio de salida.*
- *Disponer la presentación de la información en un formato aceptable.*
- *Decidir como distribuir la salida entre los posibles destinatarios.*

3.3.3 Diseño de archivos. Incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera a emplear para guardar detalles de las transacciones, datos históricos, o información de referencia. Entre las decisiones que se toman durante el diseño de archivos, se encuentran las siguientes:

- *Los datos que deben incluirse en el formato de registros contenidos en el archivo.*

- *La longitud de cada registro, con base en las características de los datos que contenga.*
- *La secuencia a disposición de los registros dentro del archivo (La estructura de almacenamiento que puede ser secuencial, indexada o relativa).*

La mayoría de los sistemas de información ya sean implantado en sistemas de cómputos grandes o pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas utilizan u administrador de base de datos, en este caso el diseñador no construye la base de datos sino que consulta a su administrador para ponerse de acuerdo en el uso de esta en el sistema.

3.3.4 Implantación, evaluación y pruebas. Es la última fase del desarrollo de Sistemas. Es el proceso instalar equipos o Software nuevo, como resultado de un análisis y diseño previo como resultado de la sustitución o mejoramiento de la forma de llevar a cabo un proceso automatizado.

Al Implantar un Sistema de Información lo primero que se debe hacer es asegurarse que el Sistema sea operacional o sea que funcione de acuerdo a los requerimientos del análisis y permitir que los usuarios puedan operarlo.

Existen varios enfoques de Implementación. Se recomienda tener en cuenta los siguientes: darle responsabilidad a los grupos, usar de diferentes estrategias para el entrenamiento de los usuarios, ponderar la situación y proponer un plan de conversión que sea adecuado para la organización, formular medidas de desempeño con las cuales evaluar a los Usuarios, Convertir físicamente el sistema de información antiguo, al nuevo modificado.

En todo caso para la preparación de la Implantación, aunque el Sistema este bien diseñado y desarrollado correctamente, su éxito dependerá de su implantación y ejecución por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

La evaluación se lleva a cabo para identificar puntos débiles y fuertes del Sistema implantado. La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones: operacional, de impacto organizacional, de desempeño del desarrollo y de prueba de sistemas.

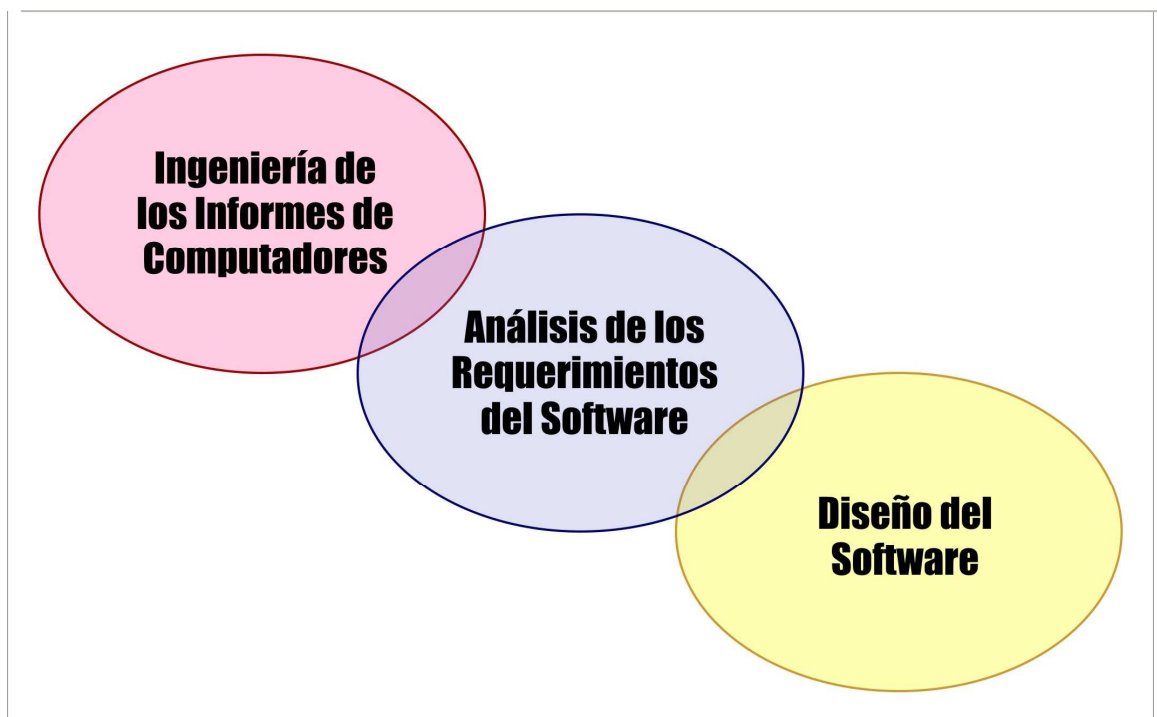
3.4 ANÁLISIS DE REQUERIMIENTOS

3.4.1 Fundamentos. Se debe tener en cuenta que el análisis de requerimientos es el conjunto de técnicas y procedimientos que nos permiten conocer los

elementos necesarios para definir un proyecto de software. Es la etapa más crucial del desarrollo de un proyecto de software.

En la figura 11 se esquematiza el proceso, que más adelante se explica.

Figura 11. Esquema del análisis de requerimientos



Fuente. Esta investigación

En general esas técnicas y procedimientos se definen como funcionales y no funcionales.

Funcionales: Condición o capacidad de un sistema requerida por el usuario para resolver un problema o alcanzar un objetivo.

No Funcionales: Condición o capacidad que debe poseer un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.

Para realizar bien el desarrollo de software es esencial realizar una especificación completa de los requerimientos de los mismos. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento, El ámbito del programa, establecido inicialmente durante la ingeniería del sistema, es refinado en detalle. Se analizan y asignan a los distintos elementos de los programas las soluciones alternativas.

Tanto el que desarrolla el software como el cliente tienen un papel activo en la especificación de requerimientos. El cliente intenta reformular su concepto, algo nebuloso, de la función y comportamiento de los programas en detalles concretos, El que desarrolla el software actúa como interrogador, consultor y el que resuelve los problemas.

El análisis y especificación de requerimientos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Puesto que el contenido de comunicación es muy alto, abundan los cambios por mala interpretación o falta de información. El dilema con el que se enfrenta un ingeniero de software puede ser comprendido repitiendo la sentencia de un cliente anónimo: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que lo que creíste oír sea lo que yo quise decir".

3.4.2 Particularidades. El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistema y el diseño de programas. Este análisis facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interfaz con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa.

Particularmente el análisis de requerimientos permite al ingeniero refinar la asignación de software y representar el dominio de la información que será tratada por el programa. Dota además al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedimental.

Finalmente, la especificación de requerimientos suministra al técnico y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.

3.4.3 Tareas del análisis. El análisis de requerimientos puede dividirse en cuatro áreas:

Tarea 1: Reconocimiento del problema

Tarea 2: Evaluación y síntesis

Tarea 3: Especificación

Tarea 4: Revisión.

Inicialmente, el analista estudia la especificación del sistema (si existe) y el plan de proyecto. Es importante comprender el contexto del sistema y revisar el ámbito de los programas que se usaron para generar las estimaciones de la planificación. A continuación, debe establecerse la comunicación necesaria para el análisis, de forma que se asegure el reconocimiento del problema.

El analista debe establecer contacto con el equipo técnico y de gestión del usuario/cliente y con la empresa o institución que vaya a desarrollar el software. El gestor del programa puede servir como coordinador para facilitar el establecimiento de los caminos de comunicación. El objetivo del analista es reconocer los elementos básicos del programa tal como lo percibe el usuario/cliente.

La evaluación del problema y la síntesis de la solución es la siguiente área principal de trabajo del análisis. El analista debe evaluar el flujo y estructura de la información, refinar en detalle todas las funciones del programa, establecer las características de la interfase del sistema y descubrir las ligaduras del diseño, Cada una de las tareas sirven para descubrir el problema de forma que pueda sintetizarse un enfoque o solución global.

Las tareas asociadas con el análisis y especificación existen para dar una representación del programa que pueda ser revisada y aprobada por el cliente (en nuestro caso la institución). En un mundo ideal el cliente desarrolla una especificación de requerimientos del software completamente por sí mismo. Esto se presenta raramente en el mundo real. En el mejor de los casos, la especificación se desarrolla conjuntamente entre el cliente y el técnico.

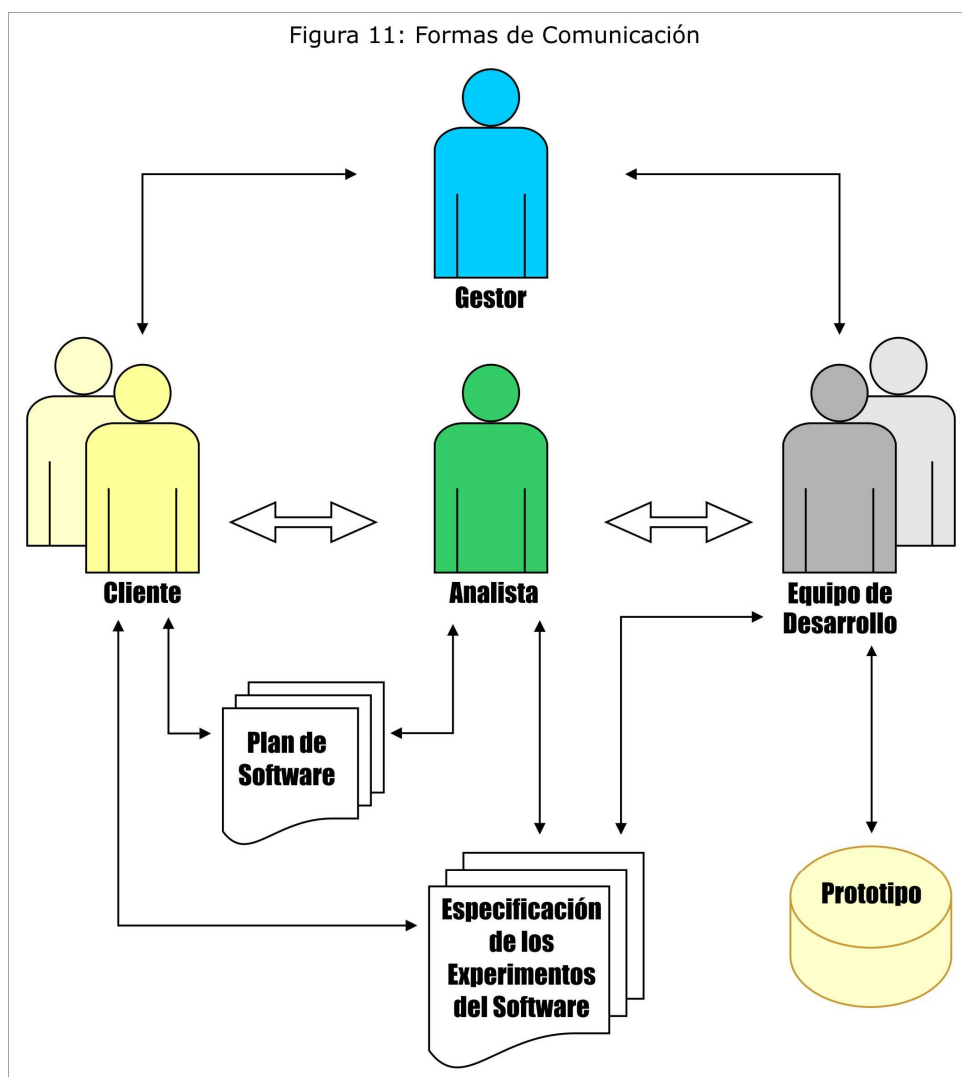
Una vez que se hayan descrito las funcionalidades básicas, comportamiento, interfase e información, se especifican los criterios de validación para demostrar una comprensión de una correcta implementación de los programas. Estos criterios sirven como base para hacer una prueba durante el desarrollo de los programas. Para definir las características y atributos del software se escribe una especificación de requerimientos formal. Además, para los casos en los que se desarrolle un prototipo se realiza un manual de usuario preliminar.

Puede parecer innecesario realizar un manual de usuario en una etapa tan temprana del proceso de desarrollo, Pero de hecho, este borrador del manual de usuario fuerza al analista a tomar el punto de vista del usuario del software. El manual permite al usuario / cliente revisar el software desde una perspectiva de ingeniería humana y frecuentemente produce el comentario: "La idea es correcta pero esta no es la forma en que pensé que se podría hacer esto". Es mejor descubrir tales comentarios lo más tempranamente posible en el proceso.

Los documentos del análisis de requerimiento (especificación y manual de usuario) sirven como base para una revisión conducida por el cliente y el técnico.

La revisión de los requerimientos casi siempre produce modificaciones en la función, comportamiento, representación de la información, ligaduras o criterios de validación. Además, se realiza una nueva apreciación del plan del proyecto de software para determinar si las primeras estimaciones siguen siendo validas después del conocimiento adicional obtenido durante el análisis.

Finalmente, se debe recalcar que uno de los componentes fundamentales de la gerencia del proyecto de software se basa en la comunicación, proceso que se esquematiza con los elementos básicos en la figura 11.



4. PROPUESTA DE UN MODELO DE PROYECTO DE SOFTWARE

4.1 PRESENTACIÓN

Tomando en cuenta los desarrollos gerenciales de proyectos de software de los capítulos anteriores, se puede proponer un modelo especialmente dirigido a las necesidades de Instituciones de Educación Superior, que puede a la vez ser adaptado para otro tipo de organizaciones.

El eje de la metodología que se propone consiste en fortalecer las actividades de los proyectos de desarrollo de software con procesos, documentos, y soportes que apoyen el desarrollo, la rentabilidad y el buen desarrollo del proyecto.

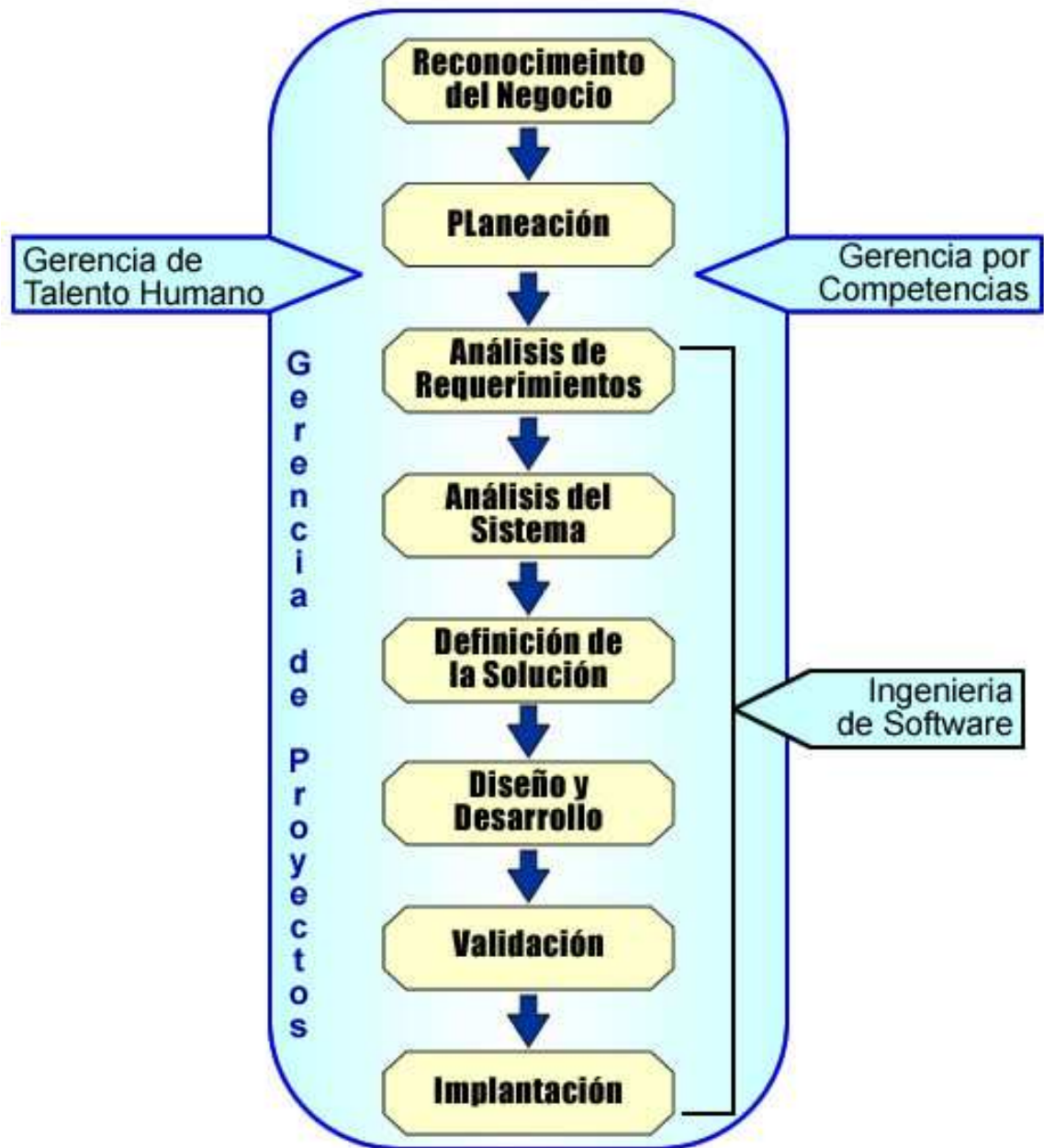
En este contexto el modelo gerencial para la administración de proyectos de software en instituciones educativas se presenta a manera de pasos secuenciales, lo cual permitirá separar las acciones y administrar las tareas y tiempos, realizar seguimientos, conocer el estado actual y mantener retroalimentación en el proyecto. En pocas palabras el modelo de administración de proyectos de software es un mapa que guiará al director de un proyecto a través del desarrollo del mismo.

Sobre estas bases, finalmente, se presenta también, como aplicación de los más recurrentes desarrollos de software en las instituciones de educación superior, la propuesta de un modelo general de software educativo.

4.2 EL MODELO GERENCIAL BÁSICO

La propuesta del modelo debe desarrollarse en etapas que corresponden a la síntesis de lo expuesto en los capítulos anteriores. En forma esquemática se presentan en la figura 12 tales etapas que, con explicaciones que se sintetizan más adelante.

Figura 12. Esquema de la propuesta básica:



Fuente. Esta investigación

PRIMERA ETAPA. ANÁLISIS PREVIO

Generalmente los proyectos de desarrollo de software siempre comienzan con el análisis de los requerimientos del sistema para el que se va a crear el software, pero desde un punto de vista administrativo esto es un error. Un buen gerente comenzaría por conocer a su cliente, su tipo de empresa, a que se dedica y cuales son sus proyecciones en su mercado. Es por esto que el primer paso del modelo propuesto debe ser el reconocimiento del negocio en el cual se desempeña el cliente de la Institución.

Reconocimiento del negocio: Esta etapa es enfocada directamente en el reconocimiento de quien es nuestro cliente. Un buen director de proyecto, que es en si un gerente, debe conocer muy bien con quien va a trabajar y a que se dedica, en otras palabras pregunte al cliente a que se dedica, que necesita y haga todo lo posible por hacérselo realidad.

Al hacer referencia a este punto se debe tener en cuenta que se denomina cliente a una necesidad, a alguien que requiere del proyecto de software. En el caso de las Instituciones de Educación Superior, bien puede ser un proyecto educativo propiamente dicho, la necesidad de gestión en cualquiera de sus formas o la de implementación de proyectos específicos, de construcción, desarrollo, control, etc. Sin embargo, sea cual sea su propósito, requerirá de inversión y objetivos, metas y recursos, a la vez que debe lograr éxito económico o académico. Se trata así de un proyecto netamente empresarial, sea del tipo de las tradicionales con ánimo de lucro, como las de sin ánimo de lucro, pero que no pueden tener ánimo de pérdida.

Por lo tanto, en este primer paso se debe definir el tipo de empresa con la que se va a trabajar, o si hablamos del desarrollo de software para oficinas internas de una institución educativa entonces se debe conocer los pormenores de dicha oficina, quienes laboran allí y cuales son sus funciones.

En esta etapa hay que ponerse en los zapatos del cliente preguntándose como y que se hizo para llegar a donde esta. El análisis de las actividades de la empresa, oficina o persona para la cual se va a desarrollar el software permite establecer hacia donde se proyectan, es decir donde se ven en un futuro cercano y que papel desempeñara el software que se va a construir dentro de sus planes.

SEGUNDA ETAPA. PROCESO DE PLANIFICACIÓN DEL PROYECTO

Esta etapa se compone de los siguientes pasos:

Paso 1. Consideraciones iniciales: El inicio de un proyecto de desarrollo de software está principalmente influenciado por tres objetivos generales: Resolver un problema, dar respuesta a directivos y aprovechar una oportunidad. Para alcanzar

estos objetivos, las empresas o instituciones emprenden proyectos por una o mas de las siguientes razones: capacidad, control, costo, comunicación y competitividad, como lo menciona Kendall²⁸ en su libro de análisis y diseño de sistemas.

Tomada la decisión acerca del proyecto, agotada la etapa previa del “reconocimiento del negocio”, se debe emprender el proceso de planeación. Al respecto es importante tener en cuenta unas consideraciones que a veces las Universidades, por ejemplo, olvidan, como las que a continuación se describen.

Las actividades de la empresa están influenciadas por su capacidad para procesar transacciones con rapidez y eficiencia. La falta de comunicación es una fuente común de dificultades que afectan a todos los que laboran en una empresa. Sin embargo, muchas empresas aumentan sus vías de comunicación por medios de redes o de desarrollo de la implementación de software que bien desarrollados tratan de ampliar la comunicación y facilitan la integración de funciones individuales.

Muchas empresas han desaparecido y muchas otras imposibilitadas para alcanzar el éxito debido al poco control sobre los costos o por el total desconocimiento para el control de estos. Un software adecuado juega un papel importante tanto con el control como en la reducción de los costos de operación. Eso sin olvidar que la informática, los software, y los sistemas de información son un arma estratégica que puede cambiar la forma en como compete una empresa en el mercado o como una dependencia realiza sus labores. Los sistemas de información mejoran la organización y ayudan a la empresa a ser más competitiva. Por lo contrario si los competidores de la empresa tienen sistemas de información más avanzados, entonces los sistemas de información pueden convertirse en una desventaja competitiva. Por lo tanto las capacidades de los sistemas de información son una consideración importante al formular la estrategia de la empresa.

Sobre esta base el proyecto para la Institución de Educación Superior debe concretar el proceso de planificación teniendo en cuenta los siguientes componentes:

- Planificación del proyecto de software
- Estimación del proyecto
- Análisis de los sistemas de computación
- Modelo de estimación

²⁸ KENDALL, Keneth y KENDALL, Julie. *Análisis y diseño de sistemas*. Tercera edición. Prentice-Hall. México, 1997

- Estudio de viabilidad
- Estudio económico, administrativo y técnico
- Modelado de la arquitectura del Sistema
- Diseño de los sistemas de computación

Paso 2. Definición del equipo de desarrollo: Todo proyecto de sistemas de información debe ser desarrollado bajo las actividades de un grupo de trabajo que se haga responsable del inicio y culminación del sistema de información. El grupo de trabajo va a depender de tamaño de acuerdo al proyecto que va a desarrollarse.

Se deben tener en cuenta los puestos claves de un grupo de trabajo, pero el equipo podría ser más grande o más pequeño o a veces una sola persona puede desarrollar varios puestos. Claro que como se dijo anteriormente el equipo va a depender el tamaño del proyecto. Por tal motivo solo muestra la apreciación personal de acuerdo a la experiencia profesional que se tiene este tema.

Generalmente en las instituciones de educación superior son los estudiantes con sus trabajos de grado, o los grupos de investigación quienes desarrollan la gran mayoría de proyectos de software. Pero las personas que eligen realizar uno de estos trabajos lo hacen porque es un tema que les llama la atención más no porque sean los más adecuados o los que cuentan con las competencias requeridas para dicho proyecto en particular. Así que en esta etapa lo que se recomienda es seguir algunos de los principios de la gerencia de talento humano y de la gerencia por competencias para seleccionar al equipo de desarrollo (sean estudiantes o un grupo de investigación) mas adecuado para el proyecto.

Para realizar esto se debe tener claro de qué se trata el proyecto y así poder definir cuales son las características y competencia necesarias en los miembros del equipo de desarrollo para su correcta realización. Generalmente para esto a nivel empresarial se crearía un formato con el fin de poder conocer a los candidatos, sus habilidades, fortalezas y debilidades. Pero al hacer parte de una institución educativa se cuenta con la ventaja que ya se conoce los datos de todos los posibles candidatos, y aquella información que no este en el sistema o que sea subjetiva se puede fácilmente conseguir con los docentes que han trabajado con estas personas.

En la organización del equipo del proyecto se deben tener en cuenta eso sí unos aspectos fundamentales, que se destacan a continuación.

Todo equipo de desarrollo de software debe contar con un líder de proyecto, un analista de sistemas y un programador o programadores.

Líder de Proyecto: Un líder de proyecto es la persona encargada de aprobar la propuesta o solicitudes de proyectos a llevarse a cabo. El líder del proyecto se encarga de entregar la propuesta una vez aprobada por el a los directivos, con el fin de obtener los recursos ya sea económicos y materiales para el inicio del proyecto.

El líder de proyecto se compromete a entregar el software en la fecha pactada; administra las actividades del grupo de trabajo; verifica y controla las actividades que han sido calendarizadas por él y el analista de sistemas, para llevar a cabo una buena administración de proyectos.

El líder del proyecto es en realidad un gerente; es quien debe hacer que su equipo trabaje mucho, bien, a gusto, y, fruto de ello, obtenga resultados y alcance el objetivo de satisfacer a sus clientes, tanto internos como externos.

Analista de sistemas: Es el encargado de hacer todo lo concerniente al análisis del problema, apoyándose en la aplicación de cuestionarios, entrevistas y observación directa, para determinar las causas concretas del problema planteado, y así poder proponer soluciones adecuadas a este.

El analista de sistemas realiza el bosquejo de la base de datos (estructura de las tablas de la base de datos, diccionarios de datos), así como también se encarga del desarrollo de procedimientos y algoritmos apoyándose en herramientas como diagramas de flujo de datos.

Dentro de sus funciones claves, el analista también es el encargado de realizar la interfaz del usuario. Después debe ofrecer toda esta información al programador para su posterior codificación.

Programador: Es la persona encargada de la codificación de procedimientos y algoritmos que fueron entregados por el analista de sistemas. Cabe mencionar que el programador mantiene una estrecha comunicación con el analista de sistemas, ya que este apoya y asesora al programador en la codificación de los módulos del sistema de información.

El programador además de codificar también tiene a su cargo la prueba de los módulos codificados, con el fin de encontrar errores lógicos y físicos para su resolución.

Una vez realizada las pruebas a los módulos ya codificados y resuelto los errores tanto lógicos como físicos y sabiendo de antemano que todo funciona como uno quiere, el programador tiene a su cargo la elaboración del manual del sistema con

la asesoría del analista de sistemas. Este manual debe contener todos los procedimientos, diagramas de flujo de datos y formas de codificar el sistema con la explicación detallada del por qué fue codificado así, dicho módulo o módulos.

Como una norma gerencial básica en este aspecto se debe tener presente que lo peor que se le puede hacer a un equipo del proyecto, es dejarlo solo y sin apoyo. Si alguien dice que sus equipos no funcionan, generalmente es gracias a que la dirección no se interesa en apoyarlos.

Paso 3. Factibilidad: Una vez establecido el grupo de trabajo se procede a determinar la factibilidad del proyecto. En esta etapa se busca establecer la factibilidad técnica, económica y operacional. Para eso se analizan los recursos financieros, logísticos, técnicos y humanos con los que se cuentan para el proyecto; se establecen presupuestos totales y se hace una organización preeliminar.

Se aplican estudios de factibilidad para saber si se puede resolver el problema o no y al término de esta etapa hay una decisión formal de continuar o no con el proyecto.

Paso 4. Ciclo de Vida del Proyecto: La definición de un ciclo de vida facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo al proyecto. El control de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

Un ciclo de vida para un proyecto se compone de fases sucesivas compuestas por tareas planificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con ciclos de realimentación, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo.

Dado que este modelo se presenta por etapas es fácil crear un ciclo de vida basándose en las mismas. Así se puede seguir el desarrollo del proyecto siguiendo el modelo, y a la vez teniendo control sobre el mismo y sus etapas, ya que el ciclo de vida estará acorde con la metodología de desarrollo.

Paso 5. Actividades: Todas las fases de desarrollo de sistemas de información involucran muchos tipos de actividades diferentes que juntos forman un proyecto. El líder del proyecto debe administrar el proyecto cuidadosamente para que llegue a ser un proyecto exitoso. La administración de proyectos involucra todas las tareas generales de planeación y control.

La planeación incluye todas las actividades requeridas para seleccionar un equipo para análisis de sistemas, la asignación de los miembros del equipo a los proyectos adecuados, la estimación del tiempo requerido para completar cada tarea y la calendarización del proyecto para que las actividades sean terminadas en forma ordenada.

Se pueden contemplar varias técnicas que se pueden utilizar en la realización del calendario. Algunas son muy sencillas y no muestran la interrelación entre las actividades, como son el diagrama de hitos, los diagramas de Gantt. Para mostrar dicha interrelación, se hace necesario el análisis de las redes de precedencia por medio de la técnica PERT. Claro que en esta etapa son muy útiles las herramientas informáticas como Microsoft Project Manager, y otras que pueden ayudar al ordenamiento y control del desarrollo de las actividades del proyecto.

TERCERA ETAPA. ANÁLISIS DE REQUERIMIENTOS.

Esta actividad trata lo relacionado con la interpretación, aclaración y entendimiento de los requerimientos del cliente.

Siempre se debe definir primero los procesos inmersos en el requerimiento y sobre ellos realizar el planteamiento del programa. Es decir primero se debe plantear cuales son los procesos de cada requerimiento y una vez se cuenta con todos los procedimientos de cada requerimiento definidos, entonces se puede comenzar a plantear el programa macro que será la solución general.

Al realizar un análisis de este tipo se debe generar un documento de especificación de requerimientos del software. Este es la descripción completa del comportamiento y la lista de funcionalidades del programa a desarrollar. Dicho documento no tiene un formato estricto y puede variar dependiendo del tipo de empresa, del área en el que se desempeñara el software, del director del proyecto así como de aquello que los miembros del equipo de desarrollo vean pertinente.

El documento de especificación de requerimientos también debe incluir los requerimientos no funcionales que son aquellos que limitan tanto el diseño como la implementación, entre ellos tenemos la confiabilidad, la escalabilidad y el costo.

Es importante dentro del proceso de análisis de requerimientos, sobre todo cuando se trata de proyectos para Instituciones de educación superior, acudir a diversas metodologías, que muchas veces pueden resumirse en los talleres de trabajo y entrevistas.

Las entrevistas están implícitas en las reuniones, juntas, conferencias, diálogos y demás espacios donde se analizan los requerimientos mediante el método de pregunta y respuesta. Es preciso tener preparado con anticipación al encuentro un cuestionario, o un libreto de los temas que se necesita abordar para conseguir la

información requerida. Sin embargo, sobre la marcha de la reunión pueden aparecer nuevas preguntas que complementen el interrogatorio.

Se recomienda grabar la entrevista en audio y/o vídeo con el fin de repasar lo conversado y sobre todo para realizar su transcripción en los documentos de soporte establecidos. Toda entrevista debe estar debidamente documentada.

CUARTA ETAPA. ANÁLISIS DEL SISTEMA.

En esta etapa se deberá aplicar un análisis a los sistemas que están involucrados dentro del proyecto para al final poder crear un algoritmo que muestre claramente su funcionamiento y en base a este construir los prototipos que planteen una solución inicial.

Dentro de la ingeniería de sistemas y más específicamente dentro de la ingeniería de software existen varios métodos de análisis y diseño de sistemas. Todos son válidos a la hora de comenzar a desarrollar una herramienta informática y la elección dependerá del director del proyecto y del equipo de desarrollo. Lo importante es que se realice un análisis minucioso y siguiendo todos los pasos del modelo de análisis y diseño de sistemas que se adopte.

Por otro lado la presente metodología sugiere incluir además el análisis documental, las funcionalidades y la definición del sistema.

A. Análisis Documental.

Por lo general todo proceso debe cumplir con un marco regulado por documentos (leyes, decretos, normas, políticas, circulares, principios, etc.). Es necesario entregar al equipo de desarrollo dichos documentos y una explicación de los procesos involucrados ya que lo escrito en estos se puede convertir en límites de un proyecto. Por esto el análisis documental ayuda a entender y precisar los requerimientos.

B. Funcionalidades.

Una funcionalidad es una parte del programa que tiene significado para el cliente y debe ser ejecutable, medible, probable y descrita con suficiente detalle para mejor comprensión del programador.

Las funciones dentro de los procesos de una empresa pueden convertirse fácilmente en algoritmos que ayudaran a esclarecer el problema que se pretende solucionar con el software a desarrollar.

La lista de funcionalidades es el eje fundamental de esta metodología. Las funcionalidades complejas se deben desagregar en funcionalidades más simples y

de más fácil comprensión. Cada funcionalidad se desarrolla individualmente y precisa su propio diseño, implementación, pruebas, aceptación e integración.

C. Definición del Sistema.

Con la información analizada previamente y aplicando el análisis de sistemas se puede presentar el funcionamiento de la empresa o del tipo de proyecto, por medio de algoritmos o casos de uso, con los cuales se podrá reconocer hasta el más mínimo proceso del sistema, así como será mucho más fácil encontrar los posibles problemas a solucionar.

Un caso de uso es una técnica usada para plasmar los requerimientos funcionales de un sistema. Los casos de uso se describen en un lenguaje de usuario como UML, no deben describirse con terminología técnica, para que puedan ser leídos tanto por el cliente como por el programador. Esta metodología establece que debe haber por lo general un caso de uso por cada funcionalidad. Cuando una funcionalidad presenta variantes en su procedimiento es preciso crear un caso de uso para cada comportamiento.

QUINTA ETAPA. DEFINICIÓN DE LA SOLUCIÓN

Es el proceso que se realiza para resolver el problema y plantear la solución.

En este proceso se deben tener en cuenta las definiciones de la arquitectura, la plataforma, el cronograma de desarrollo y el plan de entregas.

a. Definición de la arquitectura.

Implica la estructura del sistema que comprende: los componentes, las propiedades externas de estos componentes y la relación entre ellos.

Un componente es todo elemento del sistema que ofrece un servicio predefinido.

Un componente puede ser: un servicio web, una biblioteca de clases, un programa, un dispositivo, un driver, etc.

Algunos patrones o estilos de arquitectura son: cliente-servidor, modelo por capas, computación distribuida, Puesto-A-Puesto, por plugins, orientada a servicios, entre otras.

b. Definición de la plataforma.

La plataforma hace referencia al hardware y software usado tanto para el desarrollo como para la implantación. Se debe definir el ambiente tanto de

desarrollo como el de implantación: hardware, computadores, impresoras, sistema operativo, lenguaje de desarrollo, base de datos, drivers, dispositivos, etc.

c. Definición del Cronograma de Desarrollo.

El costo de un proyecto se fundamenta en los tiempos que toma el desarrollo de cada actividad.

La duración de cada actividad se debe estimar en horas hombre.

Entre los desarrolladores hay la costumbre de subestimar los tiempos de cada tarea basándose en suposiciones y comparativos apresurados e informales. Es preciso por esto que la definición de tiempos esté acorde con la realidad bajo el análisis realizado.

Siempre hay que desconfiar de los cronogramas con tiempos muy cortos. Y se debe alejar del análisis de tiempos las presiones y premuras que provienen del solicitante, usuarios finales, administrador del proyecto y la misma empresa. Se recomienda que los días laborales no sobrepasen la carga normal de horas de trabajo. Las largas jornadas laborales no siempre llevan a acelerar las tareas.

En esta etapa ya es realmente necesaria una herramienta informática como Microsoft Project Manager, o alguna similar ya que ahora existen varias en la categoría de software libre que pueden realizar las mismas tareas. Estas herramientas brindan una mejor manera, mas practica y ordenada, de manejar un proyecto y sus recurso. Así mismo es muy permiten tener un control sobre los tiempos y el cumplimiento de las diferentes etapas de desarrollo.

d. Plan de entregas.

El plan de entregas es la planificación del desarrollo de las funcionalidades por grupos o módulos con el fin de entregar un programa totalmente funcional y listo para entrar a producción.

Para cada entrega se debe planificar las funcionalidades a desarrollar y el periodo de tiempo que tomará su desarrollo.

SEXTA ETAPA. DISEÑO Y DESARROLLO

Al igual que para el análisis, para el diseño existen diversas metodologías que otorgan herramientas para el diseño de un nuevo sistema, y así mismo depende del equipo de desarrollo escoger la mas adecuada para realizar la etapa de diseño. Así es que una vez definida una solución se puede comenzar a subdividir el sistema en módulos o procedimientos que pueden programarse o desarrollarse

paralelamente. Una vez se ha conseguido esto se crean prototipos para cada subsistema, modulo y/o procedimiento.

En esta etapa el analista usa la información recolectada anteriormente para realizar el diseño lógico de sistema de información.

Parte del diseño de sistema de información es la de diseñar la interfaz de usuario, en este caso la Institución de Educación Superior.

La interfaz conecta al usuario con el sistema; es el puente de comunicación y por lo tanto es extremadamente importante realizar un buen diseño.

Un ejemplo de interfaz de usuario incluye un teclado para introducir preguntas y respuestas, menús en pantalla para elegir comandos del usuario y un ratón para seleccionar opciones.

Dentro de la fase de diseño se incluye (si el software lo requiere) el diseño de la base de datos, la cual guardará la mayor parte de datos necesarios para los tomadores de decisiones de la empresa o de la Institución de educación superior. Una base de datos bien diseñada da como resultado una base de datos bien organizada que es el pilar para todo software de gestión, técnico o sistemas de información. En esta etapa el analista también trabaja con los usuarios para diseñar la salida de información de esas bases, salida que puede ser en pantalla o impresa, según como se satisfaga las necesidades de información.

Por ultimo el analista debe diseñar procedimientos de control y respaldo para proteger el sistema y a los datos. Los documentos que contengan las especificaciones de diseño se representaran por medio de diagramas de flujo, tablas, símbolos especiales, árboles, graficas, etc.

Los diseñadores son los responsables de dar a los programadores, las especificaciones del sistema de información completas y claramente delineadas.

Una vez comenzada la fase de programación, los diseñadores contestarán las preguntas y dudas que tengan los programadores, cuando utilicen las especificaciones de diseño, por eso es recomendable que el diseño sea lo mas claro y preciso en sus especificaciones.

Es importante resaltar que en la etapa de desarrollo el analista debe trabajar junto con el programador, para desarrollar cualquier sistema que se necesite, tomando como guía el diseño específico de sistemas.

Los programadores tienen un papel principal en esta etapa ya que son los encargados de la codificación de los módulos correspondientes, así como también

de la verificación de sintaxis en el código, para encontrar errores y ser resueltos por ellos mismos.

El programador también valida cada uno de los módulos programados y realiza pruebas integrales a cada módulo. Además los programadores también son responsables de la documentación del sistema; son encargados de elaborar el manual del usuario que sirve al cliente final para aprender a manejar el nuevo sistema, y el manual del sistema en donde viene la explicación de la forma de programar los módulos. Así mismo tiene a su cargo todo lo concerniente a los procedimientos empleados en la programación de cada módulo. Esta documentación es de vital importancia para probar el sistema y posteriormente para su mantenimiento, una vez que haya sido implantado el sistema.

Como se debe tener en cuenta la premisa gerencial de la rentabilidad y productividad, lo más recomendable es utilizar el modelo de desarrollo de software basado en componentes.

El Desarrollo de Software Basado en Componentes tiene muchas ventajas. Una de las principales se basa en la reutilización de componentes. De esta manera, los componentes se los diseña con el objetivo de poder ser reutilizados en otras etapas o incluso en otros proyectos, favoreciendo a la necesidad de realizar software complejo en cortos periodos de tiempo y a la vez con un menor esfuerzo en la realización del mismo. Sobre esta base los costos son mucho menores que desarrollar un software desde cero. Además la reutilización de un código previamente realizado y probado nos facilita el trabajo, mejorando la fiabilidad del producto final.

Estos componentes de software ya desarrollados, son combinados adecuadamente para obtener un producto final de buena calidad, pues dado que un componente puede ser construido y luego mejorado continuamente, la calidad de un software basado en componentes mejorará con el tiempo. Para esto hay que realizar una precisa búsqueda y selección de componentes apropiados que se emplearán en el desarrollo del software. Con esto hay que tener especial cuidado pues si al ensamblar los distintos componentes, uno de ellos no es confiable, el software en cuestión tendría fallas.

En resumen el Desarrollo de Software Basado en Componentes, en términos gerenciales de economía, tiene como ventajas el ahorro de tiempo, dinero y esfuerzo, por la reutilización de componentes. Además se obtiene un software de buena calidad, ya que muchos de los componentes ya han sido probados en su ambiente real.

En sus aspectos particulares en la fase de diseño y desarrollo del proyecto de software, en especial para Instituciones de Educación Superior, se deben

considerar la especificación de módulos, la construcción de prototipos, el diseño de la base de datos y de la interfaz del usuario y las pruebas respectivas.

a. Especificación de Módulos.

Se recomienda que los módulos puedan construirse de manera paralela, casi independientes unos de otros. Así mismo dentro de cada modulo se deben definir los procedimientos que conforman a cada uno y revisar si existen similitudes entre los procedimientos de módulos diferentes. Al hacer esto se puede evitar redundancia en la programación, ya que un procedimiento podría servir para más de un modulo.

b. Construcción de Prototipos de Software.

Es frecuente que los clientes no sepan lo que quieren, pero cuando ven algo y lo utilizan, pronto saben lo que no quieren. Esto se da mucho en las instituciones educativas. Por esta razón un prototipo es de gran ayuda.

Un prototipo es una representación limitada del **diseño** de un software que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales, probar independientemente sus partes y explorar su uso. Un prototipo puede ser cualquier cosa, desde un algoritmo en papel hasta un complejo programa, que puede ejecutarse para evaluar su funcionamiento.

Los prototipos responden a preguntas y apoyan el trabajo de los diseñadores probando ideas, clarificando requisitos o definiendo alternativas. Los prototipos de software son útiles para comunicar, discutir y definir ideas entre los **diseñadores** y las partes responsables.

Los prototipos apoyan el trabajo evaluando el funcionamiento, el cumplimiento de requerimientos, clarificando **requisitos de usuario** y definiendo alternativas.

En esta etapa se recomienda aplicar las pautas del modelo de prototipos para el desarrollo de software. Se trata de un modelo que funciona para crear programas completos en base a la creación de prototipos, y explica muy bien como construir un prototipo y cuales son sus ventajas.

c. Diseño de base de datos.

El diagrama Entidad-Relación es el pilar del diseño de base de datos, pero este es solo el comienzo. La idea es desarrollar un script basado en SQL de creación y/o modificación de la estructura de datos.

Se deben especificar las tablas, campos y relaciones y definir los índices, llaves primarias, llaves foráneas, constreñimientos, datos por defecto, etc.

d. Diseño de interfaz de usuario.

Esta es la parte visual del programa y la que más aprecia el usuario final. Algunos desarrolladores de software suelen minimizar la importancia de la interfaz de usuario, pero esta es tan importante como el programa mismo.

Al usuario final se le debe brindar un contacto amigable con el software, o sea, que sea fácil de usar, intuitivo, estándar y personalizable. Para esto debe tenerse en cuenta los colores, el contraste, el tamaño de las ventanas, la ergonomía y todas las pautas del diseño gráfico y multimedial.

e. Pruebas.

Antes de implantar el sistema es necesario realizarle pruebas para saber si funciona de acuerdo con las especificaciones y en la forma en que los usuarios, las Universidades, esperan que lo haga.

Estas pruebas consisten en hacer funcionar al sistema como si estuviera realizando las operaciones cotidianas para lo cual fue desarrollado. Para ello se introducen entradas de conjunto de datos para su procesamiento y después se examinan sus salidas o resultados.

Es conveniente que las pruebas sean realizadas por personas ajenas al proyecto para que estas tengan validez; de lo contrario se comete el error de realizar pruebas guiadas, es decir, hacer pruebas sabiendo de antemano los resultados de estos y no obtener resultados favorables.

Entre las pruebas necesarias están las de desarrollo y de usuario.

Pruebas de desarrollo. Las pruebas deben ser realizadas aun en el ambiente de desarrollo pero buscando un contexto lo más parecido a la realidad. En lo posible deben usar datos reales, tanto para las bases de datos como para inspeccionar los procesos del software.

La realización de este paso incluye: interpretación, creación, reutilización, refinamiento, integración, compilación y prueba del código fuente. También hace parte de esta actividad el control de errores, manejo de transacciones, empaquetamiento de clases, comunicación entre desarrolladores, integración de código fuente, etc.

Pruebas de usuario. Muchas veces se permite a los usuarios finales (aquellos usuarios que usarán el sistema constantemente) utilizar el sistema como ellos lo usarían, sin limitarlos; es decir, dejarlos en forma libre manejarlo a su antojo, para así poder detectar fallas o errores, para no encontrados en el proceso de desarrollo del sistema.

Claro está que es difícil hacer pruebas al sistema y no encontrar ningún error, ya que los errores nos ayudan a mejorar nuestros sistemas; si no tuviéramos errores realmente no sabríamos si todo está bien o de lo contrario que todo está mal.

SÉPTIMA ETAPA. VALIDACIÓN

Una vez desarrollado todo el proceso anterior se ha llegado al punto de la validación. Es donde la institución o la empresa toma la decisión sobre la aceptación del proyecto y el paso o no a las etapas finales. Consta de la aceptación y la creación de manuales.

a. Aceptación.

De acuerdo con las pruebas, el usuario, la Universidad, determina la aceptación del paso a producción del programa con las nuevas funcionalidades. La no aceptación implica un requerimiento de corrección al desarrollador y la realización de nuevas pruebas.

b. Creación de manuales.

Se deben crear el manual de usuario y el técnico. El de manual de usuario requiere contener ejemplos para la realización de cada proceso; debe ser contextual y brindar la posibilidad de solicitar la ayuda en línea. Por otro lado el manual técnico debe tener una sección que indique las diferencias con el sistema anterior (si lo existía), los requerimientos del sistema y otras opciones de control y administración.

OCTAVA ETAPA. IMPLANTACIÓN.

El desarrollo del proyecto de software en esta etapa llega a lo que la planificación estratégica denomina “el hacer”, que en definitiva es la concreta, la real. Se trata entonces de implantar el sistema diseñado para la institución, empresa o cliente.

Por implantación se define al proceso de instalar, verificar y adecuar el nuevo software. En su desarrollo hay que llevar a cabo la integración y entrega, las capacitaciones respectivas y efectuar lo que se denomina como la “post-implantación”.

a. Integración y entrega.

En este momento se crea ya el software como tal, uniendo los módulos, las clases, los procedimientos, las bases de datos, la interfaz de usuario, en un solo programa que será el producto final y funcional del proyecto.

En ese proceso debe generarse el programa ejecutable y/o instalador según sea el caso.

Se entrega el programa y los scripts de creación de la estructura de datos, en lo cual hay necesidad de tener muy claras las versiones para no crear confusión.

b. Capacitaciones.

Se trata de un proceso de “coaching”, como se denomina en la gerencia del talento humano. Hay necesidad de preparar, ilustrar y entrenar a quienes serán los ejecutores del software.

Esta capacitación no se debe realizar en un lenguaje técnico sino basado en la terminología del proceso en que se apoya.

c. Post – Implantación.

Un nuevo software gracias a su utilidad tiende a crear dependencia, por lo tanto las actividades posteriores al desarrollo del mismo, también se consideran parte del funcionamiento del sistema.

Esta etapa incluye actividades como la realización de copias de respaldo, actualización de componentes, soporte técnico, gestión de errores y gestión de mejoras.

4.3 MODELO DE UN SOFTWARE EDUCATIVO

a. Objetivo general

Desarrollar un Software Educativo como Apoyo Didáctico a un Sistema Presencial-Virtual para Informática Educativa bajo Plataforma Web

b. Desarrollo del software

Los modelos de metodologías para el desarrollo de software educativo son lineales y no lineales. Las fases en un modelo lineal pueden ser claras, diferencian las tareas a ejecutar, pero cuando se aplican se observa como la separación de tareas y la linealidad es difícil de mantener. La práctica ha demostrado que es

difícil cerrar fases hasta que el software esté terminado. Hay una revisión continua en función de los resultados que se van obteniendo.

Las razones antes expuestas han motivado la utilización de modelos no lineales, como por ejemplo el modelo de desarrollo rápido de prototipos de Tripp & Bichel-Meyer²⁹.

Existen diversas metodologías para el desarrollo de software educativo, propuestas por diferentes autores, en las cuales se pueden notar fases tanto comunes como diferentes, pero dirigiendo la atención a las distintas actividades que se realizan, éstas en su mayoría son comunes. Cualquier metodología permitirá llegar a buen término la realización de un software, sin embargo, es preciso dividir las fases en pasos adecuados al contexto en que se encuentra el desarrollador.

Tratando de cumplir ésta premisa se propone utilizar una metodología híbrida no lineal, la cuál consta de seis fases: Planeación, Diseño, Producción, Prueba Piloto, Evaluación y Mejoramiento.

Se propone así mismo un ciclo en las fases diseño-producción-prueba piloto y luego culminada la versión 1 un ciclo evaluación-mejoramiento.

Posteriormente, en las futuras evaluaciones, se propone tomar la decisión de revisar los documentos desde el principio, es decir retomar la fase de planeación.

Esta metodología propuesta la llamaremos Metodología TESEO.

Gráficamente el proceso de desarrollo propuesto para un software educativo se presenta en la figura 13.

c. Fases de la metodología propuesta

Son seis las fases propuestas, que en el proceso de investigación para este trabajo de grado fueron cumplidas en las que corresponden a las iniciales, sobre todo en lo referente a la planificación.

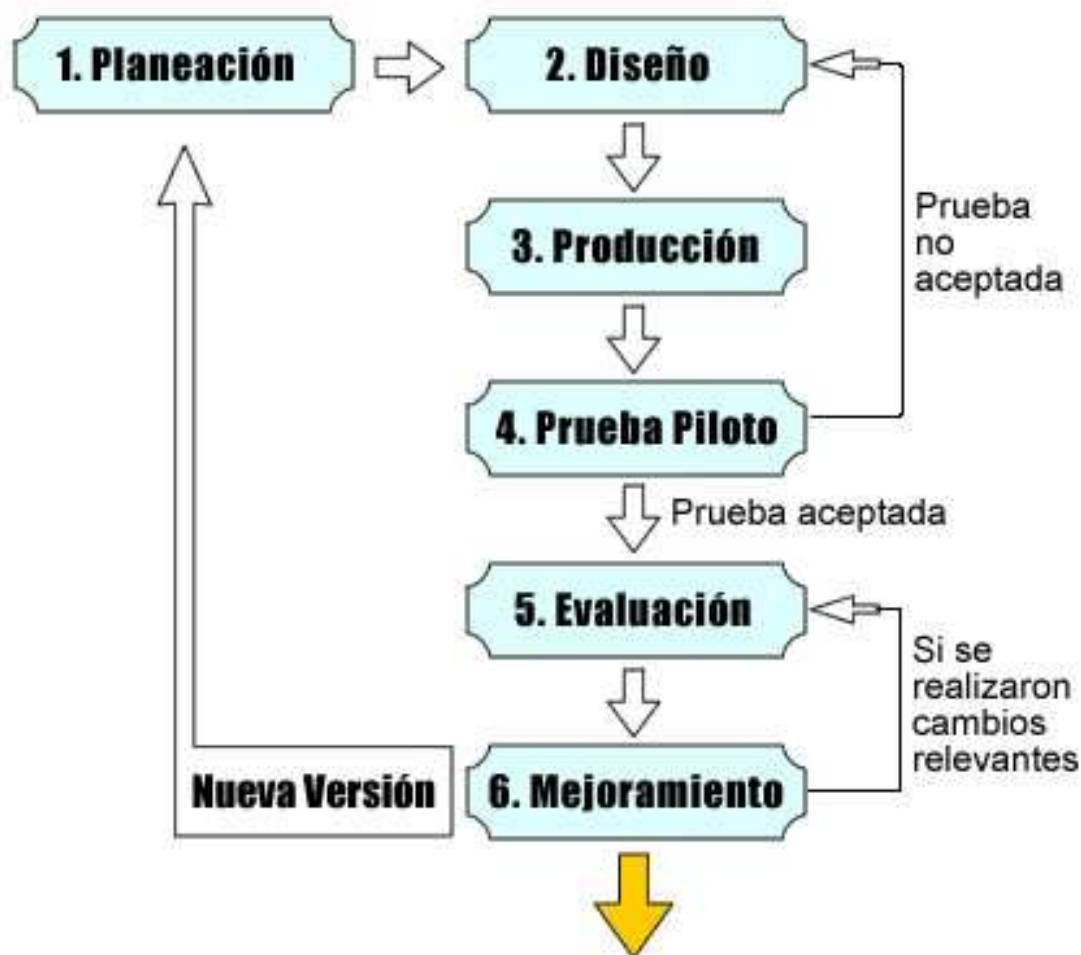
Fase 1. Planeación

Idea inicial: La idea inicial constituye lo que se quiere crear, contiene el qué (materia) y el cómo (estrategia didáctica). Con esta idea se identifican las

²⁹ GROS, B. *Diseños y Programas Educativos*. Ed. Ariel S.A. 1ra Edición. España, 1997.

necesidades, se reconoce el problema definiendo el propósito del software y se organiza un plan de actividades en donde se define el tiempo de desarrollo.

Figura 13.



En la propuesta se tuvo la idea inicial de Desarrollar un Software Educativo de Apoyo Didáctico.

En el proceso se llevó a cabo una reunión de trabajo entre un grupo de docentes y estudiantes previamente seleccionados por sus conocimientos del tema para identificar las posibles necesidades de desarrollo de la idea.

La identificación de las necesidades permitió reconocer el o los problemas, detectando a qué se debieron y qué puede contribuir a su solución. Una vez estudiado el problema se realizó un análisis de las necesidades y se culminó esta fase con la formulación de un plan para llevar a cabo las siguientes etapas.

Sobre esta base se realizó una consulta acerca de los recursos disponibles y las alternativas de usarlos para cada una de las etapas siguientes.

En esta fase se establecieron las condiciones de utilización y que resultados se esperaban.

Con este fin se determinaron las características de los futuros usuarios, los contenidos a desarrollar, el tipo de software que se debería generar, el lenguaje de programación y los recursos necesarios para llegar a feliz término.

Posteriormente, en términos gerenciales, se hizo el estudio de requerimientos en lo referente a personal y tiempo, así como los recursos computacionales que se necesitan para cada fase.

Las actividades anteriores corresponden a las fases de “análisis” o “definición de requerimientos”, que se han descrito en el modelo gerencial ya expuesto.

En términos de desarrollo del software se estudiaron los siguientes aspectos que se propone se aplique para otros proyectos:

Contenido técnico. Responde a los siguientes interrogantes:

- ¿Existe tecnología para realizar el software?
- ¿Con cuántos y con cuáles equipos se cuentan?
- ¿Puede trabajar el software con estos equipos?
- ¿Se necesitan nuevas tecnologías y posibilidad de desarrollarlas?

Contenido operativo. Para realizar el estudio se resolvieron los siguientes cuestionamientos, que también se propone plantear para otros tipos de proyecto:

- ¿Habrà resistencia al cambio?
- ¿Se adaptarán fácilmente los usuarios a utilizar el software?
- ¿Disminuirá el tiempo de operatividad?

Contenido gerencial-económico. El estudio realizado y propuesto se remitió básicamente a encontrar respuestas a los siguientes aspectos:

- ¿La relación costo-beneficio está dentro de las expectativas?
- ¿El costo será bajo?
- ¿Mejorarán los beneficios?
- ¿Los beneficios que se obtengan serán suficientes para aceptar los costos?

Fase 2. Diseño

El diseño es un borrador de lo que será el producto final.

En esta fase se estructuraron los contenidos que la fase de planeación determinó que serán desarrollados en el software, se elaboró un mapa de navegación para el software, el cual permite dividir en partes. Cada parte da como resultado un módulo. La cantidad y tamaño de los módulos depende de lo extenso de los contenidos.

Para los proyectos similares se propone seguir el proceso llevado a cabo y que se describe en los párrafos siguientes.

Se desarrollaron tres sub-tareas: el diseño instruccional, el diseño de la interfaz y el diseño computacional, las cuales se detallan a continuación.

El diseño instruccional debe contener: el marco teórico que sustentará el plan instruccional, definición del perfil de los usuarios, diseño de pantallas y notas de producción.

El plan instruccional debe indicar los objetivos generales, objetivos terminales por unidad de contenido, objetivos específicos, contenidos programáticos, estrategias de aprendizaje (eventos), recursos y estrategias de evaluación. En general el diseño instruccional representa el software educativo en el papel.

El diseño de la interfaz se refiere a la formas de interacción que los usuarios tendrán con el software. Se indican las posibles opciones que serán consideradas a la hora de realizar los menús, se establecen las áreas de la pantalla según el uso que se destine (área de botones, área de información, área de ayuda, etc.). Se establecen cuales dispositivos y cuales códigos o mensajes se utilizarán para la comunicación entre el usuario y el software.

El diseño computacional especifica los elementos multimediales de información (captura, digitalización, edición y almacenamiento de sonido, imágenes, video, animación; la transcripción de texto, etc.), así como también la producción de los distintos botones y fondos, para luego integrarlos y obtener las distintas pantallas a utilizar.

Fase 3. Producción

En esta fase se construye el software, se integran los elementos multimediales produciéndose las distintas pantallas, se crean y se enlazan los elementos correspondientes. Se materializa el borrador efectuado en la fase del diseño.

Se elabora un prototipo. Este en realidad es un modelo piloto o de prueba; evoluciona con el uso y es objeto de estudio por todo el equipo de desarrollo con la finalidad de mantener el diseño o realizar los correctivos pertinentes.

El prototipo funcionará con todas las opciones principales previstas en el diseño, pero con las bases de datos aún incompletas, sin depurar y con gráficos provisionales. El prototipo será elaborado por módulos hasta que contenga todos los módulos contemplados en el diseño.

Cumplidas estas tareas se ha logrado desarrollar el prototipo completo. Este prototipo corresponde a lo que se puede denominar como la primera versión del software.

Fase 4. Prueba Piloto

La prueba piloto tiene como finalidad depurar el prototipo a partir de su utilización por un grupo de usuarios (profesores y estudiantes).

Esta prueba hace que el usuario participe de manera más directa en la experiencia de análisis y diseño. Los usuarios pueden señalar las características que les agrada o no tener, junto con los problemas que presenta un sistema que existe y funciona, con mayor facilidad que cuando se les pide que las describa en forma teórica o por escrito.

Por otra parte en esta fase es posible hacer una evaluación inicial que permita analizar las posibles modificaciones o corrección de errores, tanto en la interfaz, como en los otros aspectos considerados en su elaboración.

Una vez terminada la prueba se utilizan sus resultados para revisar el diseño y la producción. Este proceso se debe repetir hasta que los integrantes del equipo de desarrollo decidan que el prototipo no requiere más revisión.

Fase 5. Evaluación

Ya sobre este punto se trata de llevar a cabo las mismas actividades que en la prueba piloto. La diferencia es que en esta fase se debe emplear una metodología formal, acorde al método científico. Para llevar a cabo el proceso, entonces, se debe realizar un muestreo e tipo probabilística. Es decir que en síntesis se debe tomar una muestra representativa de los usuarios a quien va dirigido. Se trata con ello de corroborar que los objetivos para los cuales se desarrollo el software se cumplen en su totalidad.

Fase 6. Mejoramiento

En esta fase se debe mejorar, corregir, agregar nuevos módulos de tal manera que se pueda tener a futuro una serie de versiones que vayan adaptándose a cambios de currículo, nuevos hardware, nuevos dispositivos, etc.

5. CONCLUSIONES Y RECOMENDACIONES

Al llegar al final del trabajo de grado se pueden sintetizar las siguientes conclusiones y recomendaciones.

5.1 CONCLUSIONES

- Es posible hallar respuesta a la dicotomía que existe entre el desarrollo de un proyecto de software en su parte técnica, su ingeniería, y la gerencia que se debe ejercer en la empresa de hacerlo real, de ejecutarlo, de administrarlo eficientemente, garantizando calidad, permanencia, competitividad y éxito en todos los campos.
- Existen caminos gerenciales para administrar óptimamente el desarrollo de los proyectos de software en organizaciones que actualmente, junto a las empresas comerciales, están especialmente dedicadas a ese propósito, como son las Universidades y demás Instituciones de Educación Superior
- Las metodologías de desarrollo tecnológico y los modelos de gestión de proyectos no son excluyentes. Por esto las empresas desarrolladoras de software han entendido que al manejar su desarrollo como un negocio que incluye productos que prestan servicios y que con una buena administración al desarrollar los proyectos se puede obtener una empresa rentable y con un amplio espectro de posibilidades.
- En relación a la gerencia de proyectos en general, aplicable a los de software, nueve son las áreas fundamentales que se refieren a su administración, a saber: Administración de la Integración, del Alcance, del tiempo, de los costos, de la calidad, de los recursos humanos, de las comunicaciones, del riesgo y de la procuración del proyecto.
- La propuesta gerencial del desarrollo de proyectos de software que se elabora en este trabajo se basa en un modelo espiral para desarrollo de software con cuatro ciclos y cuatro cuadrantes, que comprenden la prueba de concepto, la primera y segunda construcciones y el ciclo final.
- La propuesta gerencial básica de un proyecto de software de este trabajo se compone de ocho etapas: análisis previo, planificación, análisis de requerimientos, análisis del sistema, definición de la solución, diseño y desarrollo, validación e implantación del proyecto.

- En un proyecto de software educativo la propuesta gerencial se transforma en seis fases como son: Planeación, Diseño, Producción, Prueba Piloto, Evaluación y Mejoramiento.

5.2 RECOMENDACIONES

Para llevar a cabo exitosamente un proyecto de desarrollo de software, en empresas típicas o en Instituciones de Educación Superior, se recomienda especialmente, en términos de la gerencia:

- Comprender que un proyecto, cuando se trata de desarrollos de software, debe gerenciarse desde dos enfoques congruentes. Por una parte el área administrativa propiamente dicha, con los elementos de las teorías organizacionales; por otra, la del Project Management.
- El proyecto debe enfatizar su gestión en la Dirección Estratégica Integrada. Ella debe tener en cuenta la integración de la Dirección Integrada de Proyectos (DIP) y la Dirección Integrada de la Calidad (DIC) con el objetivo de lograr el éxito de los proyectos.
- La necesidad de integración de la DIP y la DIC en un sistema informático único se logra a partir de la programación estructurada de tareas y subproyectos que brinda el MS Project con el objetivo de garantizar la evaluación por cortes de los proyectos en función de la toma de decisiones de la gerencia empresarial haciendo uso del tablero de comandos.
- Todo debe comenzar con una planificación del proyecto de Software pensando no solo en tecnología sino también en rentabilidad. Su objetivo es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costos y planificación temporal.
- Es muy importante tener en cuenta el factor tiempo, ya que el desarrollo de la programación requiere determinar fechas de comienzo y finalización para las actividades del proyecto. Si las fechas de comienzo y finalización no son realistas, el proyecto tendrá pocas probabilidades de terminar. A la vez es necesario para programar los procesos que proveen entradas, especialmente la estimación de las duraciones y de costos.

BIBLIOGRAFÍA

BECK, K. *Extreme Programming Explained. Embrace Change: Pearson Education*. Traducido al español como: Una explicación de la programación extrema. Aceptar el cambio., Addison Wesley, 2000.

BOEHM, B. A. *Spiral Model of Software Development and Enhancemen*. IEEE Computer. Vol. 21, # 5. 1988.

CROSBY, P. *Dinámica gerencial*. México. McGraw Hill, 1988

CUEVAS, Gonzalo. *Gestión del Proceso de Software*. Editorial Universitaria Ramon Areces, 2004.

CHUNG, K. y MEGGINSON, L. *Organizational behavior*. New York: Harper and Row, publishers, 1991.

DELGADO, R. VEREZ, M. *La Dirección Integrada de Proyectos haciendo uso de las tecnologías de la informática y las comunicaciones en el marco del perfeccionamiento empresarial*. Monografías.com Internet. 2003

DELGADO R.; MONTES M. y TORRES M. *La Dirección Integrada de Proyectos apoyada por las bases de datos*. Ponencia. Forum de Ciencia y Tecnica. Junio/2003

DELGADO R.; VEREZ M.; ROEHRIC K. U. Humboldt. *La Dirección Integrada de Proyectos (Project Management) haciendo uso de las Nuevas Tecnologías de la Informática y las comunicaciones aplicadas al CGLALE*. Kologische Hefte der wirtschaft. Helf 1 b / 2 001.

DE MARTÍNEZ, JULIÁN. *Teoría General de la Administración*. ESAP, Modelo UAD, Bogota, 1990.

DIMOCK, Marshal. *Principios y Normas de Administración*. Ed. Libreros Mejicanos Unidos, México. 2004.

EDUCAR. *Clásicos de la Administración*. Bogota, 1980

GAGNE, R. *Instructional technology*. En Pressman Fundatios Hillsdale, Lawrece Erlbaum Associates.1987.

GALVIS, ALVARO . *Ingeniería de software educativo*. Ediciones Uniandes. Bogotá, 1992.

GARCÍA ÁVILA, Lourdes. *Modelo para la evaluación de la calidad del análisis y diseño orientados a objetos de sistemas informáticos (CADOOSI)*. Tesis de doctorado. Universidad Central de Las Villas, 2000.

GROS, B. *Diseños y Programas Educativos*. Ed. Ariel S.A. 1ra Edición. España, 1997

INFORMATIZATE <http://www.informatizate.net/> (acceso Abril 2008)

ISO 9000. *Sistemas de gestión de la calidad. Fundamentos y vocabulario*.

JACOBSON, I., BOOCH, G., RUMBAUGH, J. *The Unified Software Process*. Addison- Wesley. 1998.

KENDALL, Keneth y KENDALL, Julie. *Análisis Y Diseño De Sistemas*. Tercera edición. Prentice-Hall. México.

KORTH, Henry F.; SILBERSCHATZ, Abraham. *Análisis y Diseño de Sistemas*. Segunda Edición. Mc Graw Hill, 2003

KRYGIER, A. Consultores de Gerencia. *¿Terapeútas de las organizaciones?* Revista Gerente. Mayo, 1988.

LEVINE, Harvey. *Gerencia práctica de proyectos*. Mac Graw Hill, 2006

MARDONES E. y RUIZ M. *La Informática Educativa, un Factor de Cambio para la Educación Venezolana del Siglo XXI*. Revista Encuentro Educativo. Vol. 2 No. 2. Universidad del Zulia. Facultad de Humanidades y Educación. Centro de Documentación e Investigación Pedagógica. 1996.

MUENCH, Dean. *The Sybase Development Framework*. Oakland Calif. Sybase Inc., 1994.

OBJECT MANAGEMENT GROUP (OMG) <http://www.omg.org/> (acceso Marzo 2008)

PRESSMAN, Roger S. *Ingeniería del Software: Un enfoque práctico*. Segunda edición, Editorial McGraw Hill, 1990

RANDOLPH, W. Arnold/ POSNER, Barry Z. *Gerencia de proyectos. Como dirigir exitosamente equipos de trabajo*. Mc Graw Hill, Bogotá, 2000.

SAWYER, P. Y KONTOYA, G. *Software Requirements Engineering Knowledge Area Description*. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.

STONER, James. *Administración*. Editorial Prentice Hall, México, 1992.
THE AMERICAN HERITAGE DICTIONARY OF THE ENGLISH LANGUAGE, Third Edition. Boston, Mass.:Houghton Mifflin Company. 1992.

SEQUERA, M. *Diseño de una propuesta curricular para una maestría en informática educativa*. trabajo especial de grado para optar al título de magíster en educación área de planificación y administración educativa. Maracaibo. Universidad del Zulia,1999.

SENA. Curso Virtual. *Administración de Proyectos de Software*.

ZAMBRANO, J. *Enseñanza asistida por computador y producción de software educativo (PROSDOS)*. Caracas, Venezuela. Imprenta Universitaria Universidad Central de Venezuela,1995
Curso virtual del Sena: