

Universidad de las Ciencias Informáticas

Facultad 3



Título:

Estrategia de Planificación del Proyecto Informatización del
Convenio Cuba Venezuela.

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

Autores:

Richar Peña Trujillo.

Armando Rodríguez Hernández.

Tutor:

Ing. Nahuel Massón Padilla.

Ciudad Habana, Junio 2008.

*"Es de importancia para quien desee alcanzar una certeza en su investigación,
el saber dudar a tiempo."*

Aristóteles.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Richar Peña Trujillo

Firma del Autor

Armando Rodríguez Hernández

Firma del Autor

Ing. Nahuel Massón Padilla

Firma del Tutor

Dedicatoria

A mi padre y a mi abuela que aunque no estén presente físicamente, se que este era uno de sus sueños.

A mis madres Ana Rosa, Belkis Peña y Mary Peña por apoyarme y aconsejarme siempre que lo necesité.

Richar.

A mis padres Armando y Maricel por su amor y protección.

A Claudia y a Mima por su cariño.

A Katia por su comprensión.

Armando.

Agradecimientos

A la Universidad de las Ciencias Informáticas y a todos los profesores que de alguna forma contribuyeron a nuestra formación profesional.

A nuestro tutor, el Ing. Nahuel Massón Padilla por su colaboración.

A nuestros amigos y compañeros por darnos fuerza en este último momento y haber compartido los mejores años de nuestras vidas de estudiantes.

A nuestras familias por su comprensión y apoyo en estos cinco años.

A todos, con nuestra más profunda gratitud.

Richar y Armando.

Resumen

El presente trabajo tiene como base para su elaboración, la necesidad de hacer un estudio detallado de las tareas y recursos tanto humanos como tecnológicos que se deben llevar a cabo en el proyecto Informatización del Convenio Cuba - Venezuela, en un equipo multidisciplinario para la correcta planificación del mismo. El objetivo principal es desarrollar una estrategia de planificación para el proyecto, basándose en las características típicas del proyecto y por las ventajas que esta representa, puesto que permite tener un control y una visión general del trabajo. Para poder cumplir con los objetivos del mismo, primeramente se realizó un estudio de las principales herramientas para implementar la estrategia de planificación que existen, los mecanismos de captura de información para la planificación y seguimiento del proyecto, y los métodos de estimación de tiempos y recursos con que se trabaja en la actualidad. Basado en el estudio anterior se seleccionó las herramientas y métodos que más se ajustaban a las características del trabajo que se iba a desarrollar. Finalmente se describe la solución propuesta y una valoración del trabajo realizado.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Evolución de la planificación.....	5
1.2.1 Las características más importantes de la planificación.....	7
1.2.2 Ventajas de la planificación.....	9
1.2.3 Desventajas de la planificación.....	9
1.3 Actividades para la Planificación de un Proyecto de Software.....	10
1.3.1 Ámbito del software.....	10
1.3.2 Recursos.....	11
1.3.2.1 Recursos Humanos.....	11
1.3.2.2 Recursos de Software Reutilizables.....	12
1.3.2.3 Recursos de entorno (hardware/software).....	13
1.3.3 Técnicas para la estimación de proyectos de software.....	13
1.3.3.1 Modelo de estimación por Puntos de Función.....	14
1.3.3.2 Modelo de estimación por Casos de Uso.....	15
1.3.3.3 Puntos de Característica.....	16
1.3.3.4 Líneas de código.....	16
1.3.3.5 Modelo de estimación por COCOMO II.....	17
1.3.3.6 Comparación entre las técnicas de estimación.....	19
1.3.4 Análisis y Gestión del Riesgo.....	20
1.3.4.1 Riesgo del Software.....	21
1.3.4.2 Identificación del Riesgo.....	22
1.3.4.3 Proyección del Riesgo.....	22
1.3.4.4 Refinamiento del riesgo.....	23
1.3.4.5 Plan de Reducción, Supervisión y Gestión del Riesgo (Plan RSGR).....	23
1.3.4.6 Riesgos y peligros para la seguridad.....	23
1.4 Métodos y herramientas utilizadas para la planificación.....	24
1.4.1 El Diagrama de Gantt:.....	24
1.4.2 Técnica PERT y CPM.....	25
1.4.3 MS Project:.....	26

1.4.4	GanttProject:	28
1.4.5	DotProject:	28
1.4.6	Trac:	30
1.4.7	Open Workbench:	31
1.4.8	KPlato:	31
1.4.9	TaskJuggler:	31
1.4.10	XPlanner:	31
1.4.11	OpenProj:	32
1.4.12	Project Dune:	32
1.4.13	Agile Track:	32
1.1.14	Comparación entre los métodos y herramientas para la planificación.	32
1.5	Métodos y herramientas utilizadas a nivel de Universidad.	36
1.6	Conclusiones del capítulo.	37
Capítulo 2: Definición y Aplicación de la Estrategia de Planificación.		38
2.1	Introducción.	38
2.2	Pasos para desarrollar la planificación.	38
2.2.1	Establecer el ámbito del software que se va a desarrollar.	38
2.2.2	Estimación de los Recursos.	45
2.2.2.1	Recursos Humanos.	45
2.2.2.2	Recursos de Software Reutilizables.	46
2.2.2.3	Recursos de entorno (hardware/software). Herramientas de hardware:	46
2.2.3	Estimación del Proyecto de Software.	47
2.2.3.1	Estimación por Modelo COCOMO II.	48
2.2.4	Análisis y Gestión del Riesgo.	50
2.2.4.1	Identificación del Riesgo.	51
2.2.4.2	Proyección del Riesgo.	52
2.2.4.3	Reducción, supervisión y gestión del riesgo.	55
2.2.5	Planificación Temporal.	59
2.2.5.1	Planificación temporal por la Técnica Pert.	62
2.2.5.1.1	Representar un grafo de Pert.	64
2.2.5.1.2	Identificar el camino crítico y la holgura de las actividades.	71
2.2.5.1.3	Calendario de ejecución del proyecto.	72
2.3	Control Sistemático.	74

2.4	Conclusiones del capítulo.	74
Capítulo 3: Resultados.		75
3.1	Introducción.	75
3.2	Ámbito del software.	75
3.3	Recursos.....	75
3.4	Estimación del proyecto de software.	76
3.5	Análisis y Gestión del Riesgo.....	77
3.6	Planificación y seguimiento del proyecto.	77
3.7	Conclusiones del capítulo.	78
Conclusiones generales.		79
Recomendaciones.		80
Referencia Bibliográfica.		81
Bibliografía Consultada.....		83
Anexos.		84

Introducción.

La planificación brinda la posibilidad de tener un control, un dominio de las tareas básicas para el desarrollo de un proyecto. Mediante la misma se definen los niveles de producción, el tiempo que tendrá cada tarea, los recursos que se utilizarán, define, en fin, el desarrollo que va a tener el proyecto en un cierto tiempo cuando los objetivos que se tienen son claros.

La planificación existe desde que el hombre se convirtió en un ser racional, aún cuando todavía no se comunicaba sino a través de pinturas en las paredes de las cavernas, desde que pensó en cómo obtener alimentos para el día y conservarlos para los otros, en que las inclemencias del tiempo no le permitían salir a cazar o pescar; desde que pensó en cómo defenderse y proteger su vida, ya el hombre planeaba como ganar un combate, cómo sobrevivir, aún cuando conscientemente no supiera que eso se llamaba planear.

Según se ha desarrollado la humanidad, se ha ido perfeccionando y desarrollando la planificación y con ella los métodos utilizados para la elaboración de planes con objetivos futuros.

El surgimiento de la Informática y la digitalización de todas las empresas e instituciones han sido un avance en la ciencia y la técnica. Este gran avance ha posibilitado la construcción de software, los cuales hacen menos complejo el trabajo y se ahorra tiempo. Y como es lógico, la planificación no se ha quedado atrás, ha evolucionado de igual forma.

Cuba también se ha sumergido en todo este mundo del avance tecnológico, ya que se ha propuesto informatizar toda la sociedad empresarial con la construcción de diferentes software, cada uno con un propósito en específico, para facilitar el trabajo del hombre actual, y además, poder insertarse en el mercado mundial y así, desarrollar la economía cubana.

Uno de estos avances en el mundo de la informática en Cuba, es la creación de una nueva universidad para los estudios de esta ciencia, la Universidad de las Ciencias Informática (UCI), donde sus estudiantes están incorporados a proyectos productivos dedicados a la construcción de software, cada uno con una finalidad específica exigida por el cliente, ya sean nacionales o extranjeros.

Uno de estos proyectos es la Informatización del Convenio Cuba-Venezuela, mediante el cual se pretende informatizar la gestión de los convenios que se firman en el marco de las relaciones bilaterales entre ambos países, además se tiene interés que el sistema le imprima una elevada efectividad en el seguimiento y control de estos proyectos contratados en el marco de la mixta correspondiente. En

correspondencia con la visión que se prevé tenga la aplicación, el proyecto transcurre hoy por la fase de Inicio, según la Metodología Rational Unification Process (RUP), etapa que requiere de una organización, planificación y control extrema. Además esta es una etapa que organiza y disciplina el sistema de trabajo que se desempeñará durante el proceso productivo, el cual debe afrontarse de manera adecuada para que al final de todo el proceso, se puedan obtener resultados exitosos.

A partir de todo lo planteado anteriormente se asume como **Situación Problemática** que:

En los cinco años de existencia de la UCI no se ha podido estandarizar, ni fomentar una estrategia adecuada para la planificación de un proyecto de software que se haga extensible a todos los proyectos de la Universidad. Los estudiantes y profesores que asumen el rol de Planificador, en ocasiones no tienen un amplio dominio de las técnicas y herramientas para la planificación y seguimiento de proyectos de esta envergadura. Esto implica ineffectividad en la planeación del proyecto, incumplimiento de lo pactado con el cliente, baja productividad y también obstruye el proceso docente y hacen que el proceso de desarrollo del software se haga sin la calidad requerida. El proyecto Informatización del Convenio Cuba-Venezuela es víctima de tales dificultades. Este proyecto se encuentra actualmente en los inicios de su desarrollo. Dada la envergadura y el alcance que se prevé tenga dicha aplicación, se cuenta con un numeroso equipo de desarrolladores. En este tipo de equipos multidisciplinarios es esencial mantener un estricto control de las tareas y recursos tanto humanos como tecnológicos. Actualmente se cuenta con un responsable de organizar y planificar los procesos que dirigen el desarrollo de la aplicación, pero el trabajo que se hace es insuficiente debido a que no se tiene definida una línea, un mecanismo de trabajo por la cual guiarse para obtener una planificación eficiente como se requiere.

Teniendo en cuenta toda la situación problemática del proyecto se plantea como **Problema**:

No existe un proceso estandarizado para una planificación adecuada en el proyecto Informatización del Convenio Cuba - Venezuela, esto influye en el incumplimiento con el Plan de Desarrollo del mismo.

Teniendo en cuenta el problema planteado, se asume como **Objeto de Investigación**:

Gestión de Proyectos de Producción de Software.

En correspondencia con el objeto abordado se tomó como **Campo de Acción**:

Planificación y seguimiento del Proyecto Informatización del Convenio Cuba - Venezuela.

En correspondencia con problemas confrontados la siguiente investigación se propone como **Objetivo general**:

Establecer una Estrategia de Planificación que garantice el estricto cumplimiento del Plan de Desarrollo del Proyecto Informatización del Convenio Cuba - Venezuela, generando una adecuada organización y gestión de los Recursos Humanos y Tecnológicos.

En correspondencia con los elementos de investigación antes expuestos y con el objetivo general anteriormente definido, se desglosan los siguientes **Objetivos específicos**:

- Definir la organización de la planificación.
- Definir los mecanismos de captura de información para la planificación, y control sistemático.
- Definir métodos de estimación de tiempos y recursos.
- Definir los riesgos que puedan impactar en la planificación del proyecto.
- Definir las herramientas para implementar la estrategia de planificación.

En la presente investigación se pretende validar la siguiente **Hipótesis**:

La definición y aplicación de una estrategia de planificación y control sistemático en el proyecto Informatización del Convenio Cuba - Venezuela logrará una eficiencia en el proceso productivo y la satisfacción del cliente.

Marco Conceptual:

Un **producto** es algo que produces para un compañero, un empresario o un cliente.

Un **proyecto** normalmente produce un producto.

Una **tarea** se define como un elemento de trabajo.

Un **proceso** se define como la forma de hacer proyectos.

Los **planes** describen la forma en que un proyecto concreto va a ser hecho: cómo, cuándo y qué coste tendrá. También se pueden planificar tareas individuales.

Un **trabajo** es algo que haces, tanto un proyecto como una tarea.

Variables Dependientes:

Eficiencia en el proceso productivo y la satisfacción del cliente en el Proyecto Informatización del Convenio Cuba - Venezuela.

Variables Independientes:

Estrategia de planificación y control sistemático en el proyecto Informatización del Convenio Cuba - Venezuela.

Para dar cumplimiento a dicha investigación se trazan una serie de actividades cronológicas las cuales se definen como **Tareas para la investigación:**

- Hacer un estudio del estado del arte de las estrategias de planificación y control a utilizar en proyectos productivos.
- Realizar un estudio de los recursos que se disponen en el proyecto.
- Hacer un estudio de las herramientas utilizadas para la estimación y gestión de riesgos en proyectos productivos.
- Determinar cuales serían las herramientas idóneas para poner en práctica en el proyecto para la gestión de riesgos y la estimación.
- Estudiar las herramientas para la implementación de las estrategias de planificación y control que existen.
- Definir una línea a seguir para realizar una adecuada planificación y control sistemático del proyecto Informatización del Convenio Cuba – Venezuela.

Según los objetivos que se han propuesto se espera llegar a alcanzar como **Posible Resultado:**

- Disponer de un mecanismo que permita organizar y controlar en cualquier momento el estado real del desarrollo del sistema.
- Hacer un uso eficiente de los recursos que se dispone.
- Delimitar las responsabilidades y tareas de cada miembro del equipo.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción

En este primer capítulo se empezará advirtiendo sobre la importancia de seguir una adecuada planificación para la construcción de un proyecto de software y se evaluarán las principales características, ventajas y desventajas. Se expresarán determinados conceptos con los que ha sido denominada la planificación, por algunos de los más prestigiosos autores del tema.

Se describirán los pasos que se deben seguir para la planificación, así como las herramientas y técnicas actuales para la ejecución de cada una de estas actividades.

Por ultimo se hace un análisis entre las principales herramientas que existen para la realización de la planificación en un proyecto; se investigará en proyectos de la Universidad de las Ciencias Informáticas, cuales son sus métodos de planificación y con todo este conglomerado de información se evaluarían las posibles opciones en cuanto a sus potencialidades y se tomaría la decisión de adoptar una herramienta que se estime va a cumplir con los requerimientos, para establecer una estrategia de planificación que garantice el estricto cumplimiento del Plan de Desarrollo del Proyecto Informatización del Convenio Cuba - Venezuela, generando una adecuada organización y gestión de los Recursos Humanos y Tecnológicos.

1.2 Evolución de la planificación

La gestión de un proyecto de software comienza con la planificación de las actividades involucradas en este proceso. La planificación es un proceso de evaluación sistemática de un negocio, definiendo los objetivos, identificando metas y desarrollando estrategias para alcanzar dichos objetivos y localizando recursos para llevar a cabo dichas estrategias.

Todo proyecto de ingeniería de software debe partir con un buen plan de estimación de costos y plazos realistas, para que se obtenga un producto con calidad que responda a las necesidades del cliente.

Se pueden encontrar muchas definiciones de planificación según el contexto y el punto de vista del autor, pero todos de forma general la destacan como una actividad que debe estar presente en todos los procesos de construcción de software.

Se realizó una búsqueda de las definiciones y puntos de vista de algunos autores, de acuerdo al mismo se puede plantear lo siguiente:

"Es el proceso consciente de selección y desarrollo del mejor curso de acción para lograr el objetivo." [Jiménez, 1982]

"La planificación es un proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos" [Jiménez, 1982].

"Es el proceso de seleccionar información y hacer suposiciones respecto al futuro para formular las actividades necesarias para realizar los objetivos organizacionales" [Terry, 1987].

"Consiste en decidir con anticipación lo que hay que hacer, quién tiene que hacerlo, y cómo deberá hacerse" [Murdick, 1994].

"Es el proceso de establecer metas y elegir medios para alcanzar dichas metas" [Stoner, 1996].

"Es el proceso de definir el curso de acción y los procedimientos requeridos para alcanzar los objetivos y metas. El plan establece lo que hay que hacer para llegar al estado final deseado" [Cortés, 1998].

Dada las diferentes definiciones de planificación de algunos autores relacionados con el tema, se llega a la conclusión y elaboración de la definición de planificación.

De acuerdo a lo esbozado anteriormente se define que la planificación es el proceso de anticipar qué debe hacerse, quién debe hacerlo y cómo debe hacerse, de acuerdo con los recursos que se disponen y teniendo en cuenta los factores internos y externos que puedan influir, para lograr las metas u objetivos trazados en un tiempo apropiado.

El objetivo del proyecto es siempre triple. No basta con conseguir uno o dos objetivos, ni hay que dar más importancia a uno o a otro. [GETEC, 2007]



Figura 1.1: Objetivo triple de un proyecto.

El primer objetivo es el resultado final de proyecto, es decir, la obra que se quiere realizar y que supone el origen y justificación del proyecto, por lo que puede considerarse el objetivo más importante y significativo.

En el caso de proyectos externos, el objetivo de coste suele estar definido y tiene una importancia grande. Normalmente existe un contrato, y el proveedor deberá respetarlo. En proyectos internos es frecuente que el objetivo de coste no figure en forma explícita, algo que se debe intentar reducir.

El plazo es el objetivo que más fácilmente se deteriora, convirtiéndose así en el que mejor mide el grado de calidad de gestión del proyecto. A menudo se piensa que el plazo de realización de un proyecto no debe valorarse excesivamente, puesto que es algo que "casi nunca se respeta". Pero hay proyectos en los que este objetivo se convierte en el más importante. ¿Qué pasaría si las obras del estadio olímpico no estuvieran terminadas para la inauguración de los Juegos Olímpicos?

El aspecto triangular de los objetivos se refuerza por la necesidad de coherencia y proporción entre los mismos. Los tres son inseparables y forman un sistema en el que cada modificación de cada una de las partes afecta a las restantes. Dado que la maximización individual de los tres criterios básicos no es posible, es necesario maximizar una cierta combinación entre ellos, priorizando aquellos que se adapten mejor a las estrategias de la empresa.

Después de dar a conocer los conceptos y objetivos de la planificación vale la pena caracterizar esta actividad organizativa para su mejor entendimiento.

1.2.1 Las características más importantes de la planificación.

A continuación se muestran algunas de las principales características de la planificación, así como sus ventajas y desventajas. [Costa, 2007]

- La planificación es un proceso permanente y continuo: no se agota en ningún plan de acción, sino que se realiza continuamente en la empresa.
- La planificación está siempre orientada hacia el futuro: la planificación está ligada a la previsión.
- La planificación busca la racionalidad en la toma de decisiones: al establecer esquemas para el futuro, la planificación funciona como un medio orientador del proceso decisorio, que le da mayor racionalidad y disminuye la incertidumbre inherente en cualquier toma de decisión.
- La planificación busca seleccionar un curso de acción entre varias alternativas: la planificación constituye un curso de acción escogido entre varias alternativas de caminos potenciales.
- La planificación es sistemática: la planificación debe tener en cuenta el sistema y subsistemas que lo conforman; debe abarcar la organización como totalidad.
- La planificación es repetitiva: incluye pasos o fases que se suceden. Es un proceso que forma parte de otro mayor: el proceso administrativo.
- La planificación es una técnica de asignación de recursos: tiene por fin la definición, el dimensionamiento y la asignación de los recursos humanos y no humanos en la empresa, según se haya estudiado y decidido con anterioridad.
- La planificación es una técnica cíclica: la planificación se convierte en realidad a medida que se ejecuta. A medida que va ejecutándose, la planificación permite condiciones de evaluación y medición para establecer una nueva planificación con información y perspectivas más seguras y correctas.
- La planificación es una función administrativa que interactúa con las demás; está estrechamente ligada a las demás funciones – organización, dirección y control – sobre las que influye y de las que recibe influencia en todo momento y en todos los niveles de la organización.
- La planificación es una técnica de coordinación e integración: permite la coordinación e integración de varias actividades para conseguir los objetivos previstos.
- La planificación es una técnica de cambio e innovación: constituye una de las mejores maneras deliberadas de introducir cambios e innovaciones en una empresa, definidos y seleccionados con anticipación y debidamente programados para el futuro.

Según estos autores la planificación es una actividad de gran importancia para la dirección, seguimiento y control de un proyecto de producción, lo que le otorga ventajas que deben estimular a los líderes de cualquier proyecto en cualquier nivel de organización.

1.2.2 Ventajas de la planificación.

Entre estas **ventajas** se encuentran: [Costa, 2007]

- Contribuye a actividades ordenadas y con un propósito. Todos los esfuerzos están apuntados hacia los resultados deseados y se logra una secuencia efectiva de tales esfuerzos. El trabajo no productivo se minimiza.
- Señala la necesidad de cambios futuros. Ayuda a visualizar las amenazas y oportunidades que se pueden presentar y evaluar nuevos campos para una posible participación en ellos. Lo anterior permite evitar la entropía, es decir, la tendencia de que las cosas sigan su curso hacia la destrucción.
- Proporciona una base para el control. La planificación y el control son inseparables, ya que son como los gemelos de la administración. Cualquier intento de controlar sin planes carece de sentido, puesto que no hay forma que las personas sepan si van en la dirección correcta, a no ser que primero tengan en claro a donde ir. Así, los planes proporcionan los estándares de control.
- Obliga a la visualización de un todo. Se obtiene una identificación constructiva con los distintos problemas y las diversas potencialidades de la organización en general. Esta forma de abarcar todo es valiosa, pues capacita al gerente a ver relaciones de importancia, a obtener una comprensión más plena de cada actividad y a apreciar las bases sobre las cuales están apoyadas sus acciones administrativas.
- Dirige la atención hacia los objetivos. Ayuda a tener siempre presente, por parte de todos los componentes de la organización, los objetivos de esta y la adecuación de ellos al medio, cuando es necesario.

1.2.3 Desventajas de la planificación.

Si bien la planificación es una función fundamental de la administración, presenta **desventajas** o limitaciones en su uso, entre las cuales se pueden destacar las siguientes: [Costa, 2007]

- Esta limitada por la poca precisión de la información y por la incertidumbre de los hechos futuros. Es imposible predecir con exactitud lo que va a ocurrir en el futuro, de ahí que sea fundamental que el administrador trabaje con escenarios alternativos y con márgenes de tolerancia.
- La planificación tiene un alto costo. Dado que es la base de una buena administración, una adecuada planificación requiere un gran esfuerzo que se traduce en tiempo, lo cual lleva a aumentar los costos en términos de horas hombre.
- La planificación puede ahogar la iniciativa. Puede darse la tendencia a realizar una planificación excesiva, lo cual puede llevar a rigideces en la actuación de los administradores y del personal de la organización general.
- Puede demorar en algunos casos las acciones. Existen situaciones que requieren de respuestas casi instantáneas, por lo que si se trata de desarrollar planes, se puede llegar tarde con las acciones correctivas pertinentes.

La planificación de un proyecto debe afrontarse de manera adecuada para que al final del mismo se pueda hablar de éxito. La planificación no se realiza solamente al inicio, o para comenzar un proyecto, durante todo el proceso del mismo debe de hacerse una planificación futura, según vaya avanzando el proyecto se deben ir haciendo nuevas planificaciones y nuevas estimaciones.

1.3 Actividades para la Planificación de un Proyecto de Software.

El objetivo de la planificación de un proyecto de software es proporcionar un marco de trabajo que permita al gestor de planificación hacer estimaciones razonables de recursos, costos y planificación temporal. La planificación de un proyecto de software se elabora basada en una serie de actividades escalonadas. Estas actividades se describen a continuación.

1.3.1 Ámbito del software.

La primera actividad de la planificación del proyecto de software es determinar el ámbito del software (prerrequisito para la estimación). Se deben evaluar la función y el rendimiento que se asignaron al software durante la ingeniería del sistema de computadora. Hasta que no se delimita el ámbito no es posible realizar una estimación con sentido. El ámbito del software describe el control y los datos a procesar, la función, el rendimiento, las restricciones, las interfaces y la fiabilidad. Se evalúan las funciones descritas en la declaración del ámbito, y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. [Pressman, 2005]

1.3.2 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.

La Figura 1.2 ilustra los recursos de desarrollo en forma de pirámide. En la base de la pirámide de recursos se encuentra el entorno de desarrollo, herramientas de hardware y software que proporciona la infraestructura de soporte al esfuerzo de desarrollo. En un nivel más alto se encuentran los componentes de software reutilizables, los bloques de software que pueden reducir drásticamente los costes de desarrollo y acelerar la entrega. En la parte más alta de la pirámide está el recurso primario, el personal. Cada recurso queda especificado mediante cuatro características: descripción del recurso, informe de disponibilidad, fecha cronológica en la que se requiere el recurso, tiempo durante el que será aplicado el recurso. Las dos últimas características pueden verse como una ventana temporal. La disponibilidad del recurso para una ventana específica tiene que establecerse lo más pronto posible. [Pressman, 2005]

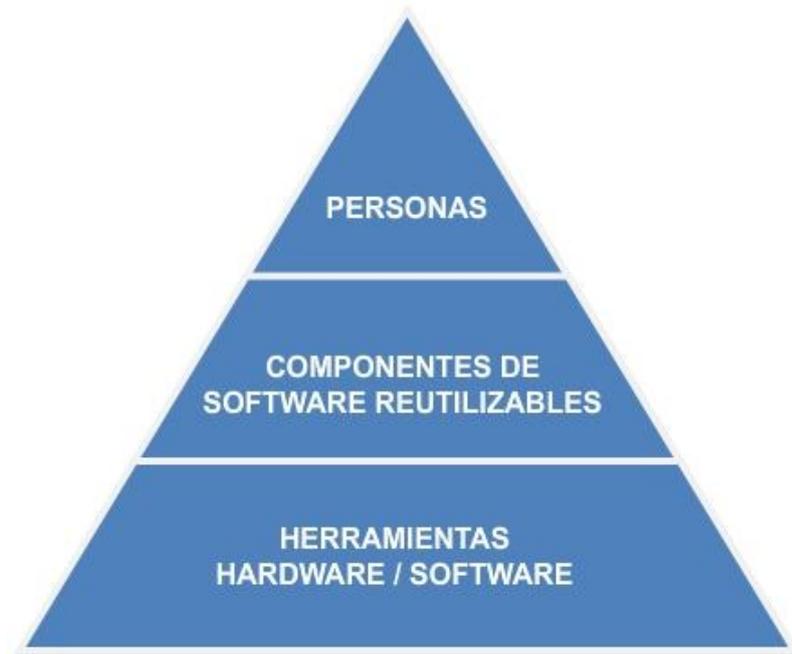


Figura 1.2: Recursos del proyecto.

1.3.2.1 Recursos Humanos.

Para la estimación de **recursos humanos** el encargado de la planificación comienza elevando el ámbito y seleccionando las habilidades que se requieren para llevar a cabo el desarrollo. Hay que especificar tanto la posición dentro de la organización (por ejemplo: gestor, ingeniero de software experimentado,

etc.) como la especialidad (por ejemplo: telecomunicaciones, bases de datos, cliente/servidor). Para proyectos relativamente pequeños (una persona-año o menos) una sola persona puede llevar acabo todos los pasos de ingeniería del software, consultando con especialistas siempre que sea necesario.

El número de personas requerido para un proyecto de software sólo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo, personas-mes). [Pressman, 2005]

Cualquier estudio sobre recursos de software 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.

1.3.2.2 Recursos de Software Reutilizables.

Para los **recursos de software reutilizables** Bennatan sugiere cuatro categorías que se deberían tener en cuenta a medida que se avanza con la planificación: [Bennatan, 1992]

1. Componentes ya desarrollados. Se pueden comprar a un tercero o haber sido desarrollados para un proyecto anterior. Están validados totalmente y listos para usarse en este proyecto.
2. Componentes ya experimentados. Existen componentes parecidos a los que se requieren para el proyecto actual. El equipo tiene experiencia en el área de aplicación y los cambios son mínimos.
3. Componentes con experiencia parcial. Existen componentes que se relacionan con los que se requieren para el proyecto actual. El equipo sólo tiene experiencia en el área de aplicación anterior y los cambios serán sustanciales.
4. Componentes nuevos. Los componentes deben construirse específicamente para este proyecto.

Si los componentes ya desarrollados cumplen con los requisitos del proyecto, se recomienda que sean comprados. El costo de compra y de integración en la mayoría de los casos será siempre menor que el costo de desarrollarlos. Además el riesgo es bajo.

Si se dispone de componentes ya experimentados, los riesgos asociados a la modificación e integración generalmente se aceptan.

Si se dispone de componentes con experiencia parcial, hay que analizar su uso con cuidado. Puede ser

que el costo de modificarlos e integrarlos sea mayor que el de desarrollar componentes nuevos.

1.3.2.3 Recursos de entorno (hardware/software).

Los **recursos de entorno** (hardware/software) o entorno de ingeniería del software (EIS) es donde se apoya el proyecto de software, incorpora hardware y software. El hardware proporciona una plataforma con las herramientas (software) requeridas para producir los productos que son el resultado de una buena práctica de la ingeniería del software. Un planificador de proyecto debe determinar la ventana temporal requerida para el hardware y el software, y verificar que estos recursos estarán disponibles. [Pressman, 2005]

1.3.3 Técnicas para la estimación de proyectos de software.

Siempre que se desee consumir un proyecto de software óptimamente, es preciso realizar una buena planificación del mismo, con el propósito de prever el mayor número de dificultades posibles. Es necesario obtener una estimación de los recursos, de la duración y del costo del mismo. Esto ayuda al equipo de trabajo a organizar su tiempo y sus tareas en objetivos concretos.

Para realizar estimaciones de costos y esfuerzos se tienen varias opciones: [Pressman, 2005]

- Dejar la estimación para más adelante (obviamente, se puede realizar una estimación al cien por cien fiable tras haber terminado el proyecto).
- Esta opción, aunque atractiva, no es práctica. Las estimaciones de costes han de ser proporcionadas a priori. Sin embargo, hay que reconocer que cuanto más tiempo se espere, más cosas se sabrán del proyecto, y cuanto más se sepa, menor será la probabilidad de cometer serios errores en las estimaciones.
- Basar las estimaciones en proyectos similares ya terminados.
- Es razonable si el cliente, condiciones de administración, el medio ambiente, los requisitos, las fechas límites, son similares a proyectos anteriores.
- A pesar de eso, la experiencia anterior no ha sido siempre un buen indicador de resultados futuros.
- Utilizar técnicas de descomposición del problema.
- Utilizan un enfoque de divide y vencerás.

- Descomponen el proyecto en sus funciones principales y la estimación del costo y esfuerzo puede realizarse en base a métricas históricas de manera más fiable.

- Desarrollar un modelo empírico de cálculo de costos y esfuerzos.

- Se basan en datos históricos y son de la forma $d = f(v_i)$ donde d es el valor estimado (esfuerzo, costo, duración del proyecto) y los v_i son algunos parámetros independientes (Líneas de Código ó Puntos de Función).

Para esta faena existen una gama de herramientas dedicadas a realizar estimaciones para el desarrollo del software. Algunas de ellas se mencionan y explican en el trabajo.

1.3.3.1 Modelo de estimación por Puntos de Función.

El modelo de Puntos de Función (PF), fue definido en 1979 por Allan Albrecht, de la gran compañía IBM, dedicada a la fabricación y comercialización de hardware, software y servicios relacionados con la informática.

El análisis por puntos de función es un método para cuantificar el tamaño y la complejidad de un sistema de software desde la perspectiva del usuario que este desarrolla o desarrollará. Es una técnica totalmente independiente del lenguaje o herramienta utilizada en el desarrollo del proyecto.

Es un método que permite medir el tamaño del software, el coste y el tiempo de desarrollo. Se aclara que está diseñado para medir aplicaciones de negocios, y no para medir aplicaciones técnicas o científicas. Esas aplicaciones generalmente median con algoritmos complejos que el método de puntos de función no está diseñado para manejar. Evalúa fácilmente la calidad y productividad de los trabajadores, el esfuerzo de adaptación, modificación y mantenimiento; además de la posibilidad de desarrollo propio y el beneficio que se pueda obtener del trabajo ejercido.

Entre las principales características del modelo de Puntos de Función se encuentran las siguientes: [Graham, 1990]

- Los puntos de función son independientes del lenguaje, herramientas o metodologías utilizadas en la implementación; por ejemplo, no tienen que considerar lenguajes de programación, sistemas de administración de bases de datos, hardware, o cualquier otra tecnología de procesamiento de datos.

- Los puntos de función pueden ser estimados a partir de la especificación de requisitos o especificaciones de diseño, haciendo posible de este modo la estimación del esfuerzo de desarrollo en etapas tempranas del mismo.

Como los puntos de función están íntimamente relacionados con la declaración de requisitos, cualquier modificación a ésta, puede ser reflejada sin mayor dificultad en una reestimación.

- Los puntos de función están basados en una visión externa del usuario del sistema, los usuarios no técnicos del software poseen un mejor entendimiento de lo que los puntos de función están midiendo. El método resuelve muchas de las inconsistencias que aparecen cuando se utiliza líneas de código como métrica del tamaño del software. [Marcela, 2008]

En resumen, los puntos de función aparecen con ventajas substanciales por sobre las líneas de código, para fines de estimación temprana del tamaño del software, y por ende, del esfuerzo de desarrollo. Además es una medida ampliamente utilizada, y con éxito, en muchas organizaciones que desarrollan software en forma masiva.

1.3.3.2 Modelo de estimación por Casos de Uso.

La estimación por casos de uso es un modelo de estimación temprana de cálculo del esfuerzo de desarrollo previsto para la ejecución de un proyecto de software; cuyos principios derivan en cierto modo de la técnica de puntos de función.

Esta estimación clasifica los casos de uso del sistema de software como: simples, medios y complejos y les asigna factores de peso de 5, 10 y 15 respectivamente. Este análisis puede hacerse basado en las transacciones (tomando como transacción cada evento que ocurre entre el actor y el sistema) o en las clases. [Navegapolis, 2008]

Basado en las transacciones serían, de 3 o menos transacciones - FP 5 – tipo simple, de 4 a 7 transacciones – FP 10 – tipo medio y más de 7 transacciones – FP 15 – tipo complejo.

Basado en clases serían, menos de 5 clases – FP 5 – tipo simple, de 5 a 10 clases – FP 10 – tipo medio, más de 10 clases – FP 15 – tipo complejo.

Tipo	Por Transacciones	Por Clases	Factor de Peso
simple	menos de 3	menos de 5	5
medio	4 a 7	5 a 10	10
complejo	más de 7	más de 10	15

Tabla 1.1: Estimación de casos de uso.

1.3.3.3 Puntos de Característica.

Este método considera los mismos elementos que considera Albrecht en su análisis por puntos de función, sólo que añade la variable "número de algoritmos" y elimina los niveles de complejidad, así, cada cuenta es pesada por un valor único para ese componente (es decir, se le asigna complejidad media). [Allan, 1983]

1.3.3.4 Líneas de código.

La métrica de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido Lines Of Code (LOC) o Source Lines Of Code (SLOC). Se han propuesto varios modelos de estimación, la mayoría de ellos son funciones de las líneas de código o de las miles de líneas de código que tendrá el software a desarrollar. Generalmente, el modelo de estimación de esfuerzo consiste de dos partes. La primera provee una base de estimación como una función del tamaño del software, y es de la siguiente forma: [Marcela, 2008]

$$E = A + B * (KLOC)^C$$

Donde E es el esfuerzo estimado en meses hombre, A , B y C son constantes y KLOC es el tamaño estimado del sistema final en miles de líneas de código. La segunda parte del modelo modifica esta estimación en base a cuantificar la influencia de factores de ambiente, por ejemplo la utilización de diferentes metodologías, habilidad del equipo de desarrollo y restricciones de hardware. [Marcela, 2008]

La definición de KLOC es importante si se quiere comparar los distintos modelos que se han propuesto en la literatura. Algunos de ellos incluyen líneas de comentarios, y otros no. Del mismo modo, la definición del esfuerzo estimado E es también importante, ya que E puede representar sólo el esfuerzo de codificación, o en el otro extremo, el esfuerzo total del análisis, diseño, codificación, test y mantenimiento. Por estas razones, comparar estos modelos se torna complejo. [Marcela, 2008]

Los principales problemas de utilizar líneas de código como métrica para estimación del esfuerzo son la falta de una definición universal de línea de código, su dependencia con el lenguaje de desarrollo y la

dificultad de estimar en fases tempranas del desarrollo la cantidad de líneas que tendrá una aplicación. [Marcela, 2008]

1.3.3.5 Modelo de estimación por COCOMO II.

El Modelo Constructivo de Costo o en inglés Constructive Cost Model (COCOMO) se publicó por primera vez en 1981 por Barry Boehm y reflejaba las prácticas en desarrollo de software de aquel momento.

Un método de estimación de coste de desarrollo no es otra cosa que establecer una relación matemática entre el esfuerzo y el tiempo requerido para desarrollar un producto / proyecto. El COCOMO es un modelo empírico obtenido de la revisión de diversos proyectos informáticos de todos los tamaños especialmente grandes. Se utiliza porque está bien documentado, es de dominio público y su uso está extendido.

COCOMO proporciona una familia de estimación de costos de software y tiene en cuenta las necesidades de cada sector y el tipo de información disponible para sostener la estimación. Presenta tres submodelos: [Boehm, 1981]

- 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.
- 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.
- 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 del proceso de ingeniería de software.

Se ajusta para tres diferentes tipos de proyectos de software: [Boehm, 1981]

- Orgánico: proyectos de software sencillos, en los que trabajan grupos de programadores experimentados.
- Semiacoplado: proyectos de software de tamaño y complejidad intermedia, en los que equipos con variados niveles de experiencia deben satisfacer requisitos poco o medio rígidos.
- Empotrado: proyectos muy complejos, en los que se carece de experiencia y se engloban en un

entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

En la década y media siguiente las técnicas de desarrollo software cambiaron drásticamente, la aplicación del modelo COCOMO original empezaba a resultar problemática. La solución al problema era reinventar el modelo para aplicarlo a los 90. Después de muchos años de esfuerzo combinado entre USC-CSE¹, IRUS y UC Irvine²² y las Organizaciones Afiliadas al Proyecto COCOMO II, el resultado es COCOMO II, un modelo de estimación de coste que refleja los cambios en la práctica de desarrollo de software profesional que ha surgido a partir de los años 70. Este nuevo y mejorado COCOMO resultó de gran ayuda para los estimadores profesionales de Coste de software. [Moreno, 1998]

Por tanto, COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. COCOMO II apunta hacia los proyectos software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años. [Moreno, 1998]

Para apoyar a los distintos sectores del mercado software, COCOMO II proporciona una familia de modelos de estimación de coste software cada vez más detallado y tiene en cuenta las necesidades de cada sector y el tipo de información disponible para sostener la estimación del coste software. Esta familia de modelos está compuesta por tres submodelos cada uno de los cuales ofrece mayor fidelidad a medida que uno avanza en la planificación del proyecto y en el proceso de diseño. Estos tres submodelos se denominan: [Moreno, 1998]

- El modelo de Composición de Aplicaciones: Indicado para proyectos construidos con herramientas modernas de construcción de interfaces gráficos para usuario.
- El modelo de Diseño anticipado: Este modelo puede utilizarse para obtener estimaciones aproximadas del coste de un proyecto antes de que esté determinada por completo su arquitectura. Utiliza un pequeño conjunto de drivers de coste nuevo y nuevas ecuaciones de estimación. Está basado en Punto de Función sin ajustar o KSLOC (Miles de Líneas de Código Fuente).
- El modelo Post-Arquitectura: Este es el modelo COCOMO II más detallado. Se utiliza una vez que se ha desarrollado por completo la arquitectura del proyecto. Tiene nuevos

drivers de coste, nuevas reglas para el recuento de líneas y nuevas ecuaciones.

COCOMO II es un método muy utilizado mundialmente. Es una jerarquía de modelos de estimación que tratan áreas diferentes. Esta característica permite que se realicen las estimaciones a medida que se van obteniendo más informaciones acerca del producto, ya que, todos los modelos afrontan el cálculo del tamaño y el costo de forma semejante, solo que se diferencian en el momento en que se aplican. Si un proyecto se acaba de concebir, y del cual no se tienen datos, COCOMO II permite obtener una idea del esfuerzo necesario para llevar adelante el mismo, y podrá ser refinada a medida que se obtenga más información. Con los otros modelos de COCOMO II, el modelo de Fase de diseño previo, y el modelo de Fase posterior a la arquitectura, el planificador podrá hacer las estimaciones pertinentes sin comenzar desde el principio nuevamente, considerando nuevas visiones y limitantes, o sea que, ya estará en condiciones, a medida que se adelante el trabajo, de tener apreciaciones a cerca del esfuerzo y el tiempo más relevantes y reales.

COCOMO II reconoce que es difícil estimar en base a LOC en etapas tempranas por lo se basa en los Puntos Objeto (PO).

Los PO son el recuento de pantallas, informes y módulos de Lenguajes de 3^{ra} Generación (3GL) desarrollados en la aplicación, cada uno ponderado mediante un factor de complejidad de tres niveles (simple, medio y complejo).

1.3.3.6 Comparación entre las técnicas de estimación.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Ésta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas. La desventaja es que está diseñado para medir aplicaciones de negocios, y no para medir aplicaciones técnicas o científicas.

Debido a que el análisis por Puntos de Función fue diseñado para software de negocios y no es fácil de generalizar a aplicaciones científicas, de tiempo real y otras, Caper Jones propuso ampliaciones a este método, generando una métrica que denominó Puntos de Característica. Esta da cabida a aplicaciones cuya complejidad algorítmica es alta.

La estimación por casos de uso es un modelo de estimación temprana de cálculo del esfuerzo de desarrollo previsto para la ejecución de un proyecto de software; cuyos principios derivan en cierto modo

de la técnica de puntos de función.

La métrica de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido líneas de código. Los principales problemas de utilizar líneas de código como métrica para estimación del esfuerzo, son la falta de una definición universal de línea de código, su dependencia con el lenguaje de desarrollo y la dificultad de estimar en fases tempranas del desarrollo la cantidad de líneas que tendrá una aplicación.

Mientras que COCOMO II es un método de estimación temprana, basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente.

1.3.4 Análisis y Gestión del Riesgo.

Una actividad que resulta muy importante en el desarrollo de un proyecto de software es el análisis y la gestión de los riesgos. Estar previamente preparado ante la aparición de un posible contratiempo resultará muy decisivo en el futuro desempeño del proyecto de software.

El análisis y la gestión del riesgo son una serie de pasos que ayudan al equipo del software a comprender y a gestionar la incertidumbre. Un proyecto de software puede estar lleno de problemas. Un riesgo es un problema potencial que puede ocurrir o no. Pero sin tener en cuenta el resultado, realmente es una buena idea identificarlo, evaluar su probabilidad de aparición, estimar su impacto, y establecer un plan de contingencia por si ocurre el problema. [Pressman, 2005]

Existen dos tipos de estrategias para el análisis y gestión del riesgo; la reactiva y la proactiva. [Pressman, 2005]

La estrategia reactiva supervisa el proyecto en previsión de posibles riesgos. Los recursos se ponen aparte, en caso de que pudieran convertirse en problemas reales. Pero lo más frecuente es que el equipo de software no haga nada respecto a los riesgos hasta que algo va mal. Después el equipo vuela para corregir el problema rápidamente y si falla, la “gestión de crisis” entra en acción y el proyecto se encuentra en peligro real.

Una estrategia considerablemente más inteligente para el control del riesgo es ser proactivo. La estrategia proactiva empieza mucho antes de que comiencen los trabajos técnicos. Se identifican los riesgos potenciales, se evalúa su probabilidad y su impacto y se establece una prioridad según su importancia. Después, el equipo de Software establece un plan para controlar el riesgo. El primer

objetivo es evitar el riesgo, pero como no se pueden evitar todos los riesgos, el equipo trabaja para desarrollar un plan de contingencia que le permita responder de una manera eficaz y controlada.

1.3.4.1 Riesgo del Software.

Aunque hay diferencia de criterios en cuanto a la definición de riesgo de software, hay acuerdo común en que el riesgo siempre implica dos características [Higuera, 1995]

Incertidumbre: el acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100 por 100 de probabilidad.

Perdida: si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

Cuando se analizan los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdidas asociado con cada riesgo. Para hacerlo, se consideran diferentes categorías de riesgos.

Los riesgos que se pudieran encontrar en la construcción de un software pudieran ser: [Pressman, 2005]

Los riesgos del proyecto amenazan al plan del proyecto; es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costes aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto en un proyecto de software. También la complejidad del proyecto, tamaño y el grado de incertidumbre estructural son factores de riesgo del proyecto.

Los riesgos técnicos amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento. Además, las ambigüedades de especificaciones, incertidumbre técnica, técnicas anticuadas y las tecnologías punta, son también factores de riesgo. Los riesgos técnicos ocurren porque el problema es más difícil de resolver de lo se pensaba.

Los riesgos del negocio amenazan la viabilidad del software a construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o el producto. Los candidatos para los cinco principales riesgos del negocio son: (1) construir un producto o sistema excelente que no quiere nadie en realidad (riesgo de mercado); (2) construir un producto que no encaja en la estrategia comercial general de la compañía (riesgo estratégico); (3) construir un producto que el departamento de ventas no sabe como vender; (4)

perder el apoyo de una gestión experta debido a cambios de enfoque o a cambios de personal (riesgo de dirección); (5) perder presupuesto o personal asignado (riesgos de presupuesto). Es extremadamente importante recalcar que no siempre funciona una categorización tan sencilla. Algunos riesgos son simplemente imposibles de predecir.

Otra categorización general de los riesgos ha sido propuesta por Charette [Charette, 1989]. Los **riesgos conocidos** son todos aquellos que se pueden descubrir después de una cuidadosa evaluación del plan del proyecto, del entorno técnico y comercial en el que se desarrolla el proyecto y otras fuentes de información fiables (por ejemplo: fechas de entrega poco realistas, falta de especificación de requisitos o de ámbito del software, o un entorno pobre de desarrollo). Los **riesgos predecibles** se extrapolan de la experiencia en proyectos anteriores (por ejemplo: cambio de personal, mala comunicación con el cliente, disminución del esfuerzo del personal a medida que atienden peticiones de mantenimiento). Los **riesgos impredecibles** son el comodín de la baraja. Pueden ocurrir, pero son extremadamente difíciles de identificar por adelantado.

1.3.4.2 Identificación del Riesgo.

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc.). Identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario. [Pressman, 2005]

Existen dos tipos diferenciados de riesgos para cada categoría presentada en la Sección 1.3.4.1: riesgos genéricos y riesgos específicos del producto. Los riesgos genéricos son una amenaza potencial para todos los proyectos de software. Los riesgos específicos de producto solo los pueden identificar los que tienen una clara visión de la tecnología, el personal y el entorno específico del proyecto en cuestión. Para identificar los riesgos específicos del producto, se examinan el plan del proyecto y la declaración del ámbito del software y se desarrolla una respuesta a la siguiente pregunta: ¿Qué características especiales de este producto pueden estar amenazadas por nuestro plan del proyecto? [Pressman, 2005]

1.3.4.3 Proyección del Riesgo.

La proyección del riesgo, también denominada estimación del riesgo, intenta medir cada riesgo de dos maneras la probabilidad de que el riesgo sea real y las consecuencias de los problemas asociados con el riesgo, si ocurriera. El jefe del proyecto, junto con otros gestores y personal técnico, realiza cuatro actividades de proyección del riesgo: establecer una escala que refleje la probabilidad percibida del

riesgo; definir las consecuencias del riesgo; estimar el impacto del riesgo en el proyecto y en el producto, y apuntar la exactitud general de la proyección del riesgo de manera que no haya confusiones. [Pressman, 2005]

1.3.4.4 Refinamiento del riesgo.

Durante las primeras etapas de la planificación del proyecto, un riesgo puede ser declarado de un modo muy general. Con el paso del tiempo y con el aprendizaje sobre el proyecto y sobre el riesgo, es posible refinar el riesgo en un conjunto de riesgos más detallados, cada uno algo más fácil de reducir, supervisar y gestionar. [Pressman, 2005]

1.3.4.5 Plan de Reducción, Supervisión y Gestión del Riesgo (Plan RSGR).

Todas las actividades de análisis de riesgo presentadas hasta ahora tienen un solo objetivo, ayudar al equipo del proyecto a desarrollar una estrategia para tratar los riesgos. Una estrategia eficaz debe considerar tres aspectos: [Pressman, 2005]

- Evitar el riesgo.
- Supervisar el riesgo.
- Gestionar el riesgo y planes de contingencia.

Si un equipo de software adopta un enfoque proactivo frente al riesgo, evitarlo es siempre la mejor estrategia. Esto se consigue desarrollando un plan de reducción del riesgo.

Se puede incluir una estrategia de gestión de riesgo en el plan del proyecto de software o se podrían organizar los pasos de gestión del riesgo en un plan diferente de reducción, supervisión y gestión del riesgo. Todos los documentos del Plan RSGR se llevan a cabo como parte del análisis de riesgo y son empleados por el jefe del proyecto como parte del Plan del Proyecto general.

1.3.4.6 Riesgos y peligros para la seguridad.

El riesgo no se limita al proyecto de software solamente. Pueden aparecer riesgos después de haber desarrollado con éxito el software y de haberlo entregado al cliente. Estos riesgos están típicamente asociados con las consecuencias del fallo del software una vez en el mercado. [Pressman, 2005]

Aunque la probabilidad de fallo de un sistema de alta ingeniería es pequeña, un defecto no detectado en

un sistema de control y supervisión basados en computadora podría provocar unas pérdidas económicas enormes, o peor, datos físicos significativos o pérdida de vidas humanas. Pero el coste y beneficios funcionales del control y supervisión basados en computadora a menudo superan al riesgo. Hoy en día, se emplean regularmente hardware y software para el control de sistemas de seguridad crítica. [Pressman, 2005]

La seguridad del software y el análisis del peligro son actividades para garantizar la calidad del software que se centra en la identificación y evaluación de peligros potenciales que pueden impactar al software negativamente y provocar que falle el sistema entero. Si se pueden identificar los peligros al principio del proceso de ingeniería del software, se pueden especificar características de diseño software que eliminen o controlen esos peligros potenciales. [Pressman, 2005]

1.4 Métodos y herramientas utilizadas para la planificación.

Entre las herramientas que se utilizan para la planificación a nivel mundial se encuentran: Diagramas de Gantt, Técnica Pert / CPM, DotProject, Trac, MS Project, GanttProject.

1.4.1 El Diagrama de Gantt:

Es un calendario, en él se puede visualizar el periodo de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo. El instrumento que desarrolló permite también que se siga el curso de cada actividad, al proporcionar información del porcentaje ejecutado de cada una de ellas, así como el grado de adelanto o atraso con respecto al plazo previsto. [Hinojosa, 2003]

El diagrama de Gantt es un diagrama representativo, que permite visualizar fácilmente la distribución temporal del proyecto, pero es poco adecuado para la realización de cálculos.

Ventajas y desventajas del Diagrama de Gantt: [Hinojosa, 2003]

- La ventaja principal del gráfico de Gantt radica en que su trazado requiere un nivel mínimo de planificación, es decir, es necesario que haya un plan que ha de representarse en forma de gráfico.
- Los gráficos de Gantt se revelan muy eficaces en las etapas iniciales de la planificación. Sin embargo, después de iniciada la ejecución de la actividad y cuando comienza a efectuarse modificaciones, el gráfico tiende a volverse confuso.

- El Gráfico de Gantt no ofrece condiciones para el análisis de opciones, ni toma en cuenta factores como el costo.
- No permite, tampoco, la visualización de la relación entre las actividades cuando el número de éstas es grande.

En resumen, para la planificación de actividades relativamente simples, el gráfico de Gantt representa un instrumento de bajo costo y extrema simplicidad en su utilización. Para proyectos complejos, sus limitaciones son bastantes serias.

1.4.2 Técnica PERT y CPM.

En 1957, la Marina de EE.UU. desarrolló el proyecto de los misiles Polaris, un sistema de armamentos basado en submarinos. Lockheed, el contratista principal del proyecto, creó el sistema de programación Técnica de Revisión y Evaluación de Programas o en inglés Program Evaluation and Review Technique (Pert).

Un diagrama Pert (también llamado diagrama de red o de precedencias), está basado principalmente en la evaluación de programa y técnica de revisión, el cual hace referencia a la representación gráfica de las relaciones entre tareas que constituyen el proyecto, y es uno de los métodos utilizados para la planificación de la producción que permiten programar y controlar las actividades con el fin de optimizar el proceso productivo, métodos que introducen la planificación temporal, usados en sistemas de producción complejos, compuestos por diversas actividades ejecutadas en un orden temporal concreto y de las cuales se conoce su duración. Además es muy útil cuando las actividades pueden ser realizadas en paralelo en vez de en secuencia.

Pert trabaja con tiempos probabilísticos debido al nivel de incertidumbre en el tiempo de las actividades de los proyectos para los cuales fue desarrollado, debido a la no existencia de bases de datos para los tiempos de estas actividades.

El diagrama Pert es un grafo de precedencias donde los nudos o nodos son las actividades, mientras que los arcos son las relaciones de precedencia entre actividades. De cada actividad, se debe saber la duración estimada y las actividades que le son precedentes. El resto surge casi automáticamente del mismo diagrama.

Pert emplea tres estimados de tiempo para cada actividad, estos estimados de tiempo son los siguientes: Tiempo Optimista, Tiempo Probable y Tiempo Pesimista. Con estos tres valores se calcula el

tiempo de conclusión o duración esperada y la respectiva varianza, donde el tiempo más probable es el tiempo requerido para completar la actividad bajo condiciones normales. Los tiempos optimistas y pesimistas proporcionan una medida de la incertidumbre inherente en la actividad, incluyendo desperfectos en el equipo, disponibilidad de mano de obra, retardo en los materiales y otros factores.

Muy seguido al surgimiento del Pert, la Compañía DuPont, junto con la División UNIVAC de la Remington Rand, desarrolló el Método del Camino Crítico o en inglés Critical Path Model (CPM) para controlar el mantenimiento de proyectos de plantas químicas de DuPont, buscando el control y la optimización de los costos de operación mediante la planeación adecuada de las actividades componentes estos proyectos.

El CPM es idéntico al Pert en concepto y metodología, por lo que la mayoría de los autores del tema lo tratan como Pert/CPM. La diferencia principal entre ellos es el método por medio del cual se realizan estimados de tiempo para las actividades del proyecto. Con CPM, los tiempos de las actividades son determinísticos y se pueden variar cambiando el nivel de recursos utilizados, mientras que con Pert, los tiempos de las actividades son probabilísticos o estocásticos. A diferencia de Pert, CPM se utiliza en proyectos con poca incertidumbre.

El conjunto de las técnicas Pert y el método del camino crítico proporcionan herramientas cuantitativas que permiten determinar lo siguiente: [Barceló]

- El camino crítico del proyecto, que es la secuencia o cadena de actividades (que no tienen ningún margen) que determina la duración total del proyecto.
- Las estimaciones de tiempos más probables, tanto para la totalidad del proyecto como para el inicio y el final de cada una de las tareas o actividades individuales.
- Los márgenes de tiempo que se dan para cada tarea o actividad individual y que no impliquen un retraso del proyecto.

1.4.3 MS Project:

Microsoft Project es un potente programa de gestión de proyectos que se utiliza y demanda cada vez más por parte de las empresas, para crear planes de proyecto, introducción de datos reales de evolución y realizar un completo seguimiento y control de cada una de sus tareas, así como contabilizar la variación que se produce en el transcurso de un proyecto respecto a lo que inicialmente se había programado. [Adrfornacion, 2007]

Con este programa se gestionan y controlan tanto las tareas que integran un proyecto, como los recursos que se utilizan para su desarrollo, además de las asignaciones recurso-tarea. Además, está comprobado que con el MS Project el administrador o planificador del proyecto posee un control total del proyecto.

Con Microsoft Office Project, se puede dominar con rapidez el proceso de administración de proyectos mediante la Guía de proyectos, la cual consiste en una ayuda interactiva paso a paso que permite configurar proyectos mediante asistentes, administrar tareas y recursos, realizar un seguimiento de los procesos, así como crear informes a partir de la información de los proyectos.

Entre muchas posibilidades de trabajo, MS Project permite:

- Planificar y programar tareas así como asignar recursos a dichas tareas de manera adecuada y sencilla.
- Realizar un control, organización y seguimiento, así como coordinar toda la información que conlleva los requisitos del proyecto, la duración y los recursos asignados a las diferentes tareas.
- Visualizar el Plan de Proyecto en formatos estándar y con un diseño de diagramas muy apropiado y fácil de interpretar.
- Establecer escenarios dentro del proyecto para crear análisis de hipótesis. Planteamientos del tipo "Que pasaría si..."
- Intercambiar información de proyecto con todos sus participantes a través de una red local, Internet o de una Intranet.

Lo más importante del MS Project para la ayuda a la planificación de proyectos es que ofrece la posibilidad de mostrar el Método del Camino Crítico y el diagrama Pert (o red de actividades). Ambas técnicas desarrollan una descripción de la red de tareas del proyecto, es decir, una representación gráfica o tabular de las tareas que deben realizarse desde el principio hasta el final del proyecto. También ofrecen una gestión de recursos completa y un conjunto de diagramas y mecanismos de control que permiten realizar una presentación brillante y muy profesional de la planificación de un proyecto.

Además del diagrama Pert, se suele utilizar un diagrama temporal, el diagrama de Gantt, el cual es un diagrama sencillo pero muy eficiente. Por tal razón, este diagrama es utilizado por la mayoría de las herramientas informatizadas de ayuda a la planificación, para introducir los datos de las diferentes tareas

o actividades que forman la Work Breakdown Structure (WBS) o Estructura Desglosada del Trabajo (EDT) del proyecto.

Las herramientas informatizadas ofrecen también diagramas basados en el calendario para marcar los días de inicio y final de cada tarea, aunque el diagrama de Gantt es suficientemente completo en este aspecto.

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

Existen varias versiones de MS Project como: Microsoft Office Project 2000, Microsoft Office Project 2002, Microsoft Office Project 2003, Microsoft Office Project 2007, entre otras. Las más utilizadas para el desarrollo de la planificación han sido la del 2002 y la del 2003, resaltando que la versión del 2007 es la más mejorada y perfeccionada que ha lanzado Microsoft al mercado.

Cada una de estas versiones trae consigo sus propias versiones como es el caso del Project Standard, Project Professional, Project Server, Project Web Access, Enterprise Project Management, entre otras.

1.4.4 GanttProject:

Es una aplicación de escritorio, desarrollada en Java, por tanto es multiplataforma, libre y gratuito. Permite dividir el proyecto en un árbol de tareas y asignar recursos humanos a cada una de ellas, establecer dependencias entre tareas, de esta forma, una tarea no podrá empezar hasta que esté acabada la anterior. Permite calcular el gráfico Pert, generar informes PDF y HTML, exportar su trabajo a una imagen JPG o PNG, intercambiar datos con MS Project y hojas de cálculo, importar y exportar información de Microsoft Project. [Arquitectura, 2007]

1.4.5 DotProject:

Es una aplicación Web creada en PHP, una herramienta para la administración de proyectos, creada por la comunidad de Software Libre internacional, que incluye módulos para compañías, proyectos, tareas (con gráficas de Gantt), foros, archivos, calendario, contactos, ticket de soporte, etc. Es multi-idioma y soporta distintos niveles de permisos de uso de módulos. [AbartiaTeam, 2007]

DotProject está construido por aplicaciones de código abierto. Utiliza MySQL como base de datos. La plataforma recomendada para utilizar DotProject se denomina LAMP (Linux + Apache + MySQL + PHP).

La aplicación consta de un conjunto de entidades ordenadas jerárquicamente las cuales permiten brindar la funcionalidad del producto.

A continuación se mencionan las entidades más importantes de DotProject: [AbartiaTeam, 2007]

- Compañías: Son las entidades que agrupan proyectos, actividades y usuarios.
- Departamentos: Son áreas dentro de las compañías, que permiten agrupar usuarios en dicho nivel.
- Usuarios/Contactos: DotProject tiene usuarios los cuales son capaces de loguearse a DotProject y trabajar dentro del esquema de permisos que posea el rol de dicho usuario. Los contactos son usuarios especiales que asignados a un determinado proyecto pueden recibir por ejemplo: correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema DotProject. Los usuarios y contactos pertenecen a una compañía.
- Proyectos: Es la entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto.
- Actividades: son las tareas asignadas dentro de un proyecto. Son los componentes sobre los cuales se controla: la duración, dependencias, recursos asignados y progreso. Las actividades deben de pertenecer a un único proyecto.
- Diagramas de Gantt: Permite ver en forma grafica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.
- Tickets: para administrar todos los problemas relacionados a un proyecto.
- Archivos: Permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.
- Foros: Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.

- Administración del Sistema: Contiene la actividades relacionadas a la administración de usuarios, roles y configuración del sistema.
- Recursos: Permite asignar recursos no humanos (oficinas, equipamiento, etc.) a un proyecto.

1.4.6 Trac:

Es un sistema libre sencillo que integra varios componentes con capacidades suficientes para la gestión del día a día de proyectos de desarrollo de software. [Saavedra, 2007]

Es un sistema para definir y visualizar el estado de los hitos de un proyecto (un hito incluye una descripción y una fecha y se usa como atributo de los tickets, que se asocian a hitos concretos).

Es un sistema de seguimiento de eventos en el sistema (histórico de cambios en el wiki, en el sistema de control de versiones, en el sistema de gestión de incidencias o vencimiento de un hito).

Es un sistema de búsqueda que permite localizar páginas del wiki, comentarios dentro de los conjuntos de cambios o tickets en los que aparece una palabra.

Es un visualizador del repositorio Subversión asociado a un proyecto, que permite ver el estado actual del repositorio, los cambios que se han ido produciendo, comparar distintas versiones de ficheros en línea, etc.

Es un sistema de gestión de incidencias o ticketing que permite dar de alta incidencias de distinto tipo y hacer el seguimiento de su evolución (modificación de datos, cambios de estado, inserción de comentarios o ficheros adjuntos, etc.) vía Web o por correo electrónico.

Actualmente el flujo de trabajo con tickets de Trac es fijo, es decir, las reglas de como y cuando se puede pasar de un estado a otro son fijas, independientemente del tipo o características del ticket.

Es importante indicar que Trac únicamente es una interfaz de lectura para el repositorio Subversión, para acceder al Subversión se pueden usar múltiples medios (`https://`, `svn://`, `svn+ssh://`, etc.). Una opción cómoda para poder compartir los usuarios de Trac con el Subversión es montar el acceso a este último con `apache2`, de modo que un mismo servidor proporcione los dos servicios.

Trac es un sistema Web multiplataforma ligero y extensible que incluye las siguientes funcionalidades:

- Un wiki que se puede emplear para documentar cualquier aspecto del proyecto de modo colaborativo y sin necesidad de herramientas especiales (sólo se necesitan los permisos adecuados y un navegador Web).

La sintaxis del wiki se puede emplear en casi todos los módulos del sistema (por ejemplo en los tickets), se permite la definición y empleo de macros (por ejemplo para generar listas de cambios) y la inclusión de texto con sintaxis distintas a la del Wiki como html o rst.

Trac está desarrollado en torno a la idea de un núcleo al que se le pueden añadir plugins que proporcionan distintas funcionalidades (casi todos los componentes estándar son módulos que pueden ser activados, desactivados o reemplazados o modificados por otros.)

Actualmente existen plugins para añadir funcionalidades a Trac como:

- Autenticación con formularios y usuarios en LDAP, BBDD o fichero.
- Visor de documentos Excel.
- Calendario para el histórico.
- Servicios adicionales como blogs, foros, etc.

1.4.7 Open Workbench:

Aplicación de escritorio para Windows con la que se puede planificar proyectos, programarlos, administrar recursos, revisar los proyectos. [OpenWorkbench, 2008]

1.4.8 KPlato:

Aplicación de escritorio para KDE del proyecto KOffice, permite visualizar recursos, cuentas, tareas y mucho más. [The KOffice Project, 2007]

1.4.9 TaskJuggler:

Aplicación de escritorio para Linux que permite administrar tareas, recursos y costes, resuelve conflicto de tareas, informes con todo los datos que necesitas, plantillas desde las que empezar, número ilimitado de escenarios y análisis de riesgos entre otras cosas. [TaskJuggler, 2008]

1.4.10 XPlanner:

XPlanner es un sistema Web, desarrollado en Java, que puede resultar muy útil como herramienta de apoyo a la gestión de equipos de trabajo que emplean prácticas Extreme Programming, o para entornos que combinan prácticas XP con gestión Scrum. [XPlanner, 2008]

1.4.11 OpenProj:

Es un programa de escritorio para la gestión de proyectos: gratuito, open source, con versiones para Linux, Unix, Mac y Windows; compatible con ficheros MS Project y con todas las funcionalidades que ofrece Project (como aplicación de escritorio stand-alone). [Navegapolis, 2007]

1.4.12 Project Dune:

Project Dune es un sistema basado en Web que quiere integrar todos los procesos y documentación del ciclo del desarrollo. De momento se encuentra a medio camino, pero en enero de este año publicó ya la primera versión estable, y con la publicación ayer de la 1.3.0, el sistema empieza a apuntar maneras.

Se distribuye libremente con licencia GPLv2. [Navegapolis, 2007]

1.4.13 Agile Track:

Es una herramienta de interfaz sencilla para planificación y seguimiento de proyectos, Se utiliza específicamente para el desarrollo de software en equipos reducidos con metodologías ágiles, especialmente “Extreme Programming”. Gestiona ciclos de desarrollo basados en iteraciones, con seguimiento de historias de usuario, tareas y bugs.

Es multiplataforma, puede ser utilizado en Windows y Linux sin ningún problema. Consta de dos módulos: el servidor que trabaja con MySQL, y el cliente para el seguimiento de los proyectos. Es un desarrollo Open Source, de uso gratuito con licencia Academic Free License (AFL), el cual usa las gráficas de Gantt para tener una visión del seguimiento del proyecto. [Navegapolis, 2008]

Además, permite ejecutar una demo del cliente completamente operativa desde la Web del proyecto, sobre un navegador (máquina virtual Java).

1.1.14 Comparación entre los métodos y herramientas para la planificación.

Para la planificación temporal se usan una serie de métodos que representan el tiempo que será empleado por cada una de las tareas de ingeniería de software:

En aras de mejorar la planificación temporal del proyecto, se necesita hacer una representación gráfica del mismo de forma que sea posible determinar la duración mínima del proyecto, en este caso del módulo de Presentación. Conocer cuáles son las actividades sobre las que deberá ejercerse un mayor control, y obtener información sobre el estado del proyecto en cada una de sus fases.

Para darle cumplimiento a estas necesidades se realiza una comparación cualitativa entre los métodos, Diagramas de Gantt y la Técnica Pert/CPM con el objetivo de seleccionar el método adecuado para el cumplimiento de las mismas.

Gantt es un diagrama representativo, que permite visualizar fácilmente la distribución temporal del proyecto. Es útil para el control del proyecto, se puede comparar visualmente el plan con los datos reales, y eso permitirá determinar la magnitud de la variación en el diagrama que se tenga en el proyecto. Mientras que la técnica Pert hace referencia a la representación gráfica de las relaciones entre tareas que constituyen el proyecto. Permite al planificador determinar el impacto que tiene sobre el diagrama general del proyecto, los retrasos y las aceleraciones de las tareas o actividades individuales, esta característica le permite al planificador crear estimaciones más realistas del proyecto.

La principal desventaja de los diagramas de Gantt es que, si bien representan las fechas de iniciación y de terminación de las tareas, no muestran las consecuencias generales de las modificaciones del diagrama en cada tarea específica. Es decir, el diagrama de Gantt contempla las tareas como si fueran actividades independientes, no tiene en cuenta que están interrelacionadas. Es poco adecuado para la realización de cálculos.

Actualizar continuamente la red Pert suele ser bastante engorroso, y además las redes no muestran gráficamente las variaciones del diagrama, como lo hacen los diagramas de Gantt; para ver las variaciones, no basta con superponer al original un diagrama PERT actualizado.

Se recomienda Pert para grandes proyectos con alta dependencia entre las tareas: Gantt, por su parte, se recomienda para proyectos más sencillos.

Se concluye que ambos métodos pueden determinar la duración mínima del proyecto. Pero acerca de cuales son las actividades que se le debe ejercer un mayor control, Gantt no aporta ninguna información mientras que Perl a través del camino crítico muestra cuales son las actividades sobre las cuales hay que tener un control, ya que de atrasarse una de ellas implicarían un atraso en el proyecto.

Por estas razones se propone utilizar la Técnica Pert para la planificación temporal.

Comparación entre las herramientas para el seguimiento del proyecto:

A continuación se realiza una comparación entre las herramientas mundialmente más utilizadas, con el objetivo de seleccionar cual es ideal para trabajar en el proyecto Informatización del Convenio Cuba - Venezuela, en aras de lograr una mejor planificación del mismo.

Aspectos a comparar	Herramientas			
	MS Project	GanttProject	DotProject	Trac
Técnicas que utiliza.	Gantt, Pert y CPM.	Gantt	Gantt	Wiki, Navegador de código fuente y Gestor de tickets
Lenguaje en el que está programado	-----	Java	PHP	Python
Entorno en el que corre.	Windows	Windows y Linux	Windows y Linux	Windows y Linux
Alcance	Proyectos simples y complejos	Proyectos simples	Proyectos simples y complejos	Proyectos simples y complejos
Facilidad de uso	Flexible, eficaz y fácil de usar.	Flexible y posee una interfaz ligera.	Flexible, amigable y multiusuario.	Flexible.
¿Herramienta de escritorio?	Si	Si	No	No
¿Es software libre?	No	Si	Si	Si

Tabla 1.2: Comparación entre las herramientas más utilizadas a nivel mundial.

A partir de las características vistas de cada una de las herramientas se determina la utilización de la que más se ajuste a las necesidades del proyecto.

MS Project es una muy buena herramienta de gestión de proyectos, pero corre solamente en Windows y es propietaria; lo que no se ajustaría a las necesidades del proyecto, ni a la estrategia que se ha trazado el país de migración a software libre.

Gantt Project aunque es una herramienta de software libre que permite realizar la planificación de proyectos, no se ajusta a las características del proyecto Informatización del Convenio Cuba – Venezuela, ya que no es muy robusto, con muchas tareas presenta problemas y no es una herramienta

web. No permite que los miembros de un proyecto actualicen a la vez sus avances por lo que no puede ser utilizada de manera colaborativa para monitorizar la evolución del citado proyecto.

DotProject es una herramienta de gestión de proyecto que permite la realización de un plan de proyecto, igualmente dispone de un calendario con la vista de tareas y eventos para facilitar el trabajo colaborativo, un sistema de gestión de incidencias, un repositorio de ficheros y unos foros de discusión. Pero no se incluye ningún tipo de soporte para llevar a cabo la gestión de los riesgos del proyecto, un hito es considerado como una tarea, por lo que tiene su propio avance e incurrido de horas y no es posible realizar modificaciones sobre la visualización del diagrama Gantt. Se trata de una gran herramienta que facilita la planificación y gestión de los proyectos en los que se trabaja de forma colaborativa, ya que su tecnología web hace posible que los miembros del equipo trabajen de forma distribuida, compartiendo la información sobre el servidor central. Permite mantener un control de la actividad de varios departamentos de una organización mediante la descomposición de tareas de cada uno de los proyectos, si bien la tarea de replanificación y movimiento de fechas y dependencias entre proyectos es tediosa comparada con otras herramientas (como MS Project), por lo que el mantenimiento de los planes de proyecto es costoso, siendo éste su principal inconveniente.

Trac es una herramienta open source de gestión de proyectos y seguimiento de tareas, basada en web. Trac permite enlazar información entre la base de datos de las tareas (tickets), las páginas wiki de contenido y el control de revisiones. También sirve como una sofisticada interfaz para el sistema de control de versiones Subversión. Trac dispone de una infinidad de plugins que aumenta cada día gracias a las contribuciones de los usuarios: Diagramas Gantt, Blogs, Soporte para sistema control de versiones, Exportar páginas wiki a PDF, Foros de discusión, Búsqueda en los repositorios, Calendario, entre otros. El desarrollo mediante plugins puede ser la mejor forma de extender la funcionalidad, para realizar el proyecto Informatización del Convenio Cuba – Venezuela sin alterar el proceso de desarrollo del Trac. Además está programado en python y su arquitectura está basada en componentes, lo que es algo que simplifica la creación de plugins a los programadores del proyecto, que anteriormente programaban en este lenguaje, en caso de querer agregarle alguna otra funcionalidad. Tiene el inconveniente de que cada proyecto es independiente y no se comparte la base de datos entre ellos, aunque existe un plugin multiproyecto que palió este defecto. Esta es una ventaja que presenta Trac, la incorporación de nuevas funcionalidades mediante plugin, sin la necesidad de instalar una nueva versión y volver a realizar todo el trabajo. Es una herramienta que es muy usada, por lo que su madurez es excelente. En el proyecto Informatización del Convenio Cuba – Venezuela se tienen previos conocimientos sobre esta

herramienta, por lo que sería una ventaja su utilización, en cambio con la aplicación de otra herramienta, se necesitaría capacitar a los integrantes del mismo, lo que no sería factible por escases de tiempo.

1.5 Métodos y herramientas utilizadas a nivel de Universidad.

Según una entrevista no estructurada que se realizó en 8 proyectos productivos de la facultad 3, la minoría no utiliza una herramienta de gestión de proyecto, para un 12% del total. Las herramientas más utilizadas, son el MS Project, el DotProject y el Trac. Esto demuestra que a pesar de que la gran mayoría emplea herramientas para la planificación, todavía no se le presta la debida importancia. Esto provoca en muchas ocasiones el atraso de los proyectos en la entrega del producto.

En la Universidad de las Ciencias Informáticas además de los métodos y herramientas dichas anteriormente ha surgido una plataforma muy novedosa la cual se debe mencionar por sus características. La Red Nacional de Integración y Desarrollo de Software Libre (RINDE) surgió como punto de encuentro para la producción y mantenimiento de aplicaciones informáticas, y con el fin de estimular la participación de comunidades, grupos de usuarios, organismos, instituciones, universidades, sector productivo y demás personas interesadas en el desarrollo y consolidación de la industria del software libre en la República Bolivariana de Venezuela.

Es una plataforma de intercambio y colaboración de servicios especializados que ofrece una infraestructura básica para las gestiones del conocimiento relativo a las tecnologías de información libres y abiertas.

Cuenta con espacios para el registro de noticias, experiencias, proyectos, foros, cursos, programas y contenidos, así como una herramienta para la gestión de proyectos, organización de grupos de desarrollo, una wiki para el desarrollo de documentación colaborativa y una metodología apoyar el proceso de desarrollo de software de los diversos entes que integran la red nacional.

Objetivos:

- Crear un canal de participación y comunicación para el intercambio de información y divulgación de las soluciones para la migración a software libre: Socialización de Conocimiento.
- Crear un ambiente de producción de software bajo un modelo colaborativo que habilite la integración de usuarios, grupos de trabajo y soporte en software libre.

- Adoptar y adaptar modelos de desarrollo de software que garanticen óptimos niveles de calidad en los procesos de producción y los productos finales.
- Fortalecer a los desarrolladores y usuarios nacionales, a través de actividades de formación, financiamiento de proyectos y diversificación de servicios, entre otros.
- Promover el uso del Software Libre Nacional en las organizaciones públicas y privadas del país.

1.6 Conclusiones del capítulo.

Se ha concluido con este estudio que es imprescindible la realización de una planificación para alcanzar el éxito en todo proyecto de software. Esta contribuye a lograr actividades ordenadas y con un propósito donde todos los esfuerzos están apuntados hacia los resultados deseados. Es por eso que se trazará una línea de trabajo que sea aplicada al proyecto Informatización del Convenio Cuba- Venezuela donde se realice una adecuada estrategia de planificación.

Capítulo 2: Definición y Aplicación de la Estrategia de Planificación.

2.1 Introducción.

En el presente capítulo se define la estrategia de planificación que se utilizará en el Proyecto Informatización del Convenio Cuba – Venezuela. Se definen en cada uno de los pasos para desarrollar la planificación, los métodos y herramientas a utilizar en la estimación, planificación y seguimiento del proyecto. Además de definir los posibles riesgos que puedan afectar el desarrollo del proyecto.

Con la aplicación de la estrategia se definen los recursos personales y tecnológicos necesarios para el desarrollo del proyecto. Se estima el costo, esfuerzo y tiempo del mismo y se analizan y gestionan cada uno de los riesgos encontrados por el equipo de desarrollo.

2.2 Pasos para desarrollar la planificación.

Para desarrollar la planificación del Proyecto Informatización del Convenio Cuba – Venezuela es necesario realizar cada una de las actividades descritas en el Capítulo 1. Cada una de ellas proporciona soluciones importantes que permiten efectuar una planificación adecuada y ejercer un control sistemático sobre las tareas que se deben realizar en un proyecto de software.

El planificador de un proyecto debe estimar tres cosas antes de que comience el proyecto: cuánto durará, cuánto esfuerzo requerirá y cuánta gente estará implicada. Además, el planificador debe predecir los recursos (de hardware y de software) que va a recurrir, y el riesgo implicado. Estas tareas se harán siguiendo una serie de pasos que se describen a continuación.

2.2.1 Establecer el ámbito del software que se va a desarrollar.

El ámbito del software es la primera actividad que debe realizarse antes de comenzar a planificar un proyecto. El ingeniero de software (el analista) es el encargado de realizar esta actividad, la cual se realiza mediante una primera reunión con el cliente.

En esta primera reunión el analista realiza preguntas de contexto libre, enfocadas al cliente, a los objetivos globales y a los beneficios, una serie de preguntas que lleven a un entendimiento básico del problema.

Preguntas tales como: [Charette, 1989]

¿Quién está detrás de este trabajo?

¿Quién utilizará la aplicación?

¿Cuál será el beneficio económico de una buena solución?

¿Hay otro camino para la solución?

El analista debe comprender mejor el problema y hacer que el cliente exprese sus percepciones sobre una solución. Para eso deberá hacer preguntas tales como: [Charette, 1989]

¿Cómo caracteriza usted como cliente un resultado correcto generado con una solución satisfactoria?

¿Con qué problemas se afrontará la solución?

¿Puede mostrarme el entorno en el que se utilizará la aplicación? ¿Hay aspectos o limitaciones especiales de rendimiento que afecten a la forma en que se aborde la solución?

Las últimas preguntas que realiza el analista en este primer encuentro con el cliente, están basadas también en las llamadas meta cuestiones: [Charette, 1989]

¿Es usted la persona apropiada para responder a estas preguntas? ¿Son oficiales sus respuestas?

¿Son relevantes mis preguntas para su problema?

¿Estoy realizando muchas preguntas?

¿Hay alguien más que pueda proporcionar información adicional?

¿Hay algo más que debiera preguntarle?

En las próximas reuniones que se desarrollan, esta sesión de preguntas y respuestas es reemplazada por la combinación de elementos de resolución de problemas, negociación y especificación.

Una vez terminada esta actividad el equipo de desarrollo debe reunirse y preguntarse: ¿El proyecto es viable? ¿Se puede construir el software de acuerdo a ese ámbito? ¿Es factible el proyecto? El estudio del ámbito no es suficiente. Una vez que se ha comprendido el ámbito, tanto el equipo de desarrollo como el resto deben trabajar para determinar si puede ser construido dentro de las dimensiones reflejadas anteriormente.

El ámbito del software describe el control y los datos a procesar, la función, el rendimiento, las restricciones, las interfaces y la fiabilidad. Se evalúan las funciones descritas en la declaración del ámbito, y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación.

Características Básicas del Proyecto.

El sistema será implementado como una aplicación Web. Existirán inicialmente 7 tipos diferentes de usuario. Los usuarios deberán autenticarse en el sistema (suministrar nombre y contraseña), pudiendo entonces consultar la información a la que tienen acceso. De no haberse registrado no podrá hacerlo. En este proceso se pedirá al usuario la información necesaria para el sistema pueda realizar un reconocimiento básico del usuario y asignar las responsabilidades que el mismo podrá utilizar dentro del sistema.

El proyecto se divide básicamente en 7 módulos, los cuales cumplen sus respectivas funcionalidades,

- Presentación.
- Contratación.
- Seguimiento.
- Configuración del sistema.
- Administración.
- Coordinación de Misiones.
- Gestión de Financiamiento en Cuba.

Inicialmente se hará referencia a 1 de ellos por prioritario, a petición del cliente.

Presentación: Se informatizarán los procesos relacionados con la etapa de presentación de proyectos en un determinado período donde se conforman los proyectos, se evalúan y se asigna el financiamiento.

Restricciones generales del proyecto.

- El sistema debe poder usarse en cualquier sistema operativo.
- El sistema debe ser Web, para que sea accesible desde Internet.

- El sistema debe de estar probando en un plazo realmente corto.

Funcionalidades del Módulo de Presentación.

Proponer apertura de un período de Presentación de Proyectos

1. Insertar datos del período de Presentación de Proyectos.
2. Rechazar propuesta del período de Presentación de Proyectos.
3. Aceptar propuesta del período de Presentación de Proyectos.
4. Notificar apertura de un período de Presentación de Proyectos.
5. Modificar datos de un período de Presentación de Proyectos.
6. Aceptar modificación del período de Presentación de Proyectos.
7. Notificar modificación de un período de Presentación de Proyectos.
8. Rechazar modificación del período de Presentación de Proyectos.

Concebir Proyectos

9. Proponer Proyecto
10. Rechazar propuesta de proyecto.
11. Notificar rechazo de propuesta de proyecto.
12. Aceptar propuesta de proyecto.
13. Notificar aceptación de propuesta de proyecto.
14. Modificar una propuesta de proyecto.
15. Eliminar propuesta de proyecto.
16. Modificar ficha de Proyecto.
17. Rechazar modificación de ficha de Proyecto.
18. Notificar rechazo de modificación de ficha de Proyecto.

19. Aceptar modificación de ficha de Proyecto.
20. Notificar aceptación de modificación de ficha de Proyecto.
21. Eliminar ficha de Proyecto.
22. Rechazar eliminación de ficha de proyecto.
23. Notificar rechazo de eliminación de ficha de Proyecto.
24. Aceptar eliminación de proyecto.
25. Notificar aceptación de eliminación de ficha de Proyecto.
26. Enviar ficha del Proyecto a los Ministerios.
27. Aceptar envío de Proyecto a los Ministerios.
28. Notificar aceptación de envío de Proyecto a los Ministerios.
29. Rechazar envío de Proyecto a los Ministerios.
30. Notificar rechazo de envío de Proyecto a los Ministerios.
31. Rechazar Proyecto por los Ministerios.
32. Notificar rechazo de proyecto por los Ministerios.
33. Aceptar Proyecto por los Ministerios.
34. Notificar aceptación de de proyectos por los Ministerios.
35. Insertar actividad.
36. Insertar los recursos humanos del proyecto.
37. Asignar recursos humanos a una actividad.
38. Asignar recursos materiales.
39. Asignar gastos administrativos.
40. Asignar otros tipos de recursos.

- 41. Modificar actividad.
- 42. Modificar recursos humanos del proyecto.
- 43. Modificar recursos humanos de una actividad.
- 44. Modificar recursos materiales.
- 45. Modificar gastos administrativos.
- 46. Modificar otros tipos de recursos.
- 47. Agregar tipo de recurso.
- 48. Reporte de ficha de proyecto.
- 49. Imprimir ficha del proyecto.

Presentar Proyectos

- 50. Proponer Presentación de Proyectos.
- 51. Aceptar Presentación de Proyectos.
- 52. Notificar aceptación de Presentación de Proyecto.
- 53. Rechazar Presentación de Proyectos.
- 54. Notificar rechazo de Presentación de Proyectos.

Presentar proyectos expeditos

- 55. Proponer Presentación de Proyectos Expeditos.
- 56. Notificar proposición de Presentación de Proyectos Expeditos.
- 57. Rechazar proposición de Presentación de Proyectos Expeditos.
- 58. Notificar rechazo de proposición de Presentación de Proyectos Expeditos.
- 59. Aceptar Presentación de Proyectos Expeditos.
- 60. Notificar aceptación de Presentación de Proyectos Expeditos.

Evaluar Proyectos

- 61. Aceptar proyecto.
- 62. Notificar aceptación de proyecto.
- 63. Rechazar proyecto.
- 64. Notificar rechazo del proyecto.

Cerrar el período de Presentación de Proyectos

- 65. Proponer cierre del período de Presentación de Proyectos.
- 66. Notificar propuesta de cierre del período de Presentación de Proyectos.
- 67. Rechazar propuesta de cierre del período de Presentación de Proyectos.
- 68. Notificar rechazo de cierre del período de Presentación de Proyectos.
- 69. Aceptar cierre del período de Presentación de Proyectos.
- 70. Notificar cierre del período de Presentación de Proyectos.
- 71. Aprobar financiamiento.
- 72. Reportar proyectos.
- 73. Reportar fichas de proyectos.
- 74. Imprimir fichas de proyectos.
- 75. Reportar inversión histórica.
- 76. Imprimir reporte de inversión histórica.
- 77. Reportar Montos invertidos Imprimir reporte de montos invertidos.
- 78. Reportar Proyectos por sectores.
- 79. Imprimir reporte de proyectos por sectores.
- 80. Reportar Inversión por sectores.

81. Imprimir reporte de inversión por sectores.
82. Reportar Proyectos por ministerios.
83. Imprimir reporte de proyectos por ministerios.
84. Reportar Inversión por ministerios.
85. Imprimir reporte de inversión por ministerios.
86. Crear Noticias.
87. Publicar Noticia.
88. Editar Noticia.
89. Eliminar Noticia.
90. Eliminar Publicación.
91. Mostrar Noticia.

2.2.2 Estimación de los Recursos.

La estimación de recursos es la segunda actividad que se realiza para la planificación del desarrollo de un software una vez establecido el ámbito del software.

En el capítulo anterior se detalla cada uno de los siguientes recursos, por lo que este se centrará en la estimación como tal de cada uno, dado el ámbito del proyecto Informatización del Convenio Cuba – Venezuela.

2.2.2.1 Recursos Humanos.

Los recursos humanos en la UCI están conformados por estudiantes y profesores. La facultad tiene varios polos productivos, a los que están asignados los proyectos y grupos que la conforman. Cuando se hace necesaria la captación de personal para la integración al equipo de desarrollo del proyecto Informatización del Convenio Cuba - Venezuela se va a los grupos que pertenecen al polo correspondiente al proyecto y se aplican pruebas, para evaluar su capacidad y seleccionar a los más capaces en dependencia de la función que se necesite deban realizar.

Una vez seleccionado el personal que trabajará en el proyecto, conjuntamente con el trabajo del mismo se comienzan hacer capacitaciones con el fin de elevar el conocimiento de cada integrante, conocimiento que posteriormente se irá evaluando con la calidad y el cumplimiento de las tareas que se le asignen.

Cada persona tiene un horario o tiempo de máquina en horas, aproximadamente 7 horas, que se controla mediante el cumplimiento de las tareas que le son asignadas.

Teniendo en cuenta que de 5 días laborables que tiene la semana el estudiante trabaja 3, pues los otros 2 se lo dedican a la docencia, y que un mes tiene aproximadamente 4 semanas, se puede decir que un estudiante trabaja 84 horas mensuales.

2.2.2.2 Recursos de Software Reutilizables.

Teniendo en cuenta las cuatro categorías que define Bennatan descritas en el Capítulo 1, y que el proyecto Informatización del Convenio Cuba – Venezuela es un proyecto nuevo, diferente a otro que se haya hecho en la UCI, no se han aplicado recursos reutilizables hasta el momento. En este caso los componentes deben construirse especialmente para este proyecto.

2.2.2.3 Recursos de entorno (hardware/software).

Herramientas de hardware:

Actualmente en el proyecto Informatización del Convenio Cuba – Venezuela se encuentran un total de 37 computadoras con los siguientes datos:

- Sistema Operativo: GNU/Linux
- Motherboard: P5LD2-VM, ASUSTeK Computer INC.
- CPU: Intel(R) Pentium(R)-4 CPU 3.00GHz
- HD: 120 GB.
- CDRom: DVD-RAM writer.

En las que se encuentran 22 computadoras con:

- RAM: 1 GB

Y 15 computadoras con:

- RAM: 512 MB

Todas las maquinas están disponibles las 24 horas del día equipadas con los software necesarios para el trabajo en el proyecto.

Herramientas de software:

Servidores.

Subversión: Herramienta para salvadas automáticas.

Trac: Gestor de proyecto y control y seguimiento de errores y tareas.

PostgreSQL: Gestor de BD.

En máquinas de desarrollo:

Eclipse: IDE para programar.

Visual Paradigm: Herramienta CASE.

Además de estos software, existen muchos otros que se utilizan, para acceder a los distintos tipos de servicios de los servidores, así como ellos para brindar los mismos, ejemplo: RapidSVN (Cliente SVN), LDAP (Conjunto de herramientas que brindan servicio de entrada por dominio), etc.

El lenguaje a utilizar es JAVA, y tanto este como las herramientas, se discuten entre los integrantes del proyecto (Equipo de desarrollo), pues al cliente le es transparente con lo que se haga, a no ser algún caso específico de alguno que lo pida.

2.2.3 Estimación del Proyecto de Software.

Una vez que se han seleccionado adecuadamente los recursos necesarios para trabajar en el proyecto, el próximo paso es estimar el costo y la duración del mismo.

Mediante todos los métodos de cálculo de costos y esfuerzos se puede hacer una estimación del esfuerzo y/o del tiempo de duración del sistema. Según la comparación que se realizó (Epígrafe 1.3.3.6) se concluye que el método de COCOMO II es el idóneo para estimar el costo y la duración del módulo de Presentación del Proyecto Informatización del Convenio Cuba – Venezuela.

Para la estimación por COCOMO II se aplica el submodelo Composición de Aplicaciones, el cual se utiliza durante las primeras etapas de la ingeniería del software, donde no se tiene prácticamente información acerca del mismo, es el caso del proyecto, la información que se tiene es el ámbito del software. Según avance el proyecto se aplicarán los siguientes modelos haciendo más certera la estimación.

2.2.3.1 Estimación por Modelo COCOMO II.

Debido a que el proyecto Informatización del Convenio Cuba – Venezuela está en una fase en la cual no esta determinada por completo su arquitectura se va a utilizar el **modelo de composición de aplicaciones**.

NOP: Nuevos Puntos de Objeto. (Cantidad de Puntos Objeto ajustados por la reutilización).

Srvr: Número de tablas de datos del servidor (mainframe o equivalente) usadas junto con la pantalla o el informe.

Clnt: Número de tablas de datos del cliente (estación de trabajo personal) usadas junto con la pantalla o el informe.

%reuse: Porcentaje de pantallas, informes y módulos 3GL reutilizados a partir de aplicaciones anteriores prorrateadas por grado de reutilización.

Se ha de destacar que el uso del término “objeto” en Puntos Objeto define pantallas, informes y módulos 3GL como objetos. Esto puede tener relación o no con otras definiciones de objetos como aquellas características de posesión tales como, por ejemplo, pertenencia a una clase, herencia, encapsulación, paso de mensajes y así sucesivamente.

Pasos a seguir para aplicar el modelo de composición de aplicaciones:

1. Hacer el recuento de objetos: Estimar el número de pantallas, informes y componentes de las que consta esta aplicación.

Cantidad de Pantallas: 75

Cantidad de Informes: 16

Cantidad de Componentes: 0

2. Clasificar cada instancia de objeto dentro de niveles de complejidad simple, media y difícil dependiendo de los valores de las dimensiones de la característica. Usar la tabla 2.1:

Para Pantallas				Para Informes			
No. de vistas que contiene	No. y fuentes de tablas de datos			No. de secciones que contiene	No. y fuentes de tablas de datos		
	Total <4 (<2 svr <3 clnt)	Total <8 (2 ó 3 svr 3 a 5 clnt)	Total 8+ (>3 svr >5 clnt)		Total <4 (<2 svr <3 clnt)	Total <8 (2 ó 3 svr 3 a 5 clnt)	Total 8+ (>3 svr >5 clnt)
< 3	simple	Simple	medio	0 ó 1	simple	simple	medio
3 a 7	simple	Medio	difícil	2 ó 3	simple	medio	difícil
> 8	medio	Difícil	difícil	4 +	medio	difícil	difícil

Tabla 2.1: Complejidad asociada a las instrucciones de objetos.

Complejidad de las Pantallas: 36 simple, 1 medio, 38 difícil.

Complejidad de los Informes: 12 medio, 4 difícil.

3. Pesar el número de cada celda usando la tabla 2.2. El peso refleja el esfuerzo relativo que se requiere para implementar una instancia de ese nivel de complejidad:

Tipo de Objeto	Complejidad-Peso		
	Simple	Medio	Difícil
Pantalla	1	2	3
Informe	2	5	8
Componentes 3GL			10

Tabla 2.2: Pesos asociados a los niveles de complejidad.

Peso de las Pantallas: 36 * 1, 1 * 2, 38 * 3.

Peso de los Informes: 12 * 5, 4 * 8.

4. Determinar Puntos Objeto: Suma todas las instancias de objeto pesadas para conseguir un número. El recuento de Puntos Objeto.

$$PO = 36 * 1 + 1 * 2 + 38 * 3 + 12 * 5 + 4 * 8 = 244$$

- Estimar el porcentaje de reutilización que se espera lograr en este proyecto. Calcular los nuevos Puntos Objeto a desarrollar.

$$NOP = [(Puntos\ Objeto) * (100 - \%reuse)] / 100$$

$$\%reuse = 0.$$

$$NOP = [244 * (100 - 0)] / 100$$

$$NOP = 244$$

- Determinar un ratio de productividad PROD = NOP/Meses-persona a partir de la tabla 2.3:

Experiencia y capacidad de los desarrolladores	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
Madurez y capacidad del entorno	Muy Bajo	Bajo	Nominal	Alto	Muy Alto
PROD	4	7	13	25	50

Tabla 2.3: Ratio de productividad PROD.

$$PROD = 13$$

- Calcular el valor Meses-persona estimado según la ecuación:

$$MM = NOP / PROD$$

$$MM = 244 / 13 = 18.77 \text{ meses x hombre.}$$

$$18.77 \text{ (meses * hombre)} / 6 \text{ (personas que trabajan en el módulo)} = 3.13 \text{ (meses * 6 hombres)}$$

El equipo de desarrollo que trabaja en el módulo de Presentación del Proyecto Informatización del Convenio Cuba – Venezuela necesita 3.13 meses de trabajo.

2.2.4 Análisis y Gestión del Riesgo.

De las dos estrategias de riesgo que se vieron en el epígrafe 1.3.4, la proactiva será la que se ponga en práctica en el proyecto Informatización del Convenio Cuba - Venezuela. La estrategia proactiva es más segura y confiable por lo que el análisis y la gestión del riesgo se harán basándose en ella.

2.2.4.1 Identificación del Riesgo.

Aquí se presenta la lista de riesgos identificados en el proyecto Informatización del Convenio Cuba – Venezuela.

Personal.

1. Falta de conocimiento en la utilización de Frameworks especializados (Spring MVC, Hibernate, Spring).
2. Falta de experiencia con la herramienta Visual Paradigm para modelar.
3. Carencia de tiempo de los desarrolladores.
4. Capacidad limitada para viajar.
5. Las actividades extraclases en el horario del proyecto (Trabajo Socialmente Útil (TSU), consultas) de los estudiantes afecta el cumplimiento de algunas tareas.
6. Falta de motivación y compromiso de algunos miembros del proyecto.
7. Pérdida de integrantes del equipo de desarrollo.

Requerimientos.

8. Modelo de procesos inconsistente.
9. Clientes a tiempo parciales.

Gestión de configuración.

10. Falta de experiencia en los encargados de la GCS.

Proyecto.

11. No existe una planificación real del proyecto.

Técnico.

12. Rotura de las computadoras.
13. Caída del sistema de electricidad.

Negocio.

- 14. Cambio de líderes del proyecto.

Conocidos.

- 15. El producto a construir es muy grande.

Organización.

- 16. Cambio de roles en el equipo de desarrollo.

2.2.4.2 Proyección del Riesgo.

Como se describe en el Epígrafe 1.3.4.3, una vez identificados los riesgos deben estimarse teniendo en cuenta su probabilidad de ocurrencia y su impacto, si se llega a manifestar. El valor de la probabilidad de cada riesgo puede estimarse por cada miembro del equipo individualmente. La probabilidad es un valor cualitativo estimado por los planificadores, el jefe del proyecto y demás personas involucradas, en base a esto se le otorga un valor de probabilidad de baja, media o alta. El desarrollo de una tabla de riesgo (Tabla 2.4) le proporciona al jefe del proyecto una sencilla técnica para la proyección del riesgo.

RIESGO	TIPO DE RIESGO	IMPACTO	DESCRIPCIÓN	PROBABILIDAD	EFFECTOS
1. Falta de conocimiento en la utilización de frameworks especializados (Spring MVC, Hibernate, Spring).	Personal	<ul style="list-style-type: none"> ▪ Dificultades en las configuraciones para la implementación de los frameworks. ▪ Atraso en la implementación de componentes de software. 	La descripción de la arquitectura candidata propone la utilización de frameworks especializados, estos tienen una alta línea de aprendizaje. Existe poca experiencia.	Media	Tolerable
2. Falta de experiencia con la herramienta Visual Paradigm para modelar.	Personal	<ul style="list-style-type: none"> ▪ Dilación en la modelación de los aspectos del sistema. 	La modelación del sistema se debe realizar sobre herramientas libres, esta herramienta es desconocida por los analistas y	Media	Tolerable

			diseñadores.		
3. Carencia de tiempo de los desarrolladores.	Personal	<ul style="list-style-type: none"> ▪ No culminación de tareas a tiempo. 	Los estudiantes tienen el tiempo compartido entre la docencia, las actividades de la universidad y sus organizaciones y el proyecto productivo.	Media	Tolerable
4. Capacidad limitada para viajar.	Personal.	<ul style="list-style-type: none"> ▪ Atraso en los cronogramas. 	Parte del equipo tiene necesidad de viajar a Venezuela en determinadas etapas del proyecto y estas posibilidades están limitadas por varios factores inherentes al proyecto.	Media	Tolerable
5. Las actividades extraclases en el horario del proyecto (TSU, consultas) de los estudiantes afecta el cumplimiento de algunas tareas.	Personal	<ul style="list-style-type: none"> ▪ Atraso en los cronogramas. 	Las actividades extradocentes en ocasiones no planificadas afectan el horario de trabajo.	Media	Tolerable
6. Falta de motivación y compromiso de algunos miembros del proyecto.	Personal	<ul style="list-style-type: none"> ▪ Atraso en los cronogramas. 	La falta de motivación y compromiso provoca que las actividades no se realicen con la calidad requerida y en el tiempo preciso.	Media	Tolerable
7. Perdida de integrantes del equipo de desarrollo.	Personal	<ul style="list-style-type: none"> ▪ Repartimiento más comprimido de las tareas. ▪ Atraso en los cronogramas. 	La pérdida de integrantes del equipo de desarrollo provoca un ajuste más apretado en el trabajo y atraso en el proyecto.	Media	Serios
8. Modelo de procesos inconsistente.	Requerimientos	<ul style="list-style-type: none"> ▪ Levantamiento de requisitos inconsistente, cambios en los requerimientos 	El modelo de procesos definido hasta el momento es inconsistente debido a la poca interacción	Alta	Serios

		e inconformidades de los clientes.	con el cliente.		
9. Clientes a tiempo parciales.	Requerimientos.	<ul style="list-style-type: none"> ▪ Levantamiento de requisitos irreal. ▪ No conformidad con los requerimientos elicitados. ▪ Cambios constantes en los requerimientos. 	Los clientes con los que se trabaja constan de dos partes, la parte cubana y la venezolana, pero ninguno de los dos tiene un compromiso establecido de tiempo de atención al proyecto.	Alta	Serios
10. Falta de experiencia en los encargados de la GCS.	Gestión de configuración	<ul style="list-style-type: none"> ▪ Cambios no controlado en la línea base. 	Los encargados de la GCS no tienen experiencia en el tema.	Media	Tolerable
11. No existe una planificación real del proyecto.	Proyecto	<ul style="list-style-type: none"> ▪ Atraso en los cronogramas 	No existe una planificación real del proyecto, provocando atrasos en las distintas actividades.	Alta	Serios
12. Rotura de las computadoras.	Técnico	<ul style="list-style-type: none"> ▪ Reasignación del horario de trabajo. ▪ Perdida de información. 	La actividad en las maquinas tiene que ser compartida, porque se reduce el tiempo de maquina y consecuentemente el tiempo de producción.	Alta	Serios
13. Caída del sistema de electricidad.	Técnico	<ul style="list-style-type: none"> ▪ Atraso en las funciones del proyecto. 	Se hace imposible trabajar en ese período.	Media	Tolerable
14. Cambio de líderes del proyecto.	Negocio	<ul style="list-style-type: none"> ▪ Disociación de los integrantes del proyecto. ▪ Falta de confianza. 	No se encuentran identificados con el nuevo líder.	Media	Tolerable
15. El producto a construir es muy grande.	Conocidos	<ul style="list-style-type: none"> ▪ Consumo de mucho tiempo. 	El producto que se construye esta compuesto de muchos módulos.	Media	Tolerable

16. Cambio de roles en el equipo de desarrollo.	Organización	<ul style="list-style-type: none"> ▪ Reajuste de los roles. ▪ Atraso en los cronogramas. 	El personal asignado carece de experiencia y conocimiento en ese nuevo rol.	Media	Tolerable
---	--------------	--	---	-------	-----------

Tabla 2.4: Tabla de Riesgo.

Una vez que sean identificados y estimados los riesgos se debe definir una priorización del riesgo. Los riesgos de mayor probabilidad e impacto serán los seleccionados por el jefe del proyecto para darle tratamiento; los otros quedarán relegados a una a una lista de priorización de segundo orden.

RIESGO	PROBABILIDAD	EFFECTOS
8	Alta	Serios
9	Alta	Serios
11	Alta	Serios
12	Alta	Serios
7	Media	Serios
1	Media	Tolerable
2	Media	Tolerable
3	Media	Tolerable
4	Media	Tolerable
5	Media	Tolerable
6	Media	Tolerable
10	Media	Tolerable
13	Media	Tolerable
14	Media	Tolerable
15	Media	Tolerable
16	Media	Tolerable

Tabla 2.5: Priorización del Riesgo.

2.2.4.3 Reducción, supervisión y gestión del riesgo.

Se priorizará el tratamiento de los riesgos de mayor probabilidad e impacto; estos se muestran de forma ordenada en la Tabla 2.5 y a otros que también se consideran significativos.

Riesgo 8. (Modelo de procesos inconsistente.)

Indicadores.

Los procesos no están bien descritos por los clientes, no tienen procedimientos escritos, solo se hacen por procedimientos empíricos los que pueden cambiar con el mejoramiento y la experiencia adquirida.

Estrategia de Mitigación.

- Reunirse con el cliente.

Plan de Contingencia.

Tratar de que la aplicación sea lo más flexible posible, para en caso de no poderse evitar el riesgo se puedan cambiar los requerimientos y no afecten en gran medida el desarrollo del proyecto.

Riesgo 9. (Clientes a tiempo parciales.)

Indicadores.

Los clientes están divididos en dos partes, la cubana (Ministerio para la Inversión Extranjera y la Colaboración Económica (MINVEC)) y la venezolana (Ministerio del Poder Popular para la Energía y Petróleo (MPPEP)), ninguna de las dos tienen un compromiso establecido de tiempo de atención al proyecto.

Estrategia de Mitigación.

- Tratar de reunirse en conjunto con las dos partes.

Plan de Contingencia.

Construir el sistema lo más flexible posible para su más fácil modificación en caso de existir cambio en los requisitos. Reunirse con los clientes y llegar a un acuerdo común.

Riesgo 11. (No existe una planificación real del proyecto.)

Indicadores.

El proyecto carece de una planificación eficiente, provocando atrasos en las distintas actividades.

Estrategia de Mitigación.

- Trazar una estrategia de planificación.
- Mantener un control sistemático sobre las actividades del proyecto.

Plan de Contingencia.

Reajustar la planificación temporal.

Riesgo 12. (Rotura de las computadoras.)

Indicadores.

El trabajo se realiza en las computadoras y una rotura significa, posibles pérdidas de documentos, pérdida de tiempo de producción y reasignación del tiempo de máquina,

Estrategia de Mitigación.

- Mantener actualizados los antivirus en las PC de trabajo.
- Subir las tareas terminadas al repositorio.
- Solicitar mantenimiento a servicios técnicos.

Plan de Contingencia.

- Solicitar servicios técnicos.
- Sustituir la PC.

Riesgo 7. (Pérdida de integrantes del equipo de desarrollo.)

Indicadores.

Existen inconformidades entre los miembros del proyecto, lo que puede llevar a la pérdida de integrantes del equipo de desarrollo.

Estrategia de Mitigación.

- Reunirse con la plantilla y tratar de darle solución a las inquietudes.
- Desarrollar técnicas que permitan la continuidad del proyecto.

Plan de Contingencia.

- Los que abandonan el proyecto deben dedicar sus últimos días a transmitir sus conocimientos.

- Reajuste de la planificación temporal.

Riesgo 1. (Falta de conocimiento en la utilización de frameworks especializados (Spring MVC, Hibernate, Spring).)

Indicadores.

La arquitectura propuesta por el equipo de arquitectura del proyecto propuso la utilización de frameworks especializados para la implementación. Dichos frameworks cuando se dominan pueden hacer mucho más rápido el desarrollo de aplicaciones pero es muy lento el aprendizaje de los mismos, el equipo cuenta con poca experiencia.

Estrategia de Mitigación.

Se está prestando mucha atención a la capacitación del equipo de desarrolladores del proyecto.

- Curso de capacitación de Hibernate.
- Curso de capacitación de Spring.

Plan de Contingencia.

Con estudiantes que ya conocen y tienen una experiencia media en el uso de estos frameworks tratar de hacer ejemplos prácticos con pequeños casos de uso a implementar del propio proyecto.

Riesgo 2. (Falta de experiencia con la herramienta Visual Paradigm para modelar.)

Indicadores.

Con vistas a lograr la soberanía tecnológica de la universidad y el país se ha tomado la medida de comenzar a realizar la modelación del sistema utilizando la herramienta Visual Paradigm la cual permite realizar la modelación mediante UML. Los desarrolladores del proyecto en su mayoría, no poseen experiencia en el uso de esta herramienta.

Estrategia de Mitigación.

- Se está gestionando un curso de capacitación para los analistas.
- Se está modelando aún con otras herramientas

Plan de Contingencia.

- Se reevaluará el uso de dicha herramienta.
- Se priorizará la capacitación en dicha herramienta.

Riesgo 3. (Carencia de tiempo de los desarrolladores.)

Indicadores.

Los integrantes que trabajan en el proyecto son en su mayoría estudiantes de 4to y 5to año. Estos son generalmente programadores o diseñadores de base de datos y algunos analistas; al ser estudiantes tienen un tiempo parcial en el proyecto puesto que tienen que compartir el tiempo con la docencia, la producción y las actividades de las organizaciones a las que pertenecen.

Estrategia de Mitigación.

- Se encuentran todos los estudiantes de 4to año en el mismo grupo.
- Algunos profesores del proyecto son sus docentes.
- Sus clases se han reajustado para darlas en dos días de la semana y que todo el resto del tiempo pudieran dedicarlo al proyecto.

Plan de Contingencia.

- Pudiera analizarse el sistema de evaluación de las asignaturas y proponer algún tipo de cambio.
- Acreditación de asignaturas por el proyecto, con previo análisis de los planes de estudio y los objetivos de cada asignatura.

Después de haber trazado una estrategia para la reducción, supervisión y gestión del riesgo se le debe dar seguimiento, se debe monitorear el riesgo según sus indicadores. A la vez que se ha desarrollado el Plan RSGR comienzan los procedimientos de reducción y supervisión del riesgo. Se tiene que controlar que se cumpla con la estrategia de mitigación asociada a cada riesgo, tratando en todos los casos de evitar el riesgo.

2.2.5 Planificación Temporal.

Después de haber hecha una estimación de los recursos humanos, recursos de software reutilizable y de

entorno, teniendo en cuenta el tiempo de duración del proyecto (fecha límite) y los posibles riesgos que puedan retrasar el proyecto; es hora de comenzar a asignar responsabilidades, velar porque estas se cumplan, asegurarse de que los posibles riesgos no se conviertan en realidad, todo esto forma parte del plan de proyecto.

La planificación temporal de un proyecto de software es una actividad que distribuye el esfuerzo estimado a lo largo de la duración prevista del proyecto, asignando el esfuerzo a las tareas específicas de la ingeniería del software. Permite la identificación de tareas, asignación de tiempos y recursos a dichas tareas y planificación de la secuencia de ejecución de forma que el tiempo de desarrollo del proyecto sea mínimo.

Principios básicos de la planificación temporal:

1. Compartimentación: Número y tareas manejables.

Teniendo en cuenta la complejidad del proyecto, según el ámbito definido y su impacto comercial, para llevar a cabo la comunicación con el cliente se desarrollan las siguientes actividades:

- Se revisa la petición del cliente.
- Se planifica una reunión con el cliente.
- Se realizan investigaciones para definir soluciones propuestas.
- Se realiza la reunión.
- Se plasma en un documento formal de trabajo, las especificaciones que reflejaron las funciones y características del software.
- Se revisan estas especificaciones para comprobar su corrección y consistencia.
- Se define el alcance del proyecto teniendo en cuenta tales especificaciones.

Todas estas especificaciones y definiciones se registran en el documento Visión del proyecto.

2. Asignación de tiempo: Cada actividad implica un tiempo de desarrollo.

A cada tarea se le asignó una fecha de inicio y otra de finalización.

3. Interdependencia: Saber que relación hay entre las tareas. Dependencia, paralelismo, independencia.

Se determinan las interdependencias de cada tarea. Teniendo en cuenta que algunas de estas pueden desarrollarse en una secuencia determinada, o pueden darse en paralelo, otras no pueden empezar

hasta que el resultado de otras no esté disponible.

4. **Validación de esfuerzo:** Revisión sobre los recursos asignados a una actividad a través del tiempo.

Para la asignación del tiempo se tiene en cuenta la cantidad de miembros del proyecto, para que no existan problemas en cuanto a la relación entre la asignación del esfuerzo y la cantidad de personas disponibles. El número de personas asignadas no puede exceder a la cantidad de personas que integran actualmente el proyecto y la asignación de tareas se hace de forma balanceada.

5. **Responsabilidad definida:** Tareas con nombre y apellido.

A cada tarea se le asigna un responsable, que responda por el desarrollo de la misma.

6. **Hitos definidos:** Cada tarea pertenece a un hito del proyecto.

Todas las tareas se asocian a un hito, el cual se consigue una vez revisado la calidad del producto.

Una vez definidas las tareas e hitos, se le da seguimiento y se controlan los mismos a medida que progresa el proyecto. Empleando tablas y gráficos de tiempo se lleva a cabo el seguimiento a través de:

- Reuniones periódicas del estado del proyecto, en las cuales el jefe de equipo informa el estado actual de cada una de las tareas asignadas a su equipo de trabajo, así como los problemas presentados en el desarrollo de las mismas.
- Verificaciones sobre el cumplimiento de los hitos definidos en la fecha programada.
- Comparaciones entre las fechas reales de inicio y las previstas para cada tarea.

7. **Resultados definidos:** Cada actividad debe aportar un resultado claramente definido (tiene que ser entregable).

Dado el tamaño y la envergadura que se cree que tenga el proyecto Informatización del Convenio Cuba – Venezuela, teniendo en cuenta la comparación que se realizó anteriormente se utilizará la técnica Pert.

Para el desarrollo de esta técnica se siguen estos pasos:

1. Representar un grafo de Pert.
2. Identificar el camino crítico y la holgura de las actividades.

3. Hacer el calendario de ejecución del proyecto.

2.2.5.1 Planificación temporal por la Técnica Pert.

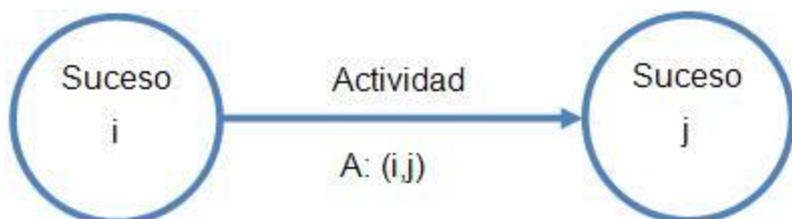
La Técnica Pert parte de la descomposición del proyecto en actividades. Entendiendo por actividad la ejecución de una tarea que exige para su realización el uso de recursos.

Se establece también el concepto de suceso: acontecimiento que indica el principio o fin de una actividad o conjunto de actividades. No consume tiempo ni recursos.

El método utiliza una estructura de grafo para la representación gráfica de las actividades o tareas de un proyecto, sus tiempos de comienzo y finalización y las dependencias entre las distintas actividades.

Las actividades se representan por líneas o flechas (arcos del grafo).

Los sucesos se representan por círculos (vértices del grafo).

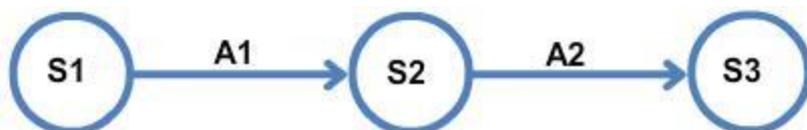


Tipos de relaciones entre las actividades:

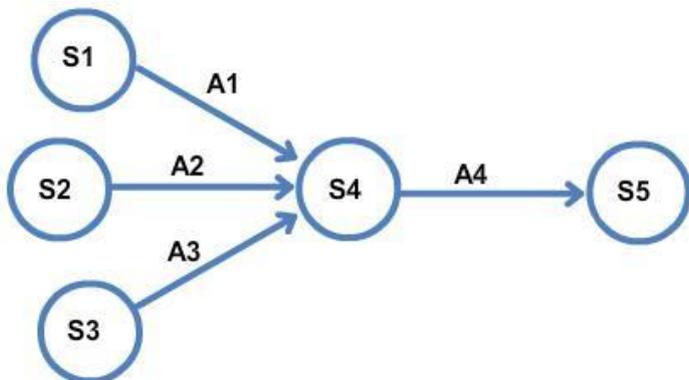
Una vez descompuesto el proyecto en actividades, la fase siguiente del Pert consiste en establecer las “prelaciones” o “prioridades” existentes entre las diferentes actividades, debidas a razones de tipo técnico, económico o jurídico. (Es decir, las diferentes actividades que constituyen un proyecto deben ejecutarse según un cierto orden).

Las prioridades o prelaciones se representan en el grafo por medio de flechas que indican que una actividad precede a otra. Existen varios tipos de prelaciones.

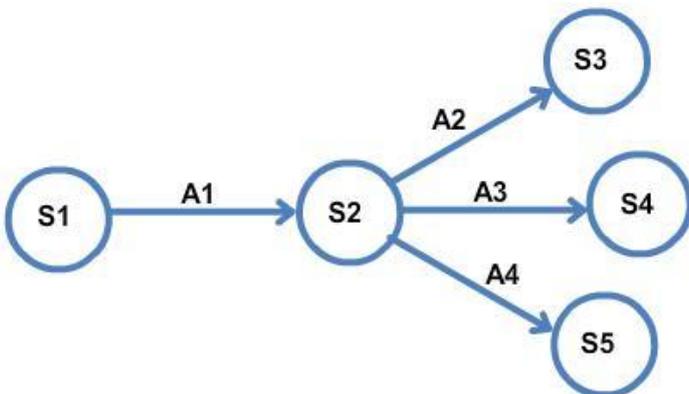
– **Prelaciones lineales:** Para poder iniciar una determinada actividad es necesario que haya finalizado una única actividad.



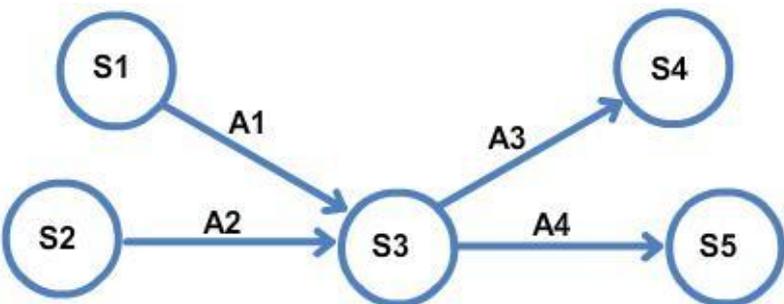
- **Prelaciones que originan una convergencia:** Para poder iniciar una determinada actividad es necesario que hayan finalizado dos o más actividades.



- **Prelaciones que originan una divergencia:** Para poder iniciarse un conjunto de actividades es necesario que haya finalizado una única actividad.



- **Prelaciones que originan convergencia-divergencia:** Para poder iniciarse un conjunto de actividades es necesario que hayan finalizado dos o más actividades.

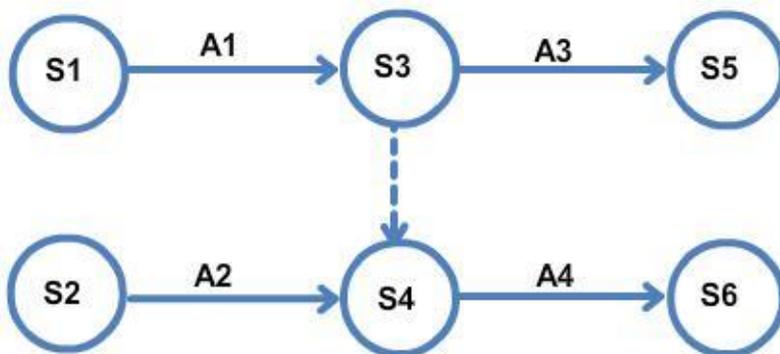


Actividades ficticias:

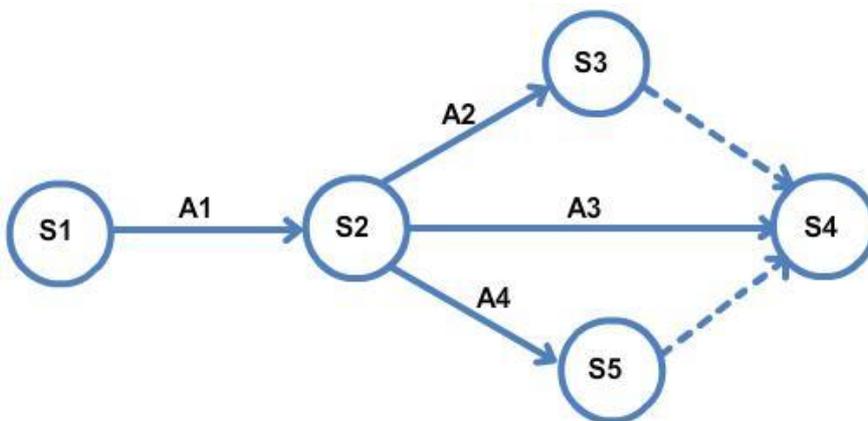
son actividades que no consumen tiempo ni recursos, sólo reflejan prelacones existentes entre distintas actividades del proyecto.

Se utilizan en dos casos:

– Cuando se presentan simultáneamente prelacones lineales y de convergencia o divergencia:



– Con actividades paralelas:



2.2.5.1.1 Representar un grafo de Pert.

Relaciones: Se comienza recogiendo de manera sistematizada toda la información referente a las prelacones entre las distintas actividades que a continuación se definen:

A1. Refinamiento y validación de los requisitos del módulo de Presentación, Venezuela.

A2. Refinamiento y validación de los requisitos del módulo de Presentación, Cuba.

- A3. Revisión por calidad del proyecto de la documentación del módulo de Presentación.
- A4. Revisión por Calidad UCI de la documentación del módulo Presentación.
- A5. Revisión por Calidad UCI de la documentación del módulo Configuración.
- A6. Firma de los requerimientos y prototipo de aplicación (Solo los módulos comunes).
- A7. Diseño del módulo de Presentación.
- A8. Implementación del módulo de Presentación.
- A9. Prueba del módulo de Presentación.
- A10. Solución a las no conformidades detectadas V0.5.
- A11. Solución a las no conformidades detectadas V1.
- A12. Creación de nuevos usuarios.
- A13. Definición de cronograma para pruebas de aceptación con ST Venezuela y Cuba.
- A14. Instalación de la aplicación en los servidores.
- A15. Carga inicial del sistema. (Usuarios, Proyectos, Ministerios, Entes).
- A16. Pruebas de aceptación de la aplicación V0.5.
- A17. Solución a las no conformidades detectadas.
- A18. Coordinar la participación en el piloto de la aplicación V0.5.
- A19. Piloto de la aplicación V0.5.
- A20. Solución a las No Conformidades detectadas en el piloto.
- A21. Superación de los organismos e incorporación de los Entes.

Existen dos procedimientos para recoger las prelacones entre las distintas actividades:

- **Matriz de encadenamientos:** matriz cuadrada cuya dimensión es igual al número de actividades en que se ha descompuesto el proyecto. Si en los puntos de cruce aparece una X indica que para

poder iniciar la actividad de la fila tiene que haber terminado la correspondiente a la columna.

Actividad	A1	A2	A3	A4	A5
A1					
A2	X				
A3	X				
A4					
A5			X		

- **Cuadro de prelações:** tabla de dos columnas, en la primera se encuentran las actividades del proyecto y en la segunda figuran las actividades precedentes de su homologa en la primera columna.

Actividad	Precedente
A1	
A2	A1
A3	A1
A4	
A5	A3

Construir el grafo: El grafo comienza en un vértice que representa el suceso inicio del proyecto y termina en otro vértice que representa el suceso fin del proyecto.

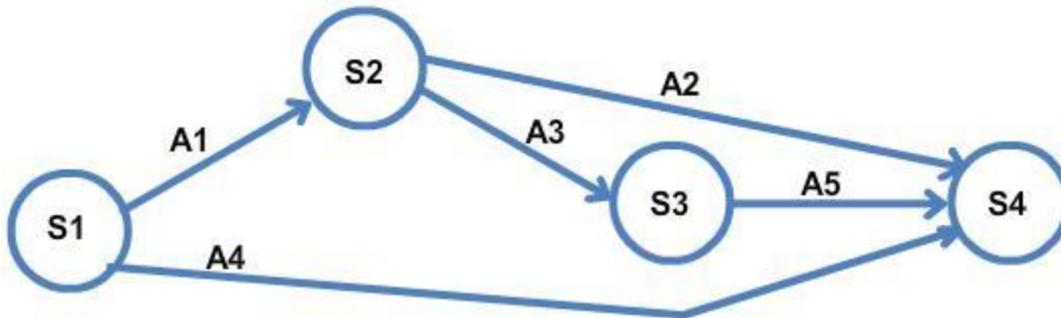
- **Suceso inicio del proyecto:** representa el inicio de una o más actividades pero no representa el fin de ninguna.
- **Suceso fin del proyecto:** representa el fin de una o más actividades pero no representa el comienzo de ninguna.
- **Actividades inicio del proyecto:** no tienen ninguna actividad precedente.
- **Actividades fin del proyecto:** no preceden a ninguna otra actividad.

La numeración de los vértices del grafo debe cumplir siempre la siguiente condición:

El número del vértice que represente el comienzo de cierta actividad debe ser menor que el número del

vértice que represente el suceso fin de esa actividad.

Construcción del Gráfico Pert:



Asignación de tiempos a las actividades

La duración de una actividad no puede fijarse, en la mayoría de los casos, con exactitud. Depende de circunstancias aleatorias (averías en las máquinas, cortes de energía eléctrica, retraso en la entrega de suministros, enfermedad del personal). Este problema es abordado por el método Pert de modo muy peculiar, pues considera tres estimaciones de tiempo distintas:

- **Estimación optimista (E_o):** tiempo mínimo en que podría ejecutarse la actividad *i* si no surgiera ningún contratiempo.
- **Estimación más probable o estimación modal (E_m):** tiempo que se empleará en ejecutar la actividad *i* en circunstancias normales.
- **Estimación pesimista (E_p):** tiempo máximo de ejecución de la actividad *i* si las circunstancias son muy desfavorables.

Para distribuciones del tipo beta las expresiones de Esperanza matemática (que expresan la duración de la actividad *i* y la varianza de la actividad *i* son las siguientes:

El tiempo Pert (**D**) o duración será la media o esperanza matemática:

$$D = (E_o + 4 \cdot E_m + E_p) / 6$$

Varianza de una actividad: Las actividades con mayor varianza tienen un mayor riesgo en la estimación de su duración.

$$\sigma_{ij}^2 = [(E_o - E_p) / 6]^2$$

Actividad	Predecesor	Optimista (Eo)	Probable (Em)	Pesimista (Ep)	Duración (Tij)
A1		14	19	21	19
A2	A1	3	6	14	7
A3		3	6	14	7
A4	A3	14	15	21	16
A5		14	15	21	16
A6	A5	1	2	3	2
A7	A2	7	15	17	14
A8	A7	28	50	60	48
A9	A8	14	15	21	16
A10		21	24	30	25
A11	A8	45	60	63	58
A12		7	15	20	15
A13	A11	1	2	3	2
A14	A13	3	5	7	5
A15	A14	1	2	3	2
A16	A15	3	5	7	5
A17	A16	1	3	5	3
A18		1	3	5	3
A19		5	7	10	7
A20	A19	5	8	14	9
A21		15	22	30	22

Tabla 2.6: Asignación de tiempos de las actividades.

La representación gráfica de la asignación de tiempos (Tabla 2.6), se muestra en el grafo de actividades de Pert (ver Anexo 1).

Cálculo de los tiempos EET y LET

Una vez construido el grafo del proyecto y asignados tiempos de ejecución a las actividades, el siguiente paso consistirá en calcular dos parámetros para cada suceso:

Sea t_{ij} el tiempo Pert de una actividad (i, j):

– **EET (Earliest Even Time)**: Se define el tiempo más pronto posible (“early”) de un suceso j . Representa el tiempo mínimo a emplear para poder empezar cualquier actividad que salga del suceso j . El EET del suceso inicial es cero, para el resto de los sucesos el EET se calcula siguiendo las siguientes reglas:

- Seleccionar todas las actividades que llegan al suceso.
- Para cada actividad que entra, se suma la duración de la actividad y el tiempo early, EET de su suceso inicial.
- Seleccionar el EET más alto que se haya obtenido.

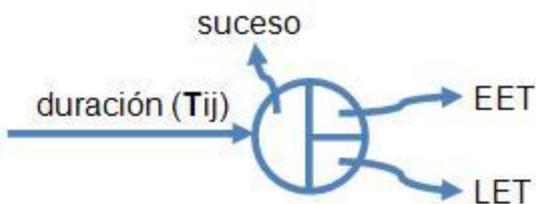
$$EET_j = \max [EET_i + D] \forall_i$$

– **LET (Latest Even Time)**: Se define como el tiempo más tarde permisible (“last”) de un suceso i . Representa lo más tarde que puede llegar a ese suceso, sin que afecte a la duración del proyecto.

El suceso fin del proyecto tiene LET igual al EET, para el resto de los sucesos se aplican las reglas siguientes:

- Considerar todas las actividades que salen del suceso.
- Restar al LET del suceso final la duración de cada actividad.
- Seleccionar el menor LET que se haya obtenido

$$LET_i = \min [LET_j - D] \forall_j$$



Cuando el grafo Pert es muy grande (muchas actividades) el cálculo de los tiempos “más pronto posible” y “más tarde permisible” puede ser muy engorroso. Por eso Zaderenko propuso un método matricial de cálculo de tiempos “más pronto posible” y “más tarde permisible”, que resulta sencillo para grafos grandes y pequeños, y además, es fácil de programar.

a) Se construye una matriz cuadrada A de dimensión igual al número de nodos del grafo y tal que a_{ij} es el tiempo PERT ó Duración de la actividad (i, j) si tal actividad existe, sino, a_{ij} no está definido. ($a_{ij} \in D; H$)

b) Para calcular los tiempos “early” se comienza por agregar una columna adicional a la izquierda de la matriz (en la que se irán escribiendo los tiempos “early”. El primer elemento de la columna es el cero. Para calcular los tiempos “early” de los demás sucesos:

$$EET_j = \max \{EET_i + D\} \text{ si el elemento } a_{ij} \text{ existe para la columna } j.$$

c) Para calcular los tiempos “más tarde permisible” se agrega una fila adicional en la parte inferior de la matriz. El último elemento de la fila es igual al tiempo “más pronto posible” del nodo final. Para calcular los tiempos “más tarde permisible” de los demás sucesos:

EET		S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	
0	S1		19	7	16	25	15	3	7	22										
19	S2										7									
7	S3																		16	
16	S4																		2	
25	S5																		0	
15	S6																		0	
3	S7																		0	
7	S8																		9	
22	S9																		0	
26	S10											14								
40	S11												48							
88	S12													58					16	
146	S13														2					
148	S14															5				
153	S15																2			
155	S16																	5		
160	S17																		3	
163	S18																			
	LET	0	19	147	161	163	163	163	163	154	163	26	40	88	146	148	153	155	160	163

Tabla 2.7: Cálculo de tiempos EET y LET.

2.2.5.1.2 Identificar el camino crítico y la holgura de las actividades.

La información que proporciona el conocimiento de los tiempos “más pronto posible” y “más tarde permisible” de los diferentes sucesos no es demasiado importante, salvo los del suceso fin del proyecto, pues este tiempo representa la duración del proyecto.

La verdadera importancia de los tiempos “más pronto posible” y “más tarde permisible” es que constituyen la base para el cálculo de las holguras, que son la pieza fundamental en todo el proceso de análisis del método Pert.

La holgura de cierto suceso i (H_i) se define como la diferencia entre los tiempos LET y EET: indica el número de unidades de tiempo en que puede retrasarse la realización del mismo, de manera que la duración del proyecto no experimente ningún retraso.

$$H_i = LET_i - EET_i$$

Holgura total de cierta actividad ij (H_{ij}^T) se define como el tiempo que resulta de restar el tiempo LET del suceso final del EET del suceso inicial y la duración de esa actividad:

$$H_{ij}^T = LET_j - EET_i - D$$

La holgura total de una actividad indica el número de unidades de tiempo en que puede retrasarse la realización de la actividad con respecto al tiempo Pert previsto, de manera que la duración del proyecto no se retrase.

Aquellas actividades cuya holgura total sea cero se denominan actividades críticas. El camino que forman se denomina camino crítico. El retraso en la realización de cualquiera de las actividades críticas producirá un retraso en la finalización del proyecto.

El camino crítico es el camino de longitud generalizada máxima que va desde el vértice que representa el suceso inicio del proyecto al vértice que representa el suceso fin del proyecto.

Cuando existen varios caminos críticos se pueden aplicar criterios estadísticos para establecer diferentes índices de criticidad (ver Anexo 2).

Las líneas con holgura cero forman el camino crítico.

2.2.5.1.3 Calendario de ejecución del proyecto.

En el calendario se establecen cuatro fechas para cada una de las actividades:

- Fecha de comienzo más temprana: EET_i
- Fecha de comienzo más tardía: $LET_j - D$
- Fecha de finalización más temprana: $EET_i + D$
- Fecha de finalización más tardía: LET_f

En el caso de las actividades críticas las fórmulas anteriores coinciden.

La holgura total de una actividad es igual a la diferencia entre las fechas de comienzo más tardía y más temprana e igual a la diferencia entre las fechas de finalización más tardía y más temprana:

$$H_{ij}^T = (LET_j - D) - EET_i = LET_f - (EET_i + D)$$

Act.	Prec.	D	H	Sucesos		Comienzos		Finalizaciones	
				Inicio	Final	Más Temprana	Más Tardía	Más Temprana	Más Tardía
						EET inicial	LET final - D	EET inicial + D	LET final
A1		19	0	S1	S2	0	0	19	19
A2	A1	7	0	S2	S10	19	19	26	26
A3		7	140	S1	S3	0	140	7	147
A4	A3	16	140	S3	S18	7	147	23	163
A5		16	145	S1	S4	0	145	16	161
A6	A5	2	145	S4	S18	16	161	18	163
A7	A2	14	0	S10	S11	26	26	40	40
A8	A7	48	0	S11	S12	40	40	88	88
A9	A8	16	59	S12	S18	88	147	104	163
A10		25	138	S1	S5	0	138	25	163
A11	A8	58	0	S12	S13	88	88	146	146
A12		15	148	S1	S6	0	148	15	163
A13	A11	2	0	S13	S14	146	146	148	148
A14	A13	5	0	S14	S15	148	148	153	153

A15	A14	2	0	S15	S16	153	153	155	155
A16	A15	5	0	S16	S17	155	155	160	160
A17	A16	3	0	S17	S18	160	160	163	163
A18		3	160	S1	S7	0	160	3	163
A19		7	147	S1	S8	0	147	7	154
A20	A19	9	147	S8	S18	7	154	16	163
A21		22	141	S1	S9	0	141	22	163

Tabla 2.8: Calendario de ejecución del proyecto I.

Actividad	Comienzos		Finalizaciones	
	Más Temprana	Más Tarde	Más Temprana	Más Tarde
	EET inicial	LET final - D	EET inicial + D	EET inicial + D + H
A1	23/01/2008	23/01/2008	11/2/2008	11/2/2008
A2	11/2/2008	11/2/2008	18/02/2008	18/02/2008
A3	23/01/2008	12/6/2008	30/01/2008	19/06/2008
A4	30/01/2008	19/06/2008	15/02/2008	5/7/2008
A5	23/01/2008	17/06/2008	8/2/2008	3/7/2008
A6	8/2/2008	3/7/2008	10/2/2008	5/7/2008
A7	18/02/2008	18/2/2008	4/3/2008	4/3/2008
A8	4/3/2008	4/3/2008	21/04/2008	21/04/2008
A9	21/04/2008	19/06/2008	7/5/2008	5/7/2008
A10	23/01/2008	10/6/2008	17/02/2008	5/7/2008
A11	21/04/2008	21/04/2008	18/06/2008	18/06/2008
A12	23/01/2008	20/06/2008	7/2/2008	5/7/2008
A13	18/06/2008	18/06/2008	20/06/2008	20/06/2008
A14	20/06/2008	20/06/2008	25/06/2008	25/06/2008
A15	25/06/2008	25/06/2008	27/06/2008	27/06/2008
A16	27/06/2008	27/06/2008	2/7/2008	2/7/2008
A17	2/7/2008	2/7/2008	5/7/2008	5/7/2008
A18	23/01/2008	2/7/2008	26/01/2008	5/7/2008
A19	23/01/2008	19/06/2008	30/01/2008	26/06/2008
A20	30/01/2008	26/06/2008	8/2/2008	5/7/2008
A21	23/01/2008	13/06/2008	14/02/2008	5/7/2008

Tabla 2.9: Calendario de ejecución del proyecto II.

2.3 Control Sistemático.

Para controlar sistemáticamente cada una de las actividades desarrolladas durante el desarrollo del proyecto se siguen una serie de actividades tales como:

- Reuniones periódicas para evaluar progreso.
- Determinar hitos cumplidos.
- Comparando la fecha real de inicio con las previstas.
- Evaluar los resultados de las revisiones.

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto.

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada periódicamente.

Mediante la herramienta Trac se sigue la evolución del proyecto, se visualiza el estado de las tareas de cada uno de los hitos.

2.4 Conclusiones del capítulo.

En este capítulo se definió la propuesta para la estrategia de planificación y se comenzó a aplicar, con el propósito de darle solución a los objetivos propuestos. Se determinaron los recursos, así como los riesgos que podrían poner en peligro el curso del proyecto, por lo que se formuló una estrategia de mitigación para evitarlos. Se efectuó la estimación del esfuerzo y duración a través del método COCOMO II. Además se estimó el tiempo de inicio y culminación de las tareas del módulo presentación, mediante el método de Pert. Inicialmente se trabajará en la planificación de este módulo como una forma de prueba para su posterior aplicación al resto del proyecto.

Capítulo 3: Resultados.

3.1 Introducción.

En el presente capítulo se hace un análisis de los resultados en cada una de las actividades desarrolladas durante la aplicación de la estrategia de gestión de planificación al módulo Presentación del proyecto Informatización del Convenio Cuba - Venezuela. Inicialmente la planificación se realizaba de manera informal, y con esta propuesta aplicada parcialmente a este primer módulo se han visualizado mejoras en cuanto a la organización.

3.2 Ámbito del software.

Con la definición del ámbito de software se logró un entendimiento básico del problema, posibilitando el desarrollo del proyecto Informatización del Convenio Cuba – Venezuela, en especial del módulo de Presentación, el cual será el primero en entregar a petición del cliente.

En sesiones de trabajo con el cliente en Venezuela se logró describir las funcionalidades de este módulo, el rendimiento, las restricciones, las interfaces y la fiabilidad a partir de un arduo trabajo protagonizado por el gerente del proyecto, el líder de la solución de software, el analista principal, el arquitecto principal y el jefe de módulo. Además en otras sesiones de trabajo ahora focalizadas en Cuba se refinan para dar más detalles antes del comienzo de la estimación.

Lógicamente el buen trabajo y bien definido de estos compañeros ayudaron en lograr efectividad en las estimaciones utilizando los métodos de estimación descritos anteriormente.

A pesar que determinados autores afirman que esta perspectiva inicial involucra reuniones entre el analista y al cliente, para definir las expectativas del proyecto y ponerse de acuerdo en los puntos de interés para su desarrollo, fue vital la participación del equipo anteriormente mencionado, así como la clara definición en los objetivos previstos para esta fase inicial del ámbito del software.

3.3 Recursos.

La estimación del proyecto nunca será una ciencia exacta, pero con la combinación de buenos datos históricos, la experiencia que se vaya obteniendo con el transcurso del tiempo y métodos de estimación se puede mejorar la misma.

Con la selección de los estudiantes y profesores, como se explica en capítulos anteriores se ha logrado

hasta el momento muy buenos resultados en cuanto a disciplina y cumplimiento de las actividades durante el desarrollo del proyecto, lo que ha llevado consigo un avance considerable en el mismo, cumpliendo con el tiempo pactado con el cliente.

Hasta el momento no se han reutilizado componentes por lo que se ha tenido que invertir más tiempo en la implementación del software.

Gracias a los recursos que disponen la universidad y en particular el proyecto tanto de hardware como de software, estos no fueron un problema, contribuyendo a un mejor desarrollo del proyecto.

Se puede resumir que el costo del proyecto ha sido el mínimo, pues solamente se han incurrido gastos en los viajes entre los países Cuba y Venezuela, mayormente en la etapa de despliegue. No se han incurrido gastos en cuanto a recursos humanos y tecnológicos, ya que el proyecto es realizado por estudiantes y profesores de la universidad (facultad 3), utilizando los recursos de hardware y software que brinda la misma.

Se cuenta con 37 computadoras de las cuales 6 se utilizan en el desarrollo del módulo presentación y 43 entre estudiantes y profesores de los cuales 6 estudiantes se encuentran trabajando el mismo.

3.4 Estimación del proyecto de software.

Aplicando el Modelo COCOMO II mencionado en capítulos anteriores se realizaron las estimaciones pertinentes al módulo de Presentación del Proyecto Informatización del Convenio Cuba – Venezuela. Esta estimación dio una idea del esfuerzo y el tiempo que se requiere para desarrollar el módulo de presentación.

Aplicando esta técnica de estimación permitió al planificador llevar a cabo el análisis de importantes variables del proyecto, tales como la fecha de entrega o la selección del personal para este módulo teniendo por delante restricciones del cliente que abogan utilizar este módulo sobre los finales de septiembre para la IX Comisión Mixta Cuba - Venezuela.

A partir de estos datos recopilados por el equipo de trabajo, el modelo implementado 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 variables bien difíciles hoy día en nuestra universidad antes que comience el proyecto: cuanto durará, cuanto esfuerzo requerirá y cuanta

gente estará implicada.

Teniendo en cuenta que se dispone de 6 computadoras y 6 estudiantes dispuestos a trabajar en el módulo presentación (recursos), y dado el esfuerzo que se requiere para el desarrollo del módulo (18.77 meses / hombre), se necesitan aproximadamente 4 meses para su culminación.

3.5 Análisis y Gestión del Riesgo.

Se detectaron los posibles riesgos que perjudicarían al proyecto por lo que se trazó un plan para la reducción y gestión del riesgo con el objetivo de mitigarlos. Con la aplicación del Plan RSGR se logró darle estabilidad al proyecto.

De los riesgos identificados en el Proyecto Informatización del Convenio Cuba – Venezuela, once de ellos se mitigaron, dos no han ocurrido y a los restantes en caso de manifestarse se le aplicaría el plan de contingencia, por no tener la certeza de su erradicación absoluta hasta la conclusión del proyecto.

3.6 Planificación y seguimiento del proyecto.

Usando la técnica Pert para la planificación temporal se logró distribuir el esfuerzo estimado y el personal de desarrollo que trabaja en el módulo de Presentación del Proyecto Informatización del Convenio Cuba – Venezuela en las diferentes tareas de la ingeniería de software relacionadas con el módulo.

Se generó una tabla con la fecha de inicio y final, temprana y tardía de cada una de las tareas ó actividades individuales lo que posibilitó la organización de la misma.

Hasta la actualidad se le ha dado seguimiento a las tareas mediante una serie de actividades descritas en el Epígrafe 2.3. Se han obtenido muy buenos resultados pues se han terminado las tareas en el tiempo estimado posibilitando que el módulo de Presentación, en el cual se está trabajando, se desarrolle en el tiempo previsto. Lógicamente esto tiene una holgura en cada actividad, determinada por la inexperiencia del equipo de desarrollo en la utilización de la arquitectura aprobada. Estos es uno de los riesgos a los que se hizo alusión anteriormente.

Aunque no se ha utilizado a plenitud la herramienta Trac, se le ha demostrado al equipo de desarrollo la gran ayuda en el seguimiento del proyecto, pues se puede tener un estricto control de las tareas a desarrollar. Se encuentra instalado en uno de los servidores que tiene a disposición el proyecto. También se le ha expuesto al equipo de desarrollo de todas las otras opciones que nos brinda esta herramienta, las cuales puede permitir una buena comunicación entre el personal de desarrollo; ver en

cada momento el estado del proyecto y tener toda la documentación del mismo a mano en el momento oportuno.

3.7 Conclusiones del capítulo.

Después de poner en práctica la estrategia de planificación propuesta para el Proyecto Informatización del Convenio Cuba – Venezuela existe una mayor organización y control de los entes. Los estudiantes se identifican más con el proyecto, hay una distribución equilibrada de los recursos, los tiempos asignados son justos, se han tenido en cuenta los riesgos, mitigándose cada unos de ellos.

Conclusiones generales.

Con la investigación realizada en el presente trabajo se arribaron a las siguientes conclusiones:

- A través del estudio del estado del arte de las estrategias de planificación y control a utilizar en proyectos productivos realizadas, se definió la organización de la planificación teniendo en cuenta las condiciones atípicas del proyecto Informatización del Convenio Cuba - Venezuela.
- Según el estudio que se hizo de los recursos que se disponían en el proyecto, se definieron los recursos humanos y tecnológicos necesarios para el desarrollo del mismo.
- Se seleccionó el método COCOMO II, lo que permitió hacer las estimaciones pertinentes en cuanto a esfuerzo y tiempo.
- Conjuntamente con el equipo de desarrollo y el líder de proyecto se analizaron los riesgos que pudieran de alguna forma afectar la planificación del proyecto, posibilitando la realización de un plan de contingencia para los mismos.
- De igual forma, a través de la investigación que se realizó de las herramientas que se utilizan para la planificación y seguimiento de proyectos de software, se seleccionó conjuntamente con el líder del proyecto y el equipo de desarrollo la idónea para esta faena. Esto posibilitó una planificación óptima de los recursos humanos y tecnológicos y darle un seguimiento apropiado.
- Con la aplicación de la estrategia definida se logró una mayor organización y control del trabajo realizado en el módulo Presentación del proyecto Informatización del Convenio Cuba – Venezuela. Se tuvo una visión del esfuerzo requerido para el desarrollo de la aplicación, así como el tiempo que duraría el mismo teniendo en cuenta los recursos que se disponían.

Recomendaciones.

- Es recomendable darle seguimiento y seguir perfeccionando la estrategia de planificación propuesta para su mejor rendimiento.
- Se recomienda al proyecto que se haga extensible la aplicación de esta estrategia a cada uno de los módulos restantes en aras de lograr una mejor planificación.
- Es recomendable la aplicación del Proceso de Software Personal (PSP), ya que proporciona métodos detallados de planificación y estimación personal.
- Es aconsejable la realización de encuentros en los que se debata y se explique en que consiste la planificación y cuan importante es en el desarrollo del proceso del software.

Referencia Bibliográfica.

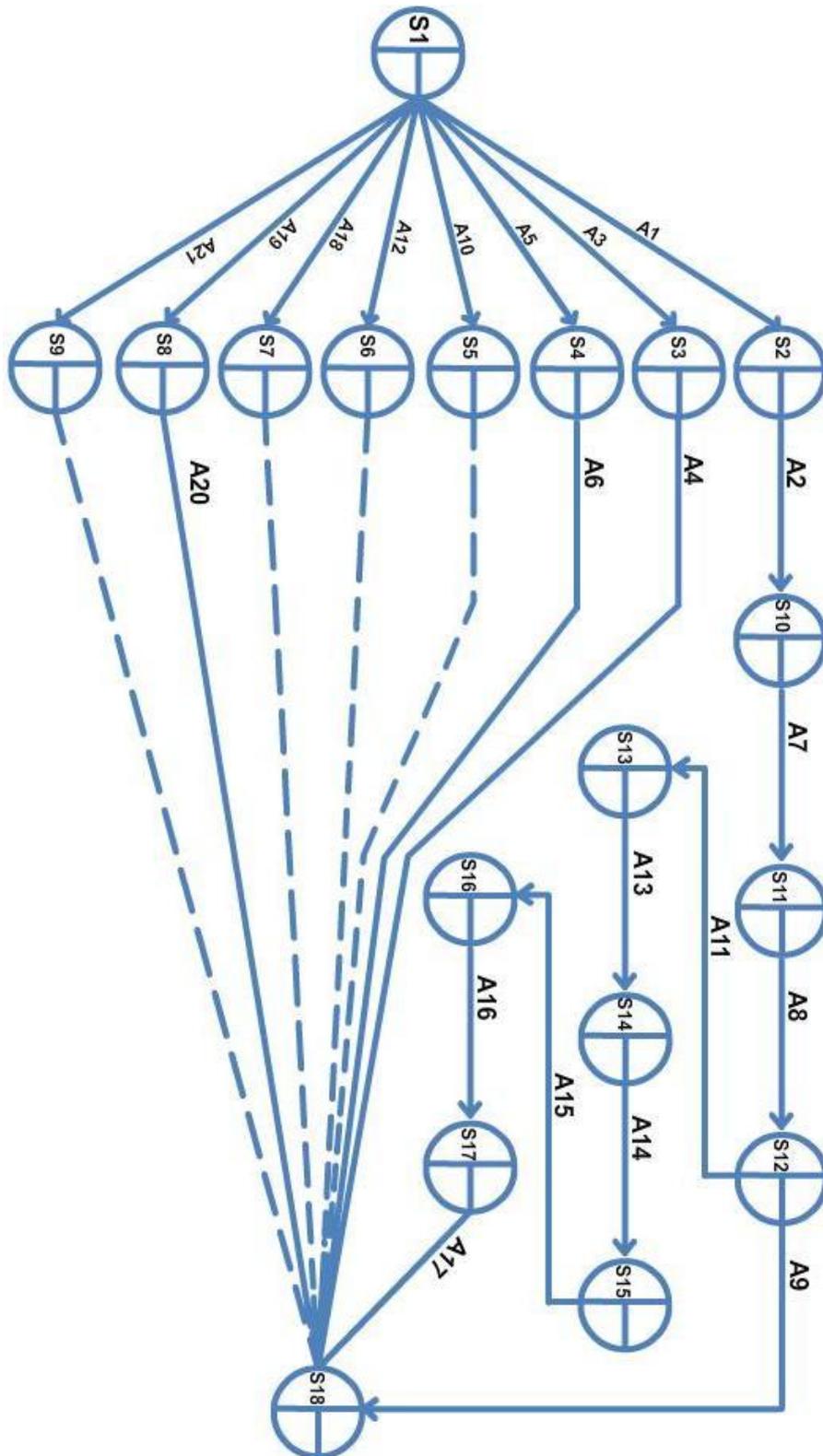
- [AbartiaTeam, 2007] AbartiaTeam. *GESTIÓN DE PROYECTOS CON DotProject*. [En Línea]. [Citado el: 7 de diciembre del 2007]. <http://www.abartiateam.com/dotproject.html>
- [Adrformacion, 2007] Adrformacion.com. *Gestión de Proyectos con Microsoft Project*. [Citado el: 4 de diciembre del 2007]. <http://www.adrformacion.com/udsimg/project/1/c03.htm>
- [Allan, 1983] Allan J. Albrecht and John E. Gaffney, "Software Function, Lines of Code, and Development Effort Prediction: A Software Science Validation", IEEE Transactions on Software Engineering, vol SE-9, No 6, November 1983.
- [Arquitectura, 2007] Arquitectura.com.ar. *GanttProject - Planificación de proyectos como Microsoft Project pero Gratis 100% Software Gratis para Planificación de Proyectos*. [En Línea]. [Citado el: 4 de diciembre del 2007]. <http://www.arquitectuba.com.ar/software-gratis/ganttproject/>
- [Barceló] Barceló M. La gestión de un proyecto informático. *Planificación, seguimiento, y aseguramiento de la calidad*.
- [Bennatan, 1992] Bennatan, E. M., (1992), *Software Project Management: A Practitioner's Approach*, McGraw-Hill.
- [Boehm, 1981] Boehm, B., *Software Engineering Economics*, Prentice-Hall. 1981.
- [Charette, 1989] Charette, R. N., (1989). *Software Engineering Risk Analysis and Management*, McGraw-Hill/Intertext.
- [Cortés, 1998] Cortés, H. (1998). *Gerencia Efectiva*. Caracas: HCZ Consulting.
- [Costa, 2007] Costa S. Hispanus.com. *La Planificación*. [En línea] [Citado el 20 de noviembre del 2007]. http://www.hispanus.com/areas/empresariales/empresariales_art03_01.html
- [GETEC, 2007] Grupo de Gestión de la Tecnología, Escuela Técnica Superior de Ingenieros de Telecomunicación Universidad Politécnica de Madrid, *Gestión de Proyectos: Planificación y control*, [En Línea]. (2007). [Citado el 12 de noviembre del 2007]. <http://www.gettec.etsit.upm.es/>
- [Graham, 1990] Graham C. Low and D. Ross Jeffery, "Function Points in the Estimation and Evaluation of the Software Process", IEEE Transactions on Software Engineering, vol 16, no 1, January 1990.
- [Higuera, 1995] Higuera, R.P, (Enero 1995) *Team Risk Management*, CrossTalk, US Dept. of Defense, P (2-4).
- [Hinojosa, 2003] Hinojosa M. GestioPolis.com. *Diagrama de Gantt*. [En Línea] marzo 2003 [Citado el: 4 de diciembre del 2007]. <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/diaggantaleja.htm>

- [Jiménez, 1982] Jiménez C., W. (1982). *Introducción al Estudio de la Teoría Administrativa*. México: FCE
- [Marcela, 2008] Marcela P. Varas C. *Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo*. [En línea]. Noviembre 1995. [Citado 12 febrero 2008]. <http://www.inf.udec.cl/~mvaras/papers/arica/arica.htm>
- [Moreno, 1998] Moreno Sánchez-Capuchino A. M, (1998). *Estimación de Proyectos de Software. COCOMO II*. [En Línea.]. [Citado 20 noviembre 2007]. http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/cocomo2k.pdf
- [Murdick, 1994] Murdick, R. (1994). *Sistemas de Información basados en computadoras*. México: Edit. Diana.
- [Navegapolis, 2007] Navegapolis.net. *Project Dune*. [En Línea]. 15 de agosto del 2007. [Citado el: 12 de febrero 2008]. <http://www.navegapolis.net/content/view/661/87/>
- [Navegapolis, 2008] Navegapolis.net, *Agile Track*, [En Línea] [Citado 13 febrero 2008] <http://www.navegapolis.net/content/view/321/78/>
- [Navegapolis, 2008] Navegapolis.net, *Estimación por casos de uso (2007)*, [En Línea]. [Citado 12 febrero 2008]. <http://www.navegapolis.net/content/view/563/87/>
- [OpenWorkbench, 2008] OpenWorkbench.org. *Open Workbench 1.1.4*. [En Línea]. [Citado el: 12 de febrero 2008]. <http://www.openworkbench.org/>
- [Pressman, 2005] Pressman, R. S., (2005), *Ingeniería del software: Un enfoque práctico*, 5ta Edición, Vol. 1.
- [Saavedra, 2007] Saavedra L. E., Opentelematics. *Gestión de Proyectos de desarrollo de Software (2007)*.
- [Stoner, 1996] Stoner, J. y otros. (1996). *Administración*. México: Prentice - Hall Interamericana.
- [TaskJuggler, 2008] TaskJuggler, *What is TaskJuggler?* [En Línea]. (2008), [Citado el: 12 de febrero de 2008]. <http://www.taskjuggler.org/>
- [Terry, 1987] Terry G. y Franklin S. (1987). *Principios de Administración*. México: CECSA.
- [The KOffice Project, 2007] The KOffice Project. *KPlato*. [En Línea]. 26 de marzo 2007 [Citado el: 12 de febrero de 2008]. <http://www.koffice.org/kplato/>
- [XPlanner, 2008] XPlanner.org. *XPlanner*. [En Línea]. [Citado el: 12 de febrero de 2008] <http://www.xplanner.org/>

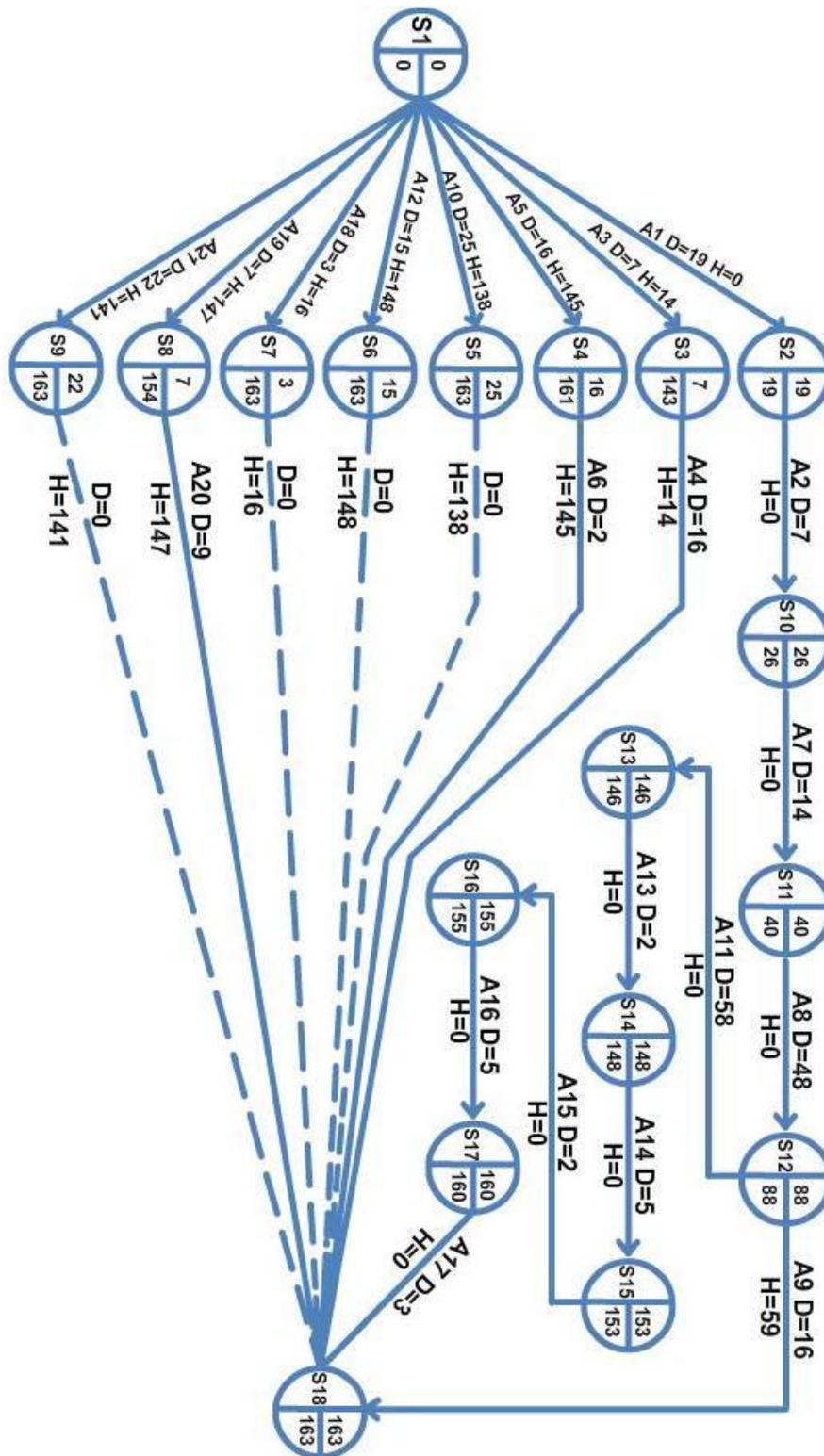
Bibliografía Consultada.

- Allan J. Albrecht and John E. Gaffney, "*Software Function, Lines of Code, and Development Effort Prediction: A Software Science Validation*", IEEE Transactions on Software Engineering, Vol SE-9, No 6, November 1983.
- Bennatan, E. M., (1992), *Software Project Management: A Practitioner's Approach*, McGraw-Hill.
- Hernández, R. Alfredo, Coello, Sayda, (2002), *El Paradigma Cuantitativo de la Investigación Científica*, Universidad de las Ciencias Informáticas.
- Humphrey, W. S., (1997), *Introducción al Proceso Software Personal (PSP)*.
- José Ignacio Cao, Eduardo Diez, Paola Britos y Ramón García-Martínez., Centro de Ingeniería de Software e Ingeniería del Conocimiento. Escuela de Postgrado. ITBA.
- Marcela P. Varas C. (Noviembre 1995) *Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo*. [En línea]. <http://www.inf.udec.cl/~mvaras/papers/arica/arica.htm>
- Peralta, Mario, *Estimación del Esfuerzo Basada en Casos de Uso*.
- Pressman, R. S., (2005), *Ingeniería del Software: Un enfoque práctico*, 5ta Edición, Vol. I y II.
- Trac. [En Línea]. <http://trac.edgewall.org/>
- UCI, Ingeniería de Software I, Conferencia 5, *Planificación y Estimación de Proyectos*.

Anexos.



Anexo 1: Grafo de Actividades. Pert (Módulo de Presentación)



Anexo 2: Cálculo de Holguras y Camino Crítico.