

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Propuesta de un procedimiento para la planificación
del Proyecto: Gestión Hospitalaria.**

Trabajo de Diploma para optar por el
título de Ingeniero Informático

Autora: Kirenia Nariño Thomas

Tutor: Ing. Paúl Pérez Zurita

Co-tutor: Ing. Eliéser Adrián Fuentes Alcolea

Ciudad de La Habana

Junio, 2007

***Somos lo que hacemos de forma repetitiva. La
excelencia, entonces, no es un acto, sino un hábito.
Aristóteles.***

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 30 días del mes de Junio del año 2007.

Kirenia Nariño Thomas

Ing: Paúl Pérez Zurita

Firma de la autora

Firma del tutor

Agradecimientos

- *A mis hermanos: Eduardo Carbonell Thomas y Coralía Nariño Thomas, por haberme apoyado en toda mi vida de estudiante, antes y después de haber perdido a nuestro padre. Les agradezco por todo, significan mucho para mí. ¡Son los mejores hermanos del mundo! ¡Los adoro!*
- *Mami, te agradezco tus consejos; tus llamadas y tu preocupación me dieron fuerza para seguir adelante. Tu amor infinito, es lo que más extrañé en la lejanía. Eres la razón de mí existir.*
- *A mi familia, a Vidal y a su esposa, por su gran apoyo incondicional.*
- *A mis vecinos y amigos, viejos y nuevos, a Cinthya, Lijebbitih, Yiyi, Maylén, Keila, Yuya, Nani, Idalmis, Yanelis, Niobis; Aleida, Cachita, Fifi, Nilda, Maité, Nidia, el Fuerte, Norbeys y Nápoles. ¡Yo tengo más que el leopardo, porque tengo buenos amigos!*
- *Yenny, agradecerte que me hayas soportado los cinco años es poco, nuestra amistad va más allá de estas líneas.*
- *A Malay; eres, sinceramente, muy especial.*
- *A todos los profesores que dieron su apoyo en mi formación profesional, y en la realización de este trabajo. Siempre los tendré presente.*
- *A nuestro Comandante Fidel Castro Ruz por haber confiado en nosotros hace cinco años.*

Dedicatoria

A mi padre: Reinaldo Nariño Cuevas.

Papi; sé que estarías orgulloso. Aunque no estés físicamente, te llevo por siempre en mi corazón.

RESUMEN

En el proceso de desarrollo del proyecto productivo Gestión Hospitalaria (GeHos), que se implementa en la Facultad 7 de la Universidad de las Ciencias Informáticas; existen inconformidades con la realización y aplicación de la planificación, impidiendo definir el tiempo óptimo en su entrega. Por lo que, el presente trabajo tiene como objetivo proponer un procedimiento para realizar la planificación en el proyecto productivo GeHos. Se tomó como muestra el módulo Bloque Quirúrgico General de este proyecto.

Se emplean como herramientas Microsoft Office Project, para realizar los diagramas de red y de Gantt de las actividades; el controlador de versiones, Subversion para supervisar y controlar las mismas. El método de puntos por casos de uso, para estimar el tiempo y esfuerzo del desarrollo. El procedimiento propuesto fue aplicado en la planificación del módulo mencionado, comprobándose su validez.

Se desea que este se aplique en los restantes módulos y que sea extendido a diferentes proyectos de la Facultad 7 y la Universidad de las Ciencias Informáticas. Con su aplicación, se obtienen resultados satisfactorios al optimizar el tiempo de entrega del software, mejorando la organización del proceso de desarrollo y se aprovechan mejor los recursos asignados.

PALABRAS CLAVE

Planificación, estimación, planificación temporal, riesgos.

ÍNDICE

RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Conceptos básicos de planificación	5
1.2 Objetivo de la planificación	7
1.3 Características	7
1.4 Importancia	8
1.5 Herramientas utilizadas a nivel mundial	9
1.6 Herramientas utilizadas en la Universidad.....	13
1.6.1 Herramientas utilizadas para el control y seguimiento de las tareas	14
CAPÍTULO 2: PLANIFICACIÓN DEL MÓDULO BLOQUE QUIRÚRGICO GENERAL	15
2.1 Módulo Bloque Quirúrgico General.....	16
2.1.1 Ámbito.....	19
2.2 Recursos	20
2.2.1 Personal.....	20
2.2.2 Recursos de software reutilizables	22
2.2.3 Recursos de entorno.....	23
2.3 Estimación del Proyecto Software	24
2.3.1 Técnicas empíricas de estimación.....	25
2.3.2 Métricas de software.....	28
2.4 Estimación del esfuerzo del módulo BQG mediante el método: Puntos de casos de uso	32
2.5 Análisis y gestión de los riesgos	43

2.5.1 Definición y Clasificación	43
2.5.2 Componentes y controladores de riesgos	44
2.5.3 Actividades para el análisis y la gestión de riesgos.....	45
CAPÍTULO 3: PLANIFICACIÓN TEMPORAL Y SEGUIMIENTO DEL PROYECTO.....	54
3.1 Objetivo	54
3.2 Conceptos básicos.....	54
3.3 Principios de la planificación temporal.....	55
3.4 Métodos de Planificación temporal	55
3.4.1 Red de tareas	57
3.4.2 Diagrama de GANTT	58
3.5 Método PERT: Principios básicos.....	59
3.6 Método de la ruta crítica	64
3.7 Propuesta	65
3.8 Ventajas de la planificación	66
CONCLUSIONES	67
RECOMENDACIONES.....	68
REFERENCIAS BIBLIOGRÁFICAS	69
BIBIOGRAFÍA.....	70
ANEXO 1	72
ANEXO 2	73
ANEXO 3	74
GLOSARIO DE TÉRMINOS	75

INTRODUCCIÓN

La informática es la ciencia que permite el flujo automático de la información, mediante las computadoras y sus diferentes componentes de entrada y salida. El uso de las computadoras, ha ido evolucionando con el transcurso de los años, uno de los más frecuentes es la creación de software, con el objetivo de automatizar las actividades en diferentes sectores económicos y sociales.

A Cuba le ha sido impuesto un injusto bloqueo que ya dura más de 40 años, aunque no ha podido destruir la Revolución; actúa sobre el desarrollo científico y tecnológico que se ha visto frenado por sus consecuencias. No obstante, se han logrado avances incuestionables, comparables con países desarrollados.

En diferentes sectores, político, económicos y sociales, nuestro país ha logrado buenos resultados. Poco a poco, se han introducido en las diferentes empresas el uso de las computadoras, con fines económicos y en aras de un perfeccionamiento profesional. El avance tecnológico se demuestra en gran medida en los eventos de Informática que se realizan en nuestro país.

Uno de los más grandes proyectos de Cuba, es la creación de la Universidad de las Ciencias Informáticas (UCI), donde estudiantes de todos los municipios del país, tienen la posibilidad de graduarse como ingenieros informáticos. Las principales funciones de esta universidad, son difundir a cada rincón los conocimientos adquiridos en pos del desarrollo informático y crear software en diferentes áreas temáticas, contribuyendo al desarrollo económico del país.

La Facultad 7 de la UCI, asume la tarea de organizar proyectos productivos siguiendo como área temática la Salud. Entre sus diferentes proyectos se encuentran los Sistemas de Información Hospitalaria (SIH).

Dentro de los SIH está insertado el proyecto **GeHos: Gestión Hospitalaria**. Este proyecto consta con la participación de estudiantes desde segundo hasta quinto año de la carrera y profesores de la facultad. Su estructura está dividida por módulos en los cuales se estudian y desarrollan los procesos que se ejecutan en las diferentes áreas de un hospital. Estos módulos, permiten realizar un trabajo bien definido, analizando cada una de las actividades que allí se efectúan.

En el equipo de un módulo, cada integrante cumple una función, en dependencia al rol que se le asigne, que puede ser: el rol de programador o implementador, el de diseñador de la base de datos y el de

analista. Aunque existen otros roles, como el de planificador, que es más amplio, ya que funciona de forma general para todo el proyecto.

Para poder realizar con éxito un proyecto de software, se necesita tener un control riguroso sobre el tiempo, las personas y los imprevistos que puedan surgir, como por ejemplo cambios en el software.

La planificación de los proyectos debe estar determinada por un notable grado de agilidad y dinamismo, no es razonable planificar un proyecto y pensar que esa planificación es ya definitiva e inmutable. En casi todos los casos, la realidad no coincide exactamente con lo previsto, por lo que es necesario ir haciendo ajustes periódicos.

La planificación es una herramienta para la gestión y la toma de decisiones, no para imaginar en un primer momento una evolución que posteriormente el tiempo se encargará de demostrar que estaba equivocada. Aunque existen técnicas de planificación muy avanzadas y elaboradas, la adecuada planificación consiste, en una actitud de anticipación que no es, sino una evidente manifestación del sentido común.

La realización de este trabajo está basada en las funciones que debe efectuar el planificador dentro de un proyecto productivo. El planificador puede ser una persona, a la que se le haya asignado esta tarea dentro del equipo de proyecto, el jefe del proyecto, o tantas personas se estimen convenientes. La persona asignada debe realizar las estimaciones lo más precisa posible y tener alguna experiencia de proyectos anteriores.

Asombrosamente, la planificación no es un tema muy conocido en el mundo de la ingeniería. Lo que trae consigo la falta de información para su aplicación a los diferentes ciclos de desarrollo del software.

El proceso de planificación en los proyectos productivos para el desarrollo de software, actualmente no se lleva a cabo o por el contrario se realiza de una forma inadecuada. Esto provoca, en gran medida, resultados insatisfactorios, tanto para la empresa que desarrolla el software como para el cliente. Aunque la estimación de costo y esfuerzos, es una de las tareas más difíciles en la gestión de un proyecto, es necesaria y muy importante su realización, para que el tiempo de desarrollo sea óptimo y se obtenga un producto con la calidad requerida.

Existe una vaga idea acerca de la planificación, muchos piensan que es innecesaria y en otros casos que es una pérdida de tiempo; trayendo esto como consecuencia, el uso inadecuado de los recursos asignados para el desarrollo del proyecto. Puede darse el caso, que al no alcanzar resultados visibles,

surja el descontento de los clientes y que el costo sea desfavorable para la organización que lleva a cabo el proyecto.

Resulta difícil realizar estudios técnicos, de mercado, financieros y de rentabilidad, así como la estimación de recursos necesarios y los costos generados para establecer la viabilidad de un proyecto, sin antes haber realizado un plan profundo del objetivo a alcanzar. Esto trae consigo, que no se produzcan elementos en los que se pueda apoyar la empresa para decidir si se realiza o no el proyecto.

Al no existir una previa planificación, no se obtiene un documento donde se especifican las actividades a realizar en el proyecto, impidiendo el control sistemático para saber en qué estado se encuentra el mismo. Lo que trae consigo, un relajamiento por parte de los desarrolladores incumpliendo con el tiempo de desarrollo estimado.

En todas las actividades productivas, se pueden presentar riesgos que influyen negativamente en su desarrollo. Al no elaborar la planificación se provoca un incremento de estos, impidiendo que se puedan detectar en tiempo, para su eliminación o supervisión.

Una vez identificada la situación problemática, se define el siguiente **Problema científico**: ¿Cómo realizar la planificación del proyecto GeHos de la Facultad 7 de la Universidad de las Ciencias Informáticas?

El problema se centra en el siguiente **Objeto de estudio**: La planificación en el proyecto GeHos de la Facultad 7 de la UCI.

El **Objetivo general** que se persigue es: Proponer un procedimiento para la planificación del proyecto GeHos de la Facultad 7 de la UCI.

El proceso de la planificación del módulo Bloque Quirúrgico General (BQG) del proyecto SIH, de la Facultad 7 en la UCI, se define como **Campo de acción**.

Se definen las siguientes **Tareas de investigación**; con el objetivo de dar cumplimiento a lo planteado anteriormente:

- Definir los métodos de estimación, de tiempo y de recursos.
- Identificar los posibles riesgos que existen en la realización del sistema.
- Definir las diferentes herramientas y tecnologías a utilizar. Establecer una comparación entre estas herramientas y proponer la más efectiva.

- Realizar una planificación temporal del módulo bloque quirúrgico general.

La investigación realizada está sostenida en 3 capítulos que se describen a continuación.

Capítulo 1: Fundamentación teórica. En este capítulo se realiza el estado del arte de la investigación, donde se esclarecen los conceptos fundamentales de la planificación, sus características y su importancia.

Capítulo 2: Planificación del Módulo Bloque Quirúrgico General. Se centras las bases de una propuesta de planificación en el módulo BQG.

Capítulo 3: Planificación temporal y seguimiento. Se explica la vía propuesta para realizar el control de las tareas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo, se estudian los fundamentos teóricos relacionados con la planificación del proceso de desarrollo de un software. Se estudian las diferentes herramientas que existen y las que se utilizan en el entorno de producción de la Universidad de las Ciencias Informáticas para conocer el resultado obtenido de la planificación temporal.

1.1 Conceptos básicos de planificación

La planificación es una actividad adjunta a la condición humana y que generalmente consiste en la organización justa de la acción y los recursos para tratar de conseguir unos objetivos o una meta deseada. La planificación está presente en todos los aspectos de la vida y la realizan consciente e inconscientemente, todos los individuos en su existencia cotidiana.

Cuando la planificación se extiende a las organizaciones, su principal objetivo es tratar de responder desde sus orígenes el dilema de dar el “*mejor*” uso a unos recursos que son escasos.

En el transcurso de los años ha evolucionado tanto la forma de concebir la planificación como sus fines, métodos y características. Las diferentes definiciones de planificación que se han elaborado en el contexto de desarrollo muestran las fabulosas percepciones que se han elaborado sobre la misma y enfatizan disímiles aspectos. Se pueden encontrar las siguientes definiciones, entre otras:

“Planificar puede significar en general cualquier intervención voluntaria en que se definen objetivos, se especifican los recursos necesarios, y se presentan un modelo causal que liga las actividades con los objetivos.” (LOMA-OSORIO)

“La planificación es el procedimiento mediante el cual se seleccionan, ordenan y diseñan las acciones que se debe realizar para el logro de determinados propósitos, procurando una utilización racional de los recursos disponibles” (LOMA-OSORIO)

“Planificar es la acción consistente en utilizar un conjunto de procedimientos mediante los cuales se introduce una mayor racionalidad y organización en un conjunto de actividades y acciones articuladas entre si que, previstas anticipadamente, tienen el propósito de influir en el curso de determinados acontecimientos, con el fin de alcanzar una situación elegida como deseable, mediante el uso eficiente de medios y recursos escasos o limitados” (LOMA-OSORIO)

"Planificar es el proceso de elección y selección entre cursos alternativos de acción, con vista a la asignación de recursos escasos, con el fin de obtener objetivos específicos sobre la base de un diagnóstico preliminar que cubre todos los factores relevantes que pueden ser identificados" (LOMA-OSORIO)

"Es el proceso de establecer metas y elegir medios para alcanzar dichas metas" (ARMAS 2006)

"Es el proceso que se sigue para determinar en forma exacta lo que la organización hará para alcanzar sus objetivos" (ARMAS 2006).

"Es el proceso de evaluar toda la información relevante y los desarrollos futuros probables, da como resultado un curso de acción recomendado: un plan", (ARMAS 2006).

"Es el proceso de establecer objetivos y escoger el medio más apropiado para el logro de los mismos antes de emprender la acción", (ARMAS 2006).

"La planificación... se anticipa a la toma de decisiones. Es un proceso de decidir... antes de que se requiera la acción" (ARMAS 2006).

"Consiste en decidir con anticipación lo que hay que hacer, quién tiene que hacerlo, y cómo deberá hacerse". Se erige como puente entre el punto en que nos encontramos y aquel donde queremos ir. (ARMAS 2006)

"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" (ARMAS 2006).

"Es el proceso consciente de selección y desarrollo del mejor curso de acción para lograr el objetivo." Implica conocer el objetivo, evaluar la situación considerar diferentes acciones que puedan realizarse y escoger la mejor. (ARMAS 2006).

"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" (ARMAS 2006).

"Planificación. f. Acción y efecto de planificar. || 2. Plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado, tal como el desarrollo armónico

de una ciudad, el desarrollo económico, la investigación científica, el funcionamiento de una industria, etc.(MICROSOFT 1993-2006).

1.2 Objetivo de la planificación

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

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

1.3 Características

Algunas de las características más importantes de la planificación son:

La planificación es un proceso permanente y continuo: no se agota en ningún plan de acción, sino que se realiza continuamente en la empresa. Está siempre orientada hacia el futuro: la planificación se halla 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 opciones, constituye un curso de acción escogido entre varias alternativas de caminos potenciales, es sistemática, tener en cuenta el sistema y subsistemas que la 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 de 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 elaborándose, 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 de varias actividades para conseguir los objetivos previstos.

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.

1.4 Importancia

La planificación es importante porque permite decidir cuánto tiempo se necesita para hacer un trabajo y controla el progreso mientras se está haciendo el mismo. Los ingenieros la utilizan para entender el estado de su proyecto. Los planes detallados indican por dónde se encuentra el proyecto frente al plan. Permite organizar mejor el tiempo, y evitar las crisis en los últimos minutos. En un trabajo en equipo; los ingenieros necesitan planificar su trabajo personal. Los planes individuales del producto, les permite cumplir con las fechas para cada una de sus tareas y cumplir con sus compromisos de forma consistente.

En los negocios, también es importante la planificación y gestión del trabajo. Un plan bien hecho incluye una estimación del coste del proyecto. Esto es necesario para contratar desarrollos, debido a que los clientes necesitan saber el precio por adelantado. Las estimaciones también son necesarias cuando se desarrollan productos. El costo del proyecto es la parte más importante del precio de un producto y debe ser lo bastante bajo para que el precio sea competitivo en el mercado.

Puesto que la planificación es tan importante, es necesario que se gestionen y planifiquen todos los proyectos y actividades más significativas. Esto ayudará a decidir el tiempo que se necesita y la fecha en que se culminará, así como llevar, el control de lo que se está haciendo, y ver si se ha avanzado o no.

En los momentos actuales, la mayor parte de las organizaciones reconocen la importancia de la planificación para su crecimiento y bienestar a largo plazo. Se ha demostrado, que si los gerentes definen eficientemente la misión de su organización, estarán en mejores condiciones de dar dirección y

orientación a sus actividades. Las organizaciones funcionan mejor gracias a ello y se tornan más sensibles ante un ambiente de constante cambio.

La planificación, es una actividad de alto nivel, en el sentido de que la alta gerencia debe participar activamente ya que desde su punto de vista más amplio, tiene la visión necesaria para considerar todos los aspectos de la organización. Además, se requiere una adhesión de la alta dirección para obtener y apoyar la aceptación en niveles más bajos.

Propicia el desarrollo de la empresa al establecer métodos de utilización racional de los recursos. Reduce los niveles de incertidumbre que se pueden presentar en el futuro, más no los elimina. Prepara a la empresa para hacer frente a las contingencias que se presenten, con las mayores garantías de éxito. Mantiene una mentalidad futurista teniendo más visión del porvenir y un afán de lograr y mejorar las cosas. Reduce al mínimo los riesgos y aprovecha al máximo las oportunidades. Promueve la eficiencia al eliminar la improvisación. Minimiza el trabajo no productivo y se obtiene una identificación constructiva de los problemas y las potencialidades de la empresa.

Por último, cabe destacar que la planificación es la primera función del proceso administrativo, por tanto, realizar una buena planificación conlleva a tener una buena organización, dirección y control de la empresa lo cual se traduce en una administración cien por ciento efectiva.

1.5 Herramientas utilizadas a nivel mundial

Mundialmente existe un sin número de herramientas para la realización de la planificación de actividades. En la investigación realizada, se puede ver a continuación que la herramienta más utilizada es Microsoft Office Project, aunque se ponen ejemplos de otras.

Microsoft Office Project 2003: La solución de Microsoft Office para la Administración de Proyectos. Es una herramienta esencial para todo aquel gerente de proyectos. Con Project 2003 Se podrá programar y organizar los recursos y las tareas, a fin de generar proyectos a tiempo y conforme a al presupuesto. Adicionalmente, se contará con la Guía Project que ayuda a dominar rápidamente el proceso, y a programarlo paso a paso, a controlar su progreso y reportar información del mismo.

La familia Microsoft Office Project 2003 esta formada por dos líneas de producto: Microsoft Project Standard 2003, y Enterprise Project Management (EPM), que es una solución empresarial compuesta por Microsoft Project Server 2003, Microsoft Project Professional 2003 y Microsoft Office Project Web Access.

Microsoft Project Standard 2003: Es una versión del programa central de administración de proyectos de Microsoft. Con herramientas conocidas sencillas de usar, Project Standard permite administrar proyectos de forma independiente en el escritorio. Con Project Standard, puede planear, administrar y comunicar información de los proyectos con más eficacia que antes.

Enterprise Project Management (EPM): Es la solución idónea para las organizaciones que necesiten un alto nivel de coordinación y estandarización entre los proyectos y los administradores de proyectos, administración centralizada de los recursos o un mayor nivel de creación de informes acerca de los proyectos y los recursos. La solución EPM de Microsoft permite que organizaciones enteras, departamentos o equipos trabajen juntos en la administración eficaz de proyectos y procesos.

Microsoft Office Project Professional 2003: Project Professional 2003 es el programa de escritorio de Microsoft para la administración de proyectos empresariales. Project Professional proporciona todas las herramientas centrales de programación de Project Standard 2003, además de eficaces capacidades de administrar recursos si se conecta al Project Server 2003

Microsoft Office Project Server 2003: Proporciona un control centralizado de todos los proyectos desarrollados por la organización. A través de un repositorio gestionado centralmente en el servidor, permite a los equipos un acceso a la información mediante una interfaz Web. Permitiendo mantener la información convenientemente actualizada.

Microsoft Office Project Web Access: Es el portal Web que permite a los usuarios conectarse al proyecto y a la información de recursos en Project Server. Mediante el acceso Web cualquier miembro del equipo puede fácilmente consultar, analizar o modificar la información sin la necesidad de tener instalado Microsoft Project en su ordenador. Con solamente un navegador, un usuario puede acceder a las utilidades necesarias para gestionar toda la información del proyecto.

Visual Studio 2005 Team System: Descubre cómo realizar la gestión de proyectos directamente desde el entorno integrado de desarrollo, gracias a la última versión de Visual Studio. Brinda una serie de herramientas para la gestión de proyectos, basadas en los software ya conocidos: Microsoft Excel, Microsoft Project, Microsoft Word, y Windows Share Point Services. Mediante la integración de Microsoft Office, la gestión del proyecto no necesita tratar los datos de estas aplicaciones para los datos usados por el equipo de desarrollo. La carpeta integra los productos de trabajos en el Visual Studio IDE para el acceso eficaz del equipo.

Microsoft Solutions Framework (MSF): Está diseñado para proporcionar una guía básica sobre cómo desarrollar con éxito un proyecto software, es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos. Se ha convertido posteriormente en un modelo práctico que facilita el éxito de los proyectos tecnológico.

Existe una tendencia a un nuevo (ya no tanto) Sistema Operativo, llamado Linux. El objetivo principal de este sistema es propulsar el software de libre distribución junto con su código fuente para que pueda ser modificado por cualquier persona, dando rienda suelta a la creatividad. También existen herramientas de uso libre para la gestión de proyectos:

Gantt PV: Es un programa gratuito, de apariencia sencilla y sin grandes complicaciones, para planificación de proyectos, descomposición, representación y seguimiento de tareas sobre diagrama de Gantt.

Gantt Project: Es una herramienta de planificación libre y fácil de usar. Basado en las tareas del Project y en el Diagrama de Gantt. Dentro de sus mejores características incluye:

Jerarquía de tareas y dependencia.

Diagrama de Gantt.

Carga diagrama de Recursos.

Diagramas generados de PERT.

Informes HTML y PDF.

Importa/Exporta a MS Project.

Es una aplicación de escritorio con interfaz similar a MS. Project permite programar y organizar las tareas y asignación de personas y recursos sobre una representación Gantt. Por supuesto es una herramienta mucho más ligera que MS Project, pero esto en el ámbito y dimensión de muchos proyectos es más una ventaja que un inconveniente.

Dotproject: Algo más veterana ésta solución en entorno Web, ofrece un marco completo para la planificación, gestión y seguimiento de múltiples proyectos para clientes diferentes, quienes pueden disponer también de acceso para monitorizar la evolución del desarrollo.

TeamWork: Impresionante es la apariencia de esta herramienta de entorno Web para registrar y gestionar los tiempos de diferentes equipos de trabajo en sus respectivos proyectos. Gestión completa de informes de tiempos y costes. Combina gestión de documentos, de equipos y de proyectos.

Planner: Aplicación de escritorio para gestión y seguimiento de proyectos, con descomposición en tareas y sub-tareas, dependencias, identificación de la ruta crítica, diagramas de Gantt. Inicialmente desarrollada para Linux, dispone de versión (beta) para Windows.

HojaExcelparaScrum: Hoja de cálculo para gestionar el trabajo en cada tarea, asignación, estado y tiempo. Genera de forma automática los gráficos para el seguimiento de esfuerzo y tareas.

AgileTrack: Herramienta para planificación y seguimiento de proyectos, de interfaz sencillo. Para desarrollo de software en equipos reducidos con metodologías ágiles, especialmente eXtreme Programming.

PPTS: Project Planning and Tracking System es una herramienta de gestión ágil de proyectos para equipos que trabajan con Scrum y/o Extreme Programming. Es un sistema Web, accesible con un navegador que puede instalarse sobre servidor Linux o Windows (con php y MySQL) y de uso libre, con licencia GNU (Licencia Publica General en español).

XPWeb: Plataforma Web para gestión de proyectos con Extreme Programming.

TUTOS: (The Ultimate Team Organization Software) es una herramienta Web de código abierto y uso gratuito para la gestión de pequeños grupos de trabajo o departamentos que incluye las siguientes funcionalidades: calendario, gestión de equipos, directorio de personas, gestión de incidencias, registros de tiempo, listas de seguimiento. (*TUTOS 18.07.2006*)

Solodox: Servicio de software que permite editar y compartir con el equipo y demás interesados planificaciones Gantt. Su versión alfa solo funciona sobre Explorer. (*Para diagrama de flujos y presentaciones 2007*)

ToDoList: Es una herramienta gratuita muy simple y efectiva para la gestión de proyectos en entornos ágiles. Escasamente ocupa 1 Megabyte, y al instalarla se puede indicar que emplee un fichero .ini para guardar la información de configuración, de forma que no toca para nada el registro de Windows y se puede llevar incluso en una memoria USB. (*ToDoList* 2007)

ClockingIT: Es un gestor de proyectos y tareas, con control de tiempos, generador de informes, repositorio de ficheros, agenda, chat, y notificaciones.

1.6 Herramientas utilizadas en la Universidad

En la Universidad de las Ciencias Informáticas, la herramienta mas utilizada en la mayoría de los proyectos productivos de las distintas facultades es **Microsoft Office Project**. Es la herramienta estudiada en la asignatura de Ingeniería del Software impartida en clases.

Project es la solución que ofrece Microsoft Office para la administración de proyectos. Es la herramienta esencial para todo aquel gerente de proyectos. Posibilita la programación y organización de los recursos y las tareas a tiempo, conforme al presupuesto.

Project, en todas sus versiones, es muy útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBOK (Management Body of Knowledge) del PMI (Project Management Institute). Basado fundamentalmente en el Método de la Ruta Critica; CPM; (Critical Path Method) y la Técnica de Revisión y Evaluación de Proyectos; PERT; (Program Evaluation and Review Technique), utilizando el Diagrama de GANTT, conjugando las tres técnicas para trabajar.

Entre muchas posibilidades de trabajo, MS Project permite: Planificar y programar tareas así como asignar recursos a dichas tareas de forma 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 apropiados y fáciles de interpretar. Establecer escenarios dentro del proyecto para crear análisis 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.

Project tiene una extensa y buena documentación en la mayoría de los sitios de los buscadores de información de la Internet. Es importante aclarar que existen documentos para aprender a trabajar en el mismo, que aunque el tema de la planificación no es el de mayor auge en la realización de software

productivos, se está trabajando y analizando por las diferentes empresas que se dedican a la producción de software, y como se puede ver en la UCI es un Rol a implementar dentro de un grupo de desarrolladores de software.

1.6.1 Herramientas utilizadas para el control y seguimiento de las tareas

En algunos proyectos de las facultades de la UCI se puede apreciar que el control y seguimiento de las tareas, se lleva a cabo, mediante reuniones e informes realizados con los estudiantes. Aunque, también se apoyan en algunas herramientas, que son más bien, para el control diario de las actividades. A continuación las herramientas más utilizadas.

Subversion: Es un sistema de control de versiones libre y de código fuente abierto. Puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. (*Subversion*)

Tortoise y AnkhSvn: Cliente para el sistema de control de versiones. Plug-in del sistema de control de versiones SubVersion que se integra al VS2005.

Trac: Trac es una herramienta código abierto (open source) de uso libre de interfaz Web, simple y minimalista que integra herramientas para comunicación, gestión, seguimiento de proyecto; y gestión de la configuración (*Trac: Gestión de proyectos y de configuración integrados 2006*).

En estos proyectos, el plan, se realiza, por puro concepto, pero no se trabaja en base al mismo. Los proyectos llevan un largo período de tiempo de realización con muy poco resultados.

En este capítulo, se realizó una breve panorámica, sobre la planificación. Sus principales características e importancia. Se presentaron algunos conceptos de planificación desde diferentes puntos de vista, donde todos coinciden, que su objetivo principal; es tener una visión clara de lo que se va a realizar, para poder controlar estas actividades y poder estimar el tiempo, costo y esfuerzo de desarrollo.

También, se trató el tema de la planificación a nivel mundial y en la Universidad de las Ciencias Informáticas, haciendo referencia a las diferentes herramientas que se utilizan para el mejor proceso de la planificación.

CAPÍTULO 2: PLANIFICACIÓN DEL MÓDULO BLOQUE QUIRÚRGICO GENERAL

La medición es fundamental en cualquier disciplina de ingeniería; y la ingeniería del software no es una excepción. Es un elemento clave de cualquier proceso de Ingeniería ya que emplea medidas para entender bien los atributos de los modelos que se crean.

Las métricas del producto son medidas cuantitativas que permiten tener una visión de la eficacia del proceso del software. Se reúnen los datos básicos de calidad y productividad. Estos datos son analizados, comparados con promedios anteriores, y evaluados para determinar las mejoras en la calidad y la productividad. También son utilizadas para señalar áreas con problemas de manera que se puedan desarrollar un plan para mejorar el proceso del software.

Las métricas del software son analizadas y evaluadas por los administradores del software. A menudo las medidas son reunidas por los ingenieros del software. Si no mides, sólo podrás juzgar basándote en una evaluación subjetiva. Mediante ella se pueden señalar las tendencias (buenas o malas), realizar mejores estimaciones, y llevar a cabo una mejora sobre el tiempo.

Se comienza realizando un conjunto limitado de medidas de procesos, proyectos y productos que sean fáciles de recoger. Estas medidas son a menudo normalizadas utilizando métricas orientadas al tamaño o a la función. El resultado se analiza y se compara con promedios anteriores de proyectos similares realizados en la organización. Se evalúan las tendencias y se generan las conclusiones. Obteniendo un conjunto de métricas del software que proporcionan una visión profunda del proceso y de la comprensión del proyecto.

Aplicando un plan de medición sencillo pero consistente, que nunca se utilice para evaluar, premiar o castigar el rendimiento individual, puedes estar seguro que lo que has hecho, está correctamente.

Después de varias horas, reunidos los líderes del proyecto GeHos, se llegó a la conclusión de realizar la medición del proyecto, utilizando el método de estimación, *Puntos por Casos de Uso*.

Este método de estimación se desarrolló en el año 1993 para poder finalmente obtener estimaciones de esfuerzo sobre productos de software orientados a objetos. Existen otros métodos para cálculo y estimaciones factibles de ser utilizados en proyectos que utilicen orientación a objetos, pero como el método de Puntos de Caso de Uso se definió como extensión del Punto de Función, se decidió definirlo como estándar del proceso, de acuerdo a las siguientes consideraciones:

- La alternativa es simple y similar a los Puntos de Función.
- Las otras alternativas no demostraron tener aceptación general ni se han utilizado ampliamente.
- Algunas alternativas son propietarias.
- No existen ejemplos de distintas fuentes respecto al uso de otros métodos.
- Algunas herramientas presume usarlas exclusivamente para la tarea de cálculo.
- La metodología del desarrollo del proyecto es GeHos, donde es fundamental la realización de los casos de uso.

El Proyecto está dividido por diferentes módulos, los cuales cumplen las necesidades de los hospitales. El trabajo para la realización de la tesis se enfocará en el módulo Bloque Quirúrgico General (BQG).

El módulo de BQG es el que realiza la gestión de la información que se genera en un hospital o institución, durante la intervención quirúrgica. El cual tiene como principales funcionalidades registrar la información referente a un paciente durante todo el proceso preoperatorio, transoperatorio y postoperatorio.

Sin lugar a dudas el BQG, es uno de los más necesarios en un Sistema de Información Hospitalaria, cuentan con el mayor peso del trabajo a realizar en los hospitales. Por estas razones se decidió, realizar la estimación del tiempo en este módulo.

2.1 Módulo Bloque Quirúrgico General

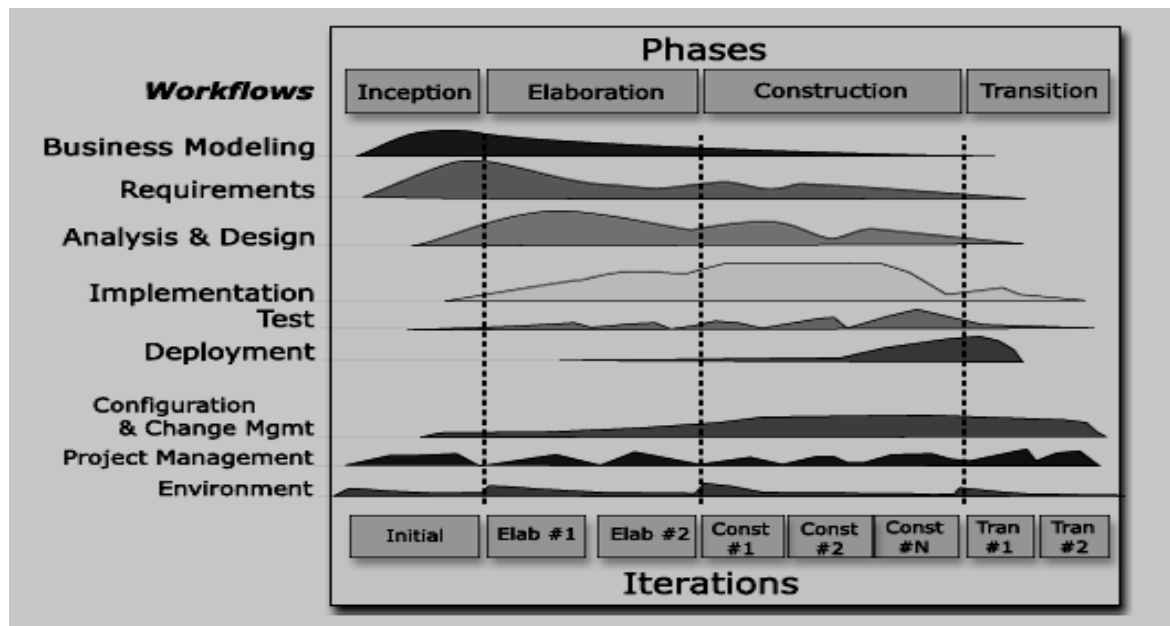
La informatización de la salud, es uno de los temas más abordados en diferentes conferencias, reuniones y eventos que se realizan. Agilizar el trabajo de las diferentes áreas de los hospitales, sería un avance en cuanto al tema se refiere. Permitiendo así, la eficiencia y eficacia del trabajo mediante un entorno agradable tanto para los pacientes como para el personal del hospital.

Realizar un sistema, efectivo, para la salud, informatizando todas las actividades que se realizan en los hospitales, desde el mismo momento en que llega un paciente a una consulta, teniendo en cuenta todo el proceso que se debe realizar en dependencia de las diferentes áreas que existen y de lo diagnosticado al paciente en dicha consulta, es la idea en la que gira el proceso de desarrollo del software en cuestión.

El producto obtenido debe recibir los datos del paciente mediante una interfaz de usuario, estos datos se guardan en un repositorio de información, donde se puede insertar, eliminar y modificar los datos. Además de que debe realizar una búsqueda de información rápida.

Existen diferentes metodologías de desarrollo del software. La metodología más utilizada a nivel mundial, es RUP (Rational Unified Process). Principalmente se utiliza para proyectos grandes, de bastante personal involucrado. Divide el proceso de desarrollo del software en cuatro fases: Inicio, Elaboración, Implementación o Construcción y Transición o Prueba.

Estas fases proporcionan a los desarrolladores del software un mejor trabajo y un eficaz rendimiento en el mismo. RUP propone nueve flujos de trabajo, los seis primeros son conocidos como flujos de ingeniería y los tres últimos como apoyo. En la figura se muestra el proceso en dos dimensiones, horizontalmente se encuentran las fases, y verticalmente los flujos de trabajo.



La administración del proyecto es el flujo número ocho, en la figura anterior se puede ver cómo este flujo está presente en todas las fases del proyecto. Este flujo realiza las actividades para producir el producto, que compensen las necesidades del cliente, las cuales son programadas y estimadas por el planificador y el líder del proyecto.

Utilizando las vistas arquitectónicas definidas por la metodología RUP, se muestran las diferentes características del sistema. Vista de Casos de Uso, Vista Lógica, Vista de Implementación y la Vista de Despliegue, dichas vistas son modeladas utilizando Rational Rose 2003 Enterprise Edition.

En este proyecto se propone el diseño de una arquitectura que permita la integración entre los sistemas de gestión de información hospitalaria desarrollados en el proyecto Hospitales.

El proyecto está dividido por siete módulos:

Inscripción y Admisión: Este módulo permitirá a los empleados de este departamento hospitalario, automatizar todos los procesos que se efectúan de manera manual, dígase el proceso de registro de ingresos, egresos y los movimientos hospitalarios gestionados, además de controlar por medio de la historia clínica del paciente todas las eventualidades del mismo dentro del hospital.

Bloque Quirúrgico General: Dentro de este módulo se especificó aparte, el módulo: **Bloque Quirúrgico Oftalmológico**. Este producto permitirá inscribir a un paciente creando la historia clínica oftalmológica de dicho paciente y registrar toda la información en una base de datos de la cual se pondrá mostrar la información existente en ella. Proporcionará la recogida y procesamientos de datos de las diferentes consultas y todo un sistema de seguridad basada en roles para la garantía de integridad y confidencialidad de la información. Permitirá además llevar un reporte estadístico del hospital.

Archivo: Un sistema de gestión de archivos es aquel sistema informático que brinda servicios a los usuarios en el uso de archivos, haciendo cumplir una serie de objetivos, como optimizar el rendimiento, proveer un soporte de entrada/salida y así tener control de los datos archivados y minimizar o eliminar la pérdida o destrucción de datos.

Farmacia: El software se encargará de la Gestión Farmacéutica en los Hospitales, este productos será integrado al Sistema de Información Hospitalaria, interactuando con los demás módulos existentes que componen dicho sistema, para gestionar la actividad farmacéutica que se genera en una institución hospitalaria.

Cuerpo de Guardia: Este módulo se encarga de gestionar toda la actividad que se realiza en esta sección. En la guardia de un centro hospitalario, también se debe gestionar todas las acciones que se realizan con el paciente que llegue a esta sección del hospital.

Banco de Sangre: Su principal objetivo es llevar un control de todos los análisis de los pacientes, almacenando sus componentes sanguíneos, para una futura operación o investigación.

Configuración y Seguridad: Es el encargado de de la gestión de todos los nomencladores, y del manejo de la seguridad y del personal que interactúa o colabora con el sistema.

2.1.1 Ámbito

Para realizar la planificación de un proyecto, la primera actividad del planificador, es definir el ámbito del software a realizar.

El ámbito del software tiene como objetivo, describir el control y los datos a procesar, la función, el rendimiento, las restricciones, las interfaces y la fiabilidad. Las funciones bien detalladas son importantes para la estimación, debido a que el coste y la planificación temporal están orientados a la función. Las especificaciones del rendimiento muestran las necesidades de respuestas, de tiempo y de procesamiento. Las restricciones definen los términos del software originados por hardware externos, por la memoria disponible o por otros sistemas existentes.

El ámbito es solo una guía, para que todo el proyecto en conjunto, decida si el software puede ser construido dentro de las dimensiones reflejadas.

2.1.1.1 Ámbito del módulo BQG.

El módulos BQG, realiza un software con el fin de controlar todas las informaciones quirúrgicas de los paciente, así como el registro de las estadísticas quirúrgicas y el material gastable del hospital, garantizando así el control y la seguridad de los datos.

El producto obtenido será un componente de un sistema de Informatización de Hospitales. Su desarrollo en Web y el uso de Web Service, será muy fácil para garantizar la comunicación con cualquier sistema de este tipo. Además de que podrá instalarse en cualquier sistema quirúrgico. Permitiendo así el acceso a la información desde cualquier lugar del mundo, creando los mecanismos de seguridad.

El software obtenido en este módulo debe inscribir a un paciente creando la historia clínica quirúrgica del mismo, y registrar toda la información en una base de datos de la cual se podrá mostrar la información existente en ella, brindándole un reporte estadístico del hospital. Además constará con una interfaz gráfica sencilla y amigable.

Para el uso factible del producto obtenido, se requiere de un servidor que tenga como características básicas: Sistema Operativo Linux o Windows y 1.5 G de memoria RAM. Para los clientes, deberán tener Windows o Linux como sistema operativo y 256 Mb de memoria RAM, pues el sistema será capaz de correr en ambas plataformas. El sistema resultante será una aplicación de escritorio soportada por

plataforma de Software Libre. Para el desarrollo del sistema se utilizará C#, con PostgreSQL como sistema gestor de bases de datos.

Estas características que debe poseer el producto fueron seleccionadas por el arquitecto del proyecto, consultadas y analizadas con todo el equipo de desarrollo; teniendo en cuenta las necesidades del cliente.

2.2 Recursos

Después de obtener el ámbito general del módulo, el planificador realiza, como segunda actividad, la valoración de los recursos necesarios para realizar la estimación del esfuerzo de desarrollo del software.

El personal que se necesita para realizar el software, es el primer recurso, que se debe estimar. Las personas deben estar bien capacitadas de los conocimientos básicos que se necesitan. En el caso de la universidad, los protagonistas de las actividades son los estudiantes conjuntamente con los profesores.

Otros recursos son: Los componentes de software. Los mismos deben ser válidos para una fácil integración de los módulos y estandarizados para una fácil aplicación. Si no se realiza una estimación previa del software reutilizables, estos se convierten en la preocupación primordial durante la fase de desarrollo del producto. En el proyecto, los softwares reutilizables son los mismos módulos y algunos estándares.

Por último, es necesario estimar el entorno de desarrollo, que son las condiciones que necesita el sistema para funcionar, como el tipo de programación, de proceso, y las características de las máquinas que lo componen.

Con la planificación de recursos se pretende determinar qué recursos serán necesarios, cuándo, cómo y dónde se obtendrán los que no están disponibles y en que forma serán generados o adquiridos.

2.2.1 Personal

Después de un análisis profundo del ámbito, se seleccionan las habilidades que se necesitan para llevar a cabo el desarrollo. Especificando así, la posición de cada persona dentro del proyecto. El total de personas que se necesitan, se calculan más adelante, después de hacer la estimación del esfuerzo de desarrollo.

Hasta ahora, en el proyecto GeHos; se ha trabajado con dos implementadores por módulos. A medida que han surgidos las necesidades del proyecto, se han reestructurado los módulos existentes. Lo que ha traído atraso en la implementación y realización de las actividades necesaria para el desarrollo del software.

Existe una nueva propuesta, para la planificación del personal del proyecto, con el objetivo de realizar una nueva estimación que se acerque más a la realidad del proyecto.

En estos momentos el módulo BQG cuenta con tres implementadores, tres analistas del sistema y dos diseñadores de la base de datos. Los cálculos de estimación realizados son con estos estudiantes.

Esta planificación no es la más óptima, pues el desarrollo del módulo está bastante completo, aunque se sugiere una nueva reestructuración del módulo y que se aplique a los restantes del proyecto. Debido a que, ya la parte del análisis y diseño, estará bastante adelantado, se necesitaría solo una persona en este rol, de igual forma sucede con el diseño de la base de datos, pues ya está organizada, en caso de que surja alguna tabla nueva, sería solo insertar o eliminar según el caso. Y por el contrario, en el rol del implementador, se necesitarían tres personas, ya que este módulo necesita, muchas personas capacitadas en este rol, con interés y mucho empeño, para que el producto salga en el tiempo estimado.

La nueva planificación de los recursos humanos del equipo de desarrollo del Proyecto GeHos, está estructurada como se ejemplifica en el **Anexo 1**.

El Jefe de Proyecto, el líder de desarrollo, el analista principal, el implementador integrado, el planificador, el Diseñador de la Base de Datos, el contador, el gestor de calidad y el Arquitecto. Son profesores de la facultad, en quienes se ha confiado y se les ha asignado este rol en el proyecto.

Se creó un equipo de desarrollo por cada módulo de los que está integrado el sistema. Los módulos están estructurados por los siguientes roles:

Jefe de Proyecto; en este caso sería el jefe del módulo, pero como cada módulo se desarrolla por separado, se toma como proyecto. Estos también son profesores, son los más relacionados con el equipo de desarrollo del módulo del cual están dirigiendo.

El analista; este rol lo realiza un estudiante, capacitado en todo lo referente al análisis y diseño del sistema.

Los implementadores: este rol lo desarrollan tres estudiantes, por módulos. Estos estudiantes deben estar bien preparados en las diferentes técnicas de programación que existen.

El diseñador de base de datos: Este estudiantes, debe tener un conocimiento previo de la realización de una base de datos y estará estrechamente relacionado con el profesor que atiende este rol en el proyecto.

Los estudiantes de la universidad, están preparados para realizar cada uno de estos roles. Algunos, se inclinan por un rol específico, en dependencia de lo que más le guste realizar. Además de tener los conocimientos necesarios, deben tener la responsabilidad de realizar el rol que se le asigne y realizarlo bien.

Por lo general, se encuentran muchos estudiantes, que se pueden quedar sin un rol específico de cada módulo. Para ubicarlos se organizaron dos grupos de apoyo; uno se encargará de la investigación necesaria y el otro de a calidad y prueba del software, dándole mantenimiento a lo que se vaya realizando sistemáticamente.

2.2.2 Recursos de software reutilizables

Los componentes de software reutilizables, permiten reducir el coste del desarrollo y agilizar la entrega del producto. En el caso del grupo de desarrollo GeHos, la universidad brinda los recursos necesarios, por lo que, el coste de producción del proyecto no es objetivo fundamental.

De todos modos, es necesaria la estimación de estos recursos. Por las características del software, el producto final debe ser la integración de todos los módulos que existen.

El módulo BQG, se sirve de otros módulos que se encuentran ya implementados como es el caso del módulo de *Configuración* que es el encargado de gestionar toda la información referente a los codificadores necesarios para el funcionamiento del módulo BQG. Y el componente de Seguridad que tiene como función principal gestionar la seguridad y los usuarios del sistema.

Para un buen trabajo del módulo BQG, se necesita la integración del mismo con otros módulos del sistema. Se utiliza el módulo *Farmacia* para obtener un registro de todos lo medicamentos que se necesitan en las diferentes operaciones. El módulo *Inscripción y Admisión*, se utiliza para buscar a los pacientes, obteniendo su historia clínica, y creando la historia clínica operatoria. La persona que realiza

esta función, tiene un resumen del paciente que se está consultando, como su carne de identidad, sus datos personales, y el por qué fue remitido para el salón de operación.

Para una operación es necesario saber lo análisis de sangre que se la han realizado la paciente, esto datos se obtienen del módulo de *Laboratorio*.

El módulo *Npgsql*, es primordial a la hora de establecer una comunicación con una base de datos postgresSQL. Es un proveedor de datos para el servidor de bases de datos PostgresSQL. Este permite establecer la comunicación entre una aplicación .Net (Console, WinForms, ASP.NET, Web Services.) con servidores de bases de datos PostgresSQL.

El Chameleon en su versión 4.1 es un subsistema que permite la construcción, validación y envío de mensajes HL7. HL7 es un estándar para el formato de datos e intercambio de información entre diferentes Sistemas de Información de Salud. Su misión es definir y publicar especificaciones y protocolos para comunicar sistemas de información sanitarios, dispersos, diferentes y heterogéneos; independientemente de: la arquitectura de los datos, la plataforma tecnológica o el lenguaje de desarrollo.

Middleware: Es la capa intermedia que contiene la implementación de los patrones, fabrica abstracta , factory method, singleton entre otros, que media la comunicación entre las clases del negocio y el acceso a datos ubicado en el servidor de bases de datos.

Se decidió tomar estos componentes porque cumplen los requisitos del módulo. Estos componentes reducen el número de defectos del producto que se le entregue al cliente.

2.2.3 Recursos de entorno

El entorno es donde se apoya el proyecto de software, el cual incorpora hardware y software. La realización de una buena planificación, del hardware y del software, se necesita para el desarrollo del software, y verifica que estos recursos estén disponibles.

La dirección de la facultad, le asignó al equipo de desarrollo dos laboratorios. Contando así con un total de 51 máquinas. El equipo está promediado a dos personas por máquinas, teniendo en cuenta un tiempo de máquina, realizado entre ellos mismos, teniendo en cuenta el horario de clases; este horario es consultado y aprobado por el jefe del proyecto y el planificador. Este lo lleva a una planificación utilizando el Microsoft Office Excel.

El horario se envía al controlador de versiones, Subversion, donde cada persona puede ver el tiempo que dispone para trabajar en la semana. Y el jefe del proyecto o del módulo, revisar quién falta a su horario de trabajo, provocando el atraso del software.

Los recursos que se describen a continuación, son en general, para todo el proyecto, o sea, cada módulo utilizará los mismos recursos que no son más que los que se han asignados en la facultad.

1. Para el modelado de todo el proceso de ingeniería, se utilizará el Rational Rose 2003 Enterprise Edition que demanda 256 MB de memoria RAM.
2. El IDE de desarrollo es, Visual Studio 2005 que para su funcionamiento es recomendado 512 MB de RAM, integrado con los plugins siguientes:
 - a. AnkhSVN 1.0 para el control de versiones.
3. Servidor de Bases de Datos: PostgreSQL 8.2 (512 MB de RAM)
4. Servidor de Versiones: Subversion.

El laboratorio de producción tiene un ambiente agradable, consta con dos aires acondicionados para el cuidado y mantenimiento de las computadoras, permitiendo una temperatura agradable para la concentración de los desarrolladores.

2.3 Estimación del Proyecto Software

La estimación de un proyecto no es exacta, existen muchos agentes externos que influyen en la variación de su cálculo; las técnicas, el entorno, las personas, que pueden afectar de forma directa el esfuerzo dedicado para el desarrollo del software.

En un software, la diferencia entre beneficios y desventajas puede ser en gran medida, un gran error en la estimación del mismo.

Una buena estimación, tiene como punto de partida, datos históricos del proyecto, con el objetivo de hacer una comparación y guiarse por el más factible. Si con los datos anteriores se obtuvo un resultado satisfactorio, o resultados desfavorables se puede ajustar la nueva estimación siempre pensando en el

mejor desarrollo. Este proyecto no tiene una base estable, pues la estimación no esta basada en datos históricos, siendo así un punto a favor de la gravedad de la estimación.

2.3.1 Técnicas empíricas de estimación

Utilizar métricas para el proyecto tiene dos aspectos fundamentales. En primer lugar, estas métricas se utilizan para minimizar la planificación de desarrollo haciendo los ajustes necesarios que eviten retrasos y reduzcan problemas y riesgos potenciales. En segundo lugar, las métricas para el proyecto se utilizan para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

A medida que mejora la calidad, se minimizan los defectos, y el tiempo que disminuye el número de defectos, la cantidad de trabajo que ha de rehacerse también se reduce. Esto lleva a una reducción del coste global del proyecto.

Existen diferentes métodos, para realizar la estimación de costo de un proyecto productivo. Primeramente se realiza la observación de algunos conceptos de Métricas.

Las métricas son una buena forma para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento. Las métricas del software se refieren a un amplio repertorio de mediciones para el proceso de computadora.

Líneas de código

El método de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido LOC (Lines Of Code) Líneas de Código o SLOC (Source Lines Of Code) Líneas de Código Fuente. 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:

$E = A + B \times (KLOC)^C$, 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.

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, debido a que E puede representar sólo el esfuerzo de codificación, o en el otro extremo, el esfuerzo total del análisis, diseño, codificación, prueba y mantenimiento. Por estas razones, comparar estos modelos se torna complejo.

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.

Puntos de función

El método del punto función, definido en la IBM, en 1979, es un método para medir el tamaño del software. Pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

El análisis por puntos de función es un método para cuantificar el tamaño y la complejidad de un sistema software en términos de las funciones de usuario que este desarrolla (o desarrollará). Esto hace que la medida sea independiente del lenguaje o herramienta utilizada en el desarrollo del proyecto.

El análisis por puntos de función está diseñado para medir aplicaciones de negocios; no es apropiado para otro tipo de aplicaciones como aplicaciones técnicas o científicas. Esas aplicaciones generalmente median con algoritmos complejos que el método de puntos de función no está diseñado para manejar.

El enfoque de puntos de función tiene características que permiten superar los principales problemas de utilizar líneas de código como métrica del tamaño del software. Primero, 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.

Segundo, 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 re estimación.

Tercero, como 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.

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.

Puntos de Característica.

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, se propuso ampliaciones a este método, generando una métrica que denominó Puntos de Característica. Ésta da cabida a aplicaciones cuya complejidad algorítmica es alta.

Este método considera los mismos elementos que se consideran en el 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).

Puntos de caso de uso

Es un método de estimación de esfuerzo de un proyecto de desarrollo de software a partir de los casos de uso.

El método utiliza los actores y casos de uso identificados para calcular el esfuerzo que constará desarrollarlos. A los casos de uso se les asigna una complejidad basada en transacciones, que son pares de pasos acción-usuario->respuesta-sistema de los escenarios de los casos de uso.

A los actores se les asigna una complejidad basada en el tipo de actor, es decir, si son interfaces con usuarios o si son interfaces con otros sistemas (API o protocolo). También se utilizan factores de entorno y de complejidad técnica para afinar el resultado.

Una vez asignada la complejidad a actores y casos de uso y establecidos los factores técnicos y de entorno, se calculan los puntos de caso de uso no ajustados o UUCP, el TCF (factor de complejidad técnica) y el EF (factor del entorno). Con ellos, se calculan los puntos de caso de uso o UCP, que finalmente se traducen a esfuerzo en horas-hombre con un sencillo cálculo.

COCOMO (COConstructive COst MOdel, por sus siglas en inglés) Modelo Constructivo de Coste.

Ésta técnica fue propuesta y desarrollada 1981, para dar una estimación del número de meses laborables que se tomará el desarrollo de un producto de software. Es uno de los modelos mejor documentados y más utilizados. La técnica permite calcular el esfuerzo y tiempo que se necesita en un proyecto de software utilizando las líneas de código que se estimen generar para la creación del software. Sus estimaciones están basadas en ecuaciones matemáticas.

Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente. La función que utiliza es $a(Kl^{dc})^b$ donde Kl^{dc} = Miles de líneas de código y mide en salarios-mes. (COCOMO 2007)

Este modelo refiere algunos inconvenientes para su utilización; los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizar las tareas. Se puede desviar de la realidad debido a la existencia de comentarios en las líneas de código. No es objetivo puesto que se realizan estimaciones sobre estimaciones del número de líneas de código que tendrá nuestro proyecto. Se mide el producto, o su tamaño, cosa totalmente diferente a medir la productividad. La medición por líneas de código no tienen sentido en la orientación a objetos. La medición se debe realizar al culminar el software, porque aquí es donde se tiene una cifra exacta de las líneas de código que tiene el sistema.

Sin embargo, esto crea una disyuntiva; se aboga, que la estimación se realice desde el inicio del proyecto, dándole un seguimiento en todo su proceso de desarrollo. Y este método no permite esta actividad.

Para un coste del proyecto final, sería bueno realizarlo, para tener una valoración.

2.3.2 Métricas de software

Las mediciones del mundo físico se pueden clasificar de dos formas: medidas directas (por ejemplo: la longitud de un tornillo) en el proceso de ingeniería se comprende el costo, y el esfuerzo aplicado, las

líneas de código producidas, la velocidad de ejecución, el tamaño de memoria y los defectos observados en un tiempo determinado. Y medidas indirectas (por ejemplo: la calidad de los tornillos producidos) donde se comprende la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc. del software.

Entre las medidas directas del proceso de la ingeniería del software de incluyen los costes y el esfuerzo aplicados. Entre las medidas directas del producto se incluyen las líneas de código (LDC) producidas, velocidad de ejecución, tamaño de memoria, y los defectos informados durante un período de tiempo establecido.

La diferencia que existe entre las medidas directas y las medidas indirectas es que el coste y el esfuerzo requerido para construir el software, el número de líneas de código producidas, y otras medidas directas son relativamente fáciles de reunir, mientras que los convenios específicos para la medición se establecen más adelante. Sin embargo, la calidad y funcionalidad del software, o su eficiencia o mantenimiento son más difíciles de evaluar y sólo pueden ser medidos indirectamente.

Las métricas del software son las relaciones con el desarrollo del software como funcionalidad, complejidad, eficiencia.

Métricas Técnicas: Está basada en las características del software. Mide la estructura del sistema, él cómo está hecho.

Métricas de la Calidad: Brinda una indicación de cómo se ajusta el software a los requisitos implícitos y evidentes del cliente. Es decir cómo voy a medir para que el sistema se adapte a las necesidades del cliente.

Existen diferentes medidas de calidad de software, pero los siguientes hitos son ventajosos para el equipo de desarrollo, en la realización del software.

La corrección no es más que el nivel en que el producto lleva a cabo su función requerida. Si el producto no trabaja correctamente, el usuario se desilusionará dándole poco valor al trabajo realizado.

La facilidad de mantenimiento, es la aptitud con la que se puede corregir un error en el programa, se debe medir de forma indirecta. Ésta es una actividad que requiere mucha voluntad en la ingeniería del software.

La integridad, brinda la información, de la capacidad que tiene el software para afrontar los ataques que pueden presentarse contra su seguridad. Los ataques se pueden medir en cualquiera de los elementos del sistema, dígame, programa, datos y documentos.

Una buena integridad se puede medir analizando la amenaza y la seguridad del software. Siendo la amenaza; la posibilidad de que un ataque (intencional o accidental) se presente en un momento determinado. Y la seguridad la posibilidad de que se pueda enfrentar este ataque.

La facilidad de uso, es la posible relación que tiene el usuario con el sistema, si le es amigable si se siente seguro en lo que está trabajando.

Estos factores expresados son algunos en los que se puede medir un proyecto para estimar la calidad del software.

Métricas de Productividad: Se centra en el rendimiento del proceso de la ingeniería del software. Es decir qué tan productivo va a ser el software que se va a diseñar.

Métricas Orientadas a la Persona: Proporciona medidas e información sobre el modo en que el equipo desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos.

Métricas Orientadas al Tamaño: Es para determinar en qué tiempo se va a realizar el software y cuántas personas se necesitan. Son medidas directas al software y el proceso por el cual se desarrollará. Si una organización de software mantiene antecedentes, se puede crear una tabla de datos orientados al tamaño como muestra la siguiente tabla:

Proyecto	LDC	Esfuerzo	Coste \$	Págs. Doc.	Errores	Defectos	Personas
Alfa	12,100	24	168	365	134	29	3
Beta	27,200	62	440	1224	321	86	5
Gamma	20,200	43	314	256	256	64	6
-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----

Figura 2.2 Métricas orientadas al tamaño.

La tabla enumera cada proyecto de desarrollo de software de los últimos realizados, mostrando sus datos basados en el tamaño de cada uno de ellos. Refiriéndonos a la entrada de la tabla (Figura 4.4) del proyecto Alfa: se desarrollaron 12.100 líneas de código (LDC) con 24 personas-mes y con un coste de \$168.000. Debe tenerse en cuenta que el esfuerzo y el coste registrados en la tabla incluyen todas las actividades de ingeniería del software (análisis, diseño, codificación y prueba) y no sólo la codificación. El proyecto Alfa indica que se desarrollaron 365 páginas de documentación, se registraron 134 errores antes de que el software se entregara y se encontraron 29 errores después de entregárselo al cliente dentro del primer año de utilización. También se conoce que trabajaron tres personas en el desarrollo del proyecto Alfa.

En los beneficios del sistema y los datos contenidos en la tabla se puede desarrollar, para cada proyecto un conjunto de métricas sencillas de productividad y calidad orientadas al tamaño. Se obtienen las siguientes fórmulas:

Productividad = $KLDC / \text{personas-mes}$

Calidad = $\text{errores} / KLDC$

Documentación = $\text{Págs. Doc} / KLDC$

Costo = $\$/KLDC$

- persona-mes es el esfuerzo.
- KLDC (miles de líneas de código)

Métricas Orientadas a la Función: Son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcular las LDC, las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Las métricas orientadas a la función fueron a principios propuestas por Albretch quien sugirió un acercamiento a la medida de la productividad denominado método del punto de función. Los puntos de función que obtienen utilizando una función empírica basando en medidas cuantitativas del dominio de información del software y valoraciones subjetivos de la complejidad del software.

2.4 Estimación del esfuerzo del módulo BQG mediante el método: Puntos de casos de uso

A la hora de decidir la factibilidad de un proyecto y de planificar su realización, es importante el papel de la estimación temprana del esfuerzo que se necesita. Se habla de una ardua labor con márgenes de error muy altos, principalmente en las empresas donde no se ha realizado una estimación previa de algún proyecto de software.

La estimación de el módulo que se analiza en este trabajo, no tiene una guía que permita hacer una comparación, debido a que anteriormente, el mismo proyecto no ha logrado resultados satisfactorios; por esta razón, se procede al siguiente paso de estimación, intentando definir un tiempo de desarrollo, el cuál se debe utilizar en proyectos futuros y en los diferentes módulos del proyecto, esta estimación debe ser predicción más optimista que tiene una probabilidad no nula de ser cierta.

Puntos por casos de uso, es la estimación mas acercada a la ingeniería del módulo que se rige por la metodología de desarrollo RUP. El proceso de esta metodología esta centrado en la arquitectura, y es dirigido por los casos de uso. También tiene en cuenta los requisitos funcionales y no funcionales del sistema, que influyen directamente en el desarrollo del trabajo.

Antes de realizar el procedimiento de estimación es necesario especificar algunos términos que se utilizan en el mismo:

Actor: El usuario que interactúa con los casos de uso, realizando un conjunto de funciones.

Casos de Uso: Descripción de una secuencia de acciones, incluyendo cursos de acción alternativos, que desarrolla un sistema y genera un resultado observable para un actor.

Transacciones: Son las respuestas que el sistema proporciona al usuario hacer una petición en el mismo.

Estimación del esfuerzo en el Módulo: Bloque Quirúrgico General del Proyecto GeHos de la facultad 7 de la Universidad de las Ciencias Informáticas.

El módulo Bloque Quirúrgico General del Proyecto Gestión Hospitalaria, de la facultad 7 en la Universidad de las Ciencias Informática, es uno de los más completos por su función en los sistemas hospitalarios. Se encarga del registro de todas las operaciones que se realizan en el hospital. Por su gran importancia y su pronta elaboración se ha realizado la estimación del esfuerzo en horas-hombre que se debe asumir para obtener la eficiencia que el módulo requiere en un corto tiempo de desarrollo.

En general, el método requiere de casos de uso en modo textual y gráfico, se revisan en detalles los casos de uso seleccionados por el analista del sistema en la etapa de análisis y diseño del proyecto, y se realizan los siguientes pasos:

La cuantificación de las características funcionales del Sistema:

- La clasificación de Actores, Obtención del Factor de Peso de Actores Sin Ajustar.
- La clasificación de los Casos de Uso, y obtención del Peso de las Transacciones sin Ajustar.
- La obtención del Peso o Puntos de Casos de Uso Sin Ajustar.

La cuantificación de las características no funcionales del sistema

- La clasificación del Factor de Complejidad Técnica.
- La clasificación de los Factores Ambientales
- El cálculo de Puntos de Casos de Uso Ajustados.

Cuantificación de las Características Funcionales del Sistema.

La cuantificación de los requisitos funcionales, se basa en la extracción de la información del modelo de casos de uso en su forma textual de acuerdo a una clasificación de Actores y Transacciones de los Casos de Uso.

Clasificación de Actores, Obtención del Factor de Peso de los Actores Sin Ajustar:

Se realizó un registro de todos los actores del sistema y se clasificaron en simples, promedio y complejos, de acuerdo al siguiente criterio:

- **Actor Simple:** Se trata de otro sistema interactuando a través de una interfaz de programación definida y conocida.
- **Actor Promedio:** Es otro sistema interactuando a través de un protocolo, (como TCP/IP)
- **Actor Complejo:** El actor complejo es una persona que interactúa con el sistema a través de una interfaz gráfica de usuario o página Web.

El módulo BQO tiene en su diseño del modelo de los Casos de Uso doce (12) actores del sistema.

Cinco (5) Simples; porque son sistemas que interactúan mediante una interfaz de programación definida:

- Módulo Inscripción y Admisión
- Módulo Administración
- Módulo Farmacia
- Módulo de Seguridad
- Módulo Laboratorio

Siete (7) Complejos; porque son personas del hospital que interactúan con el sistema mediante una interfaz gráfica de usuario:

- Anestesista
- Auxiliar de salón
- Clínico
- Especialista
- Jefe de servicio
- Pediatra
- Vice-Director Quirúrgico

Después de realizar la clasificación de los actores se debe asociarle el factor de peso de acuerdo a la siguiente tabla:

Tipo de Actor	Descripción	Factor
Simple	Interfaz de programación de aplicaciones.	1
Promedio	Interfaz de comunicación de vía protocolo.	2
Complejo	Interfaz gráfica de usuario.	3

Tabla: Factor de peso de los actores según su descripción.

Después de darles la clasificación a los actores, se procede con asignarle su factor de peso. Se cuentan los factores de acuerdo a su grado de complejidad, multiplicando cada subtotal por su factor de complejidad y sumando cada producto obteniéndose el peso de los actores sin ajustar.

$$(5*1) + (7*3) = 5 + 21 = 26 \text{ dando así el factor de Peso de los Actores Sin Ajustar.}$$

Clasificación de los Casos de Uso, y Obtención del Peso de las Transacciones Sin Ajustar

Con el modelo de los casos de uso, cada uno debe clasificarse en simple, medio o complejo, de acuerdo al número de transacciones descritas en el caso de uso, incluyendo los cursos de acción alternativos. Para realizar esta actividad se utiliza los siguientes criterios:

- **Casos de Uso Simple:** Tres (3) o menos transacciones.
- **Caso de Uso Medio:** Entre cuatro (4) y siete (7) transacciones.
- **Casos de Uso Complejo:** Más de ocho (8) transacciones.

El módulo BQ tiene descritos 17 casos de uso del sistema.

Diez (10) Casos de Uso complejos; porque tienen más de siete (8) transacciones.

- Gestionar Anuncio Operatorio
- Gestionar Hoja de Especialista
- Gestionar Informe Operatorio
- Gestionar Hoja del Pediatra
- Gestionar Hoja del Clínico
- Gestionar Hoja Consulta de Anestesista
- Gestionar Hoja A Acto Quirúrgico.
- Gestionar Planificación
- Buscar Planificación
- Cambiar el estado del Anuncio Operatorio.

Cuatro (4) Casos de Uso promedio; porque tienes de cuatro (4) a siete (7) transacciones.

- Buscar Anuncio Operatorio
- Buscar Consulta
- Gestionar Planificación Personal
- Buscar Informe Operatorio

Tres (3) Casos de Uso Simples; porque tienen tres (3) o menos transacciones.

- Gestionar Anuncio Operatorio Oftalmológico.
- Gestionar Consulta Especialista Oftalmológica
- Gestionar Informe Oftalmológico

Después de realizar la clasificación de los casos de uso del sistema se le debe asociar el factor de peso de acuerdo a la tabla siguiente:

Tipo de Caso de Uso	Descripción	Factor
Simple	3 o menos transacciones	5
Promedio	De 4 a 7 Transacciones	10
Complejo	Más de 8 transacciones	15

Tabla: Factor de peso de los casos de uso según su descripción.

De la misma forma que se realizó la clasificación con los actores del sistema, la cuenta de la transacción de los casos de uso se multiplican la complejidad y finalmente se suman los productos, obteniéndose el Peso de las Transacciones sin ajustar.

$$(3*5) + (4*10) + (10*15) = 205$$

La obtención del Peso o Puntos de Casos de Uso Sin Ajustar.

El Peso de los Caso de Uso Sin Ajustar es igual a la suma del Peso de los Actores Sin Ajustar, más el Peso de las Transacciones Sin Ajustar, es decir:

$$205 + 26 = 231$$

Cuantificación de las características no funcionales del sistema

El método por casos de uso, tiene en cuenta las características de complejidad técnica, analizando algunos requisitos no funcionales como un Factor de Ajuste al Sistema, y además, Factores Ambientales que se concentran en las características del equipo de desarrollo.

Factor de complejidad Técnica: Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla

se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Factor de Peso
T1	Sistema distribuido	2
T2	Objetivo de performance, o rendimiento o tiempo de respuesta.	1
T3	Eficiencia del usuario final	1
T4	Complejidad de procesamiento interno	1
T5	El código debe ser reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidades de uso	0.5
T8	Portabilidad	2
T9	Facilidad de cambio	1
T10	Concurrencia	1
T11	Características de seguridad; incluye objetivos especiales de seguridad	1
T12	Provee acceso directo a terceras partes	1
T13	Requerimientos de entrenamiento especial; se requieren facilidades especiales de entrenamiento.	1

Tabla: Factor de peso de los factores de complejidad técnica.

Factor	Valor	Comentario
T1	5	Permite tener la información en varios lugares brindado así que si se rompe un servidor, los demás no dejen de funcionar.
T2	4	La velocidad es rápida, en dependencia de las entradas previstas por los usuarios
T3	3	Tiene algunas restricciones de eficiencia
T4	3	Existe cálculos complejos, no son la gran mayoría
T5	5	Necesidad de que el código sea reutilizable.
T6	4	Existen requisitos para la facilidad de la instalación.
T7	5	Muy buena facilidad de uso.
T8	4	Es necesario que el sistema sea portable
T9	4	Es importante la facilidad de cambio en el sistema

T10	5	Es importante la concurrencia de los usuarios del sistema
T11	5	Es muy importante la seguridad del sistema
T12	0	Cualquier usuario Web no tiene acceso directo, solos los del hospital
T13	3	Algunos usuarios internos, sistema fácil de usar.

Tabla: Descripción y valor de los factores de complejidad técnicas del módulo BQG

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$TCF = 1.135$$

Clasificación de los Factores Ambientales

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.

Factores	Descripción	Factor de peso
F1	Familiaridad con el modelos del proyecto utilizado	1.5
F2	Experiencia en la aplicación	0.5
F3	Experiencia en orientación a objeto	1
F4	Capacidad del analista líder	0.5
F5	Motivación	1
F6	Estabilidad de los Requisitos	2
F7	Personal part.-time	-1
F8	Dificultad del lenguaje de programación	-1

Tabla: Factor de peso de los factores ambientales.

- Para los factores F1 al F4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor F5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor F6, 0 significa requisitos extremadamente inestables, 3 estabilidad media y 5 requisitos estables sin posibilidad de cambios.

- Para el factor F7, 0 significa que no hay personal part.-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part.-time (nadie es full-time).
- Para el factor F8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

Factor	Comentario	Valor
E1	El grupo está bastante familiarizado con el modelo.	4
E2	La gran mayoría de los estudiantes han trabajado en ésta aplicación	4
E3	Todos los implementadores trabajan en Programación OO	5
E4	Los analistas son estudiantes, que no tienen mucha experiencia.	3
E5	La gran motivación la tiene los estudiantes de 5to, no se ve mucho en los demás años	4
E6	Sujeto a cambios	3
E7	Todo el grupo, no están a full time, solo los implementadores.	5
E8	Se utiliza C#	0

Tabla: Descripción y valor de los factores ambientales del módulo BQG

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso } i \times \text{Valor } i)$$

$$EF = 0.815$$

El cálculo de Puntos de Casos de Uso Ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

UCP = UUCP x TCF x EF; donde,

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

$$UCP = 231 * 1.135 * 0.815 = 2135.6807$$

Calcular el Esfuerzo: De los Puntos de Casos de Uso a la estimación del esfuerzo;

- Se contabilizan cuántos factores de los que afectan al Factor de Ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Total EF = Cantidad EF < 3 (entre E1 –E6) + Cantidad EF > 3 (entre E7, E8)

Total EF = 0 + 1 =1

Como el Factor de Conversión (FC) es de 20 horas/hombre.

El esfuerzo en horas-hombre viene dado por:

$E = UCP \times CF$

$E = 213.6807 * 20 \text{ horas/hombre}$

$E = 4273.6125 \text{ horas/hombre}$

Donde:

E: Esfuerzo de implementación estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso, en la implementación.

Para una estimación más completa de la duración total del módulo, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

- Análisis, llevaría un 10% del esfuerzo general del proyecto.
- Diseño, llevaría un 20% del esfuerzo general del proyecto.
- Implementación, llevaría un 40% del esfuerzo general del proyecto.
- Prueba, llevaría un 15% del esfuerzo general del proyecto.
- Sobrecarga, llevaría un 15% del esfuerzo general del proyecto.

El esfuerzo calculado es el de la Implementación, para calcular el de las diferentes actividades, es muy fácil, por la regla de 3.

O sea; el esfuerzo calculado, el de la implementación, es el 40% de 100, cuál será el valor del esfuerzo del análisis si éste consta de un 10% de las actividades. Y así para las distintas actividades.

El resultado del valor para cada una de las actividades, se muestra en siguiente tabla.

Actividad	% Esfuerzo	Valor Esfuerzo
Análisis	10%	1068.4038
Diseño	20%	2136.8077
<u>Implementación</u>	<u>40%</u>	<u>4273.6155</u>
Prueba	15%	1602.6058
Sobrecarga	15%	1602.6058
Total	100%	10684.0387

Tabla: Por ciento del esfuerzo en las actividades

Con esta estimación realizada, que no se puede decir que es estricta, ya que influyen muchos factores empíricos, como se había dicho anteriormente, se tiene una idea de, cuántas horas se trabajarán y cuántas personas trabajarán en el proyecto, y planificar una fecha de culminación aproximada, basada en estos datos.

El módulos BQG, tendrá un valor de esfuerzo aproximado de 10684 horas por todos los estudiantes del equipo de trabajo. Lo que implica un aproximado de 445 días.

445 días son aproximadamente un año y tres meses, estos datos son aproximados pues, se debe tener en cuenta que los domingos no son días laborables en el proyecto. La estimación es un dato aproximado, que se debe ir verificando, cada un mes en el desarrollo del proyecto, para no dar una tiempo de terminación que después no se pueda cumplir.

Según lo establecido en el tiempo de máquina. Un estudiante trabaja, lunes, martes y miércoles, desde las ocho de la mañana hasta las doce del día, y desde las dos de la tarde hasta las seis de la tarde, lo que resulta, 24 horas. El mismo estudiante trabaja los martes y miércoles desde las seis de la tarde, hasta las 12 de la noche, para un resultado de doce horas y los sábados cuatro horas al día. Con un aproximado de 30 horas por semana.

Si en el proyecto trabajan ocho estudiantes, el mismo puede terminar en aproximadamente 44 semanas. Siendo este cálculo un aproximado de 11 meses de trabajo.

Después de obtener la estimación de esfuerzo de la fase de implementación, se realiza, mediante una proporción, la estimación de las restantes fases. Para un futuro proyecto, no es necesario dedicarle mucho esfuerzo al análisis y diseño del sistema, pues ya están enmarcadas las funcionalidades del mismo, aunque puede surgir alguna nueva, pero esto no tomará mucho tiempo.

La implementación si debe ser una fase que necesita el mayor esfuerzo para su desarrollo, además depende en gran medida de la existencia del análisis y diseño. La prueba es una fase que no necesariamente se realiza después de culminado el sistema, por lo general, se le aplica una prueba a cada función que se implemente, para detectar el error en tiempo y poder erradicarlo.

Herramienta: Estimación de Proyectos Basada en el Análisis de Puntos de Casos de Uso (ESTIMAC).

En la universidad de las ciencias Informáticas, en la facultad 4, se realizó una herramienta, con el objetivo de agilizar, teniendo los datos suficientes, el proceso de la estimación del proyecto, basada en los puntos de los casos de uso. Esta herramienta es la que se utiliza en la asignatura Ingeniería informática, para el aprendizaje de la misma.

Su autor es un profesor de la facultad 4, fue programada en el lenguaje de programación Delphi 6.0. Se utiliza en las clases práctica de la asignatura de ingeniería del software, agilizando y perfeccionando el trabajo realizado en las mismas.

Entre muchas de las opciones que brinda, está la ayuda, para la mejor comprensión de los cálculos, y un reporte de lo estimado con el nombre del proyecto.

2.5 Análisis y gestión de los riesgos

A menudo los proyectos de desarrollo de software se enfrentan a una variedad de problemas, los cuáles no se pueden ignorar. Muchos de estos problemas están centrados en la incompetencia en gestión de proyectos, las planificaciones y por lo general en la falta de motivación de los desarrolladores.

El fracaso y la mala calidad de muchos productos software están dados por los diferentes problemas de comunicación entre los integrantes del equipo y entre los clientes y el equipo de desarrollo. Estos factores que afectan el éxito de un proyecto, se pueden clasificar como riesgos.

El análisis y la gestión de los riesgos, tiene como objetivo; identificar, controlar y eliminar las fuentes de riesgos, antes de que empiecen a afectar a cumplimiento de los objetivos de los proyectos

2.5.1 Definición y Clasificación

El riesgo se define como “la exposición a la posibilidad de ocurrencia de ciertos entes tales como pérdida o ganancia económica, daño físico, retrasos, daño a la salud pública, etc. que surgen como consecuencia de seguir un curso particular de acción.

Riesgo: Posibilidad futura de un evento adverso, desgracia o contratiempo que pueden ocasionar pérdidas si ocurre. Es un problema potencial, puede ocurrir o no.

Riesgo: Cada una de las contingencias que pueden ser objeto de un contrato de seguro. Dicho de acometer una empresa o de celebrar un contrato: Sometiéndose a influjo de suerte o evento, sin poder reclamar por la acción de estos.

Análisis y gestión de riesgos: Serie de pasos que ayuda al equipo de software a comprender y a gestionar la incertidumbre.

Para un buen análisis y gestión del riesgo, es válido aclarar que el riesgo implica incertidumbre y pérdida potencial. La **Incetidumbre:** Es el evento que define al riesgo, este evento, puede ocurrir o no. Y la **Pérdida Potencial:** Las consecuencias indeseadas en caso de que el riesgo se convierta en una realidad.

Para realizar el análisis del riesgo, existen diferentes categorías de clasificación del riesgo del software, con el objetivo de darle un nivel de incertidumbre y un grado de pérdidas a cada uno. Además de poder identificar los tipos de riesgos que se pueden encontrar en la construcción del software.

Riesgos del Proyecto (RP): Si los riesgos que amenazan al proyecto se hacen realidad; puede atrasar la planificación temporal y aumentar el coste del proyecto, afectando así la calidad del mismo. Estos riesgos permiten identificar los problemas de planificación temporal, de personal, de recursos, de clientes y de requisitos.

Riesgos Técnicos (RT): Estos riesgos están relacionados con el producto, amenazando su calidad. Estos riesgos identifican los posibles problemas de incertidumbre técnica, diseño, implementación, verificación, mantenimiento, etc.

Riesgos del negocio (RN): Estos riesgos amenazan la capacidad del software. Dentro del negocio se identifican los siguientes riesgos:

- *Riesgos de Mercado:* Dándose la situación de realizar un sistema excelente y que en realidad nadie quiera.
- *Riesgo de dirección:* Perder el apoyo de una gestión experta, debido a cambios de enfoque o a cambios de personal.
- *Riesgo estratégico:* Realizar un producto que no se ajusta a la estrategia comercial de la compañía.
- *Riesgo de Ventas:* Después de realizado, el producto no es vendible.
- *Riesgos de presupuestos:* Perder presupuesto o personal asignado.

2.5.2 Componentes y controladores de riesgos

Los componentes de riesgo se definen:

- *Riesgo de Rendimiento:* No es más que el grado de incertidumbre con que el producto encontrará sus requisitos y se adecue para su empleo pretendido.
- *Riesgo del Coste:* El grado de incertidumbre que mantendrá el presupuesto del proyecto.

- **Riesgo de Soporte:** El grado de incertidumbre de la facilidad del software para corregirse, adaptarse y ser mejorado.

- *Riesgo de Planificación Temporal:* El grado de incertidumbre con que se podrá mantener la planificación temporal y de que el producto se entregue a tiempo.

2.5.3 Actividades para el análisis y la gestión de riesgos

Debido a la importancia de la gestión de los riesgos, es bueno, identificarlos, evaluar su probabilidad de aparición, estimar su impacto en caso de que ocurra, priorizándolos en función de su probabilidad e impacto, determinando así cuáles son los más importantes y establecer un plan de contingencia por si ocurre el problema, implementando una estrategia para resolverlos y asegurando su eficacia.

2.5.3.1 Identificación

La Identificación de los riesgos, para poderlos evitar cuando sea posible y controlarlos cuando sea necesario. Crear una Lista de comprobación de elementos de riesgos, es un método para identificarlos.

Riesgos	Categoría
Las actividades extra-docentes, inciden directamente en el tiempo de realización del proyecto. Atrasando el trabajo a realizar, provocando poco aprovechamiento del proyecto.	RP
Los fallos eléctricos que se pueden ocasionar, trayendo consigo, la descomposición de las máquinas. Lo que implica que la pérdida de información y el atraso del desarrollo del sistema, porque tendría que iniciar de nuevo el proceso.	RT
Los fallos eléctricos que se pueden ocasionar, trayendo consigo, la descomposición de las máquinas. Lo que implica que la pérdida de información y el atraso del desarrollo del sistema, porque tendría que iniciar de nuevo el proceso.	RN
Error en la asignación de los recursos humanos a las tareas, así como la máquina que se necesita. Provocando que no se aproveche este recurso al máximo.	RT
El cambio constante de personal. Lo que implica que las personas nuevas tengan que ponerse al tanto en las actividades que se realizan en el proyecto, atrasando el desarrollo del sistema.	RN

La tecnología a utilizar, puede que no sea fácil, o necesite licencia y/o no se pueda integrar al final.	RT
La motivación del equipo de trabajo, para realizar las tareas que se les asigne.	RN
Una mala planificación y estimación del tiempo, después no se pueda entregar el producto en el tiempo estimado.	RP
Cambios en el proceso del negocio, o sea cambios en decisiones del ámbito del proyecto.	RN

2.5.3.2 Estimación y proyección del riesgo

Una vez identificado los riesgos, comienza con una etapa de estimación o proyección del mismo, donde la actividad principal se centra, en cuantificar la probabilidad de ocurrencia y el impacto.

Establecer una escala que refleje la probabilidad percibida del riesgo, definir las consecuencias del riesgo, estimar el impacto en el proyecto y en el producto y evaluar la exactitud de esta proyección, sería lo más ideal.

Para mejor la comprensión de la estimación, se puede acudir a una tabla de riesgo.

Riesgos	Categoría	Probabilidad	Impacto
1. Las actividades extra-docentes, inciden directamente en el tiempo de realización del proyecto. Atrasando el trabajo a realizar, provocando poco aprovechamiento del proyecto.	RP	60%	2
2. Los fallos eléctricos que se pueden ocasionar, trayendo consigo, la descomposición de las máquinas. Lo que implica que la pérdida de información y el atraso del desarrollo del sistema, porque tendría que iniciar de nuevo el proceso.	RT	30%	3
3. Indecisión de los clientes, no saben que es lo que quieren, o no se saben explicar. Lo que provoca que el analista no pueda definir las funciones del sistema y a su vez no se puede desarrollar la implementación del mismo.	RN	50%	2
4. Error en la asignación de los recursos humanos a las tareas, así como la máquina que se necesita. Provocando que no se aproveche este recurso al máximo.	RT	20%	3

5. El cambio constante de personal. Lo que implica que las personas nuevas tengan que ponerse al tanto en las actividades que se realizan en el proyecto, atrasando el desarrollo del sistema.	RN	80%	1
6. La tecnología a utilizar, puede que no sea fácil, o necesite licencia y/o no se pueda integrar al final.	RT	10%	2
7. La motivación del equipo de trabajo, para realizar las tareas que se les asigne.	RN	70%	2
8. Una mala planificación y estimación del tiempo, después no se pueda entregar el producto en el tiempo estimado.	RP	80%	1
9. Cambios en el proceso del negocio, o sea cambios en decisiones del ámbito del proyecto.	RN	80%	1

Tabla: Categoría, probabilidad e impacto del riesgo.

Para listar los riesgos y la categoría se realiza una lista de comprobación de riesgos. La probabilidad de aparición de cada riesgo se analiza con los integrantes del proyecto. Seguidamente se evalúa el impacto de cada riesgo. Para esto es necesario tener en cuenta los siguientes puntos:

La naturaleza del riesgo proporciona la información de los problemas probables que aparecería si este ocurre.

El alcance de un riesgo ajusta la seriedad (¿Cuán riguroso es el problema?) con su distribución en el proyecto (¿cuántos clientes se verán perjudicados?).

La temporización de un riesgo considera cuándo y por cuánto tiempo se dejará sentir el impacto.

Se puede dividir el impacto de cada riesgo en 4 categorías –despreciable, marginal, crítico y catastrófico--. En el **Anexo 2**, se describe una caracterización de las consecuencias potenciales de errores (fila etiquetada con 1) o la imposibilidad de conseguir el producto deseado (fila etiquetada con 2). La categoría de impacto es elegida basándonos en la caracterización que mejor encaja con la descripción de la tabla

Estas actividades se aplican reiteradamente a medida que avance el proyecto. El encargado de esta actividad, reunido con el equipo del proyecto, debería volver a la tabla de riesgos, volver a evaluar cada riesgo para determinar qué nuevas circunstancias han podido cambiar su impacto y probabilidad. A consecuencia de esta actividad, podrá ser necesario añadir nuevos riesgos a la tabla, quitar algunos que no sean relevantes y cambiar la posición relativa de los otros.

El valor del impacto se clasifica de la siguiente forma:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

2.5.3.3 Reducción y supervisión de la gestión del riesgo

Todo lo realizado hasta ahora en el análisis y la gestión del riesgo tiene un único objetivo, que es ayudar al grupo de trabajo del proyecto a desarrollar una estrategia para tratar los riesgos. Para que esta estrategia sea eficaz debe tener en cuenta los siguientes aspectos:

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

Por tanto tratar de evitar el riesgo es la mejor estrategia, seguro te preguntarás cómo lo hago. Bueno, desarrollando un Plan de Reducción y Supervisión del Riesgo.

El plan RSGR

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 (Plan RSGR). 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. A continuación se expone un esquema del Plan RSGR.

Los riesgos en esta tabla son ordenados en dependencia de su impacto y su probabilidad. Los riesgos con un alto impacto y una elevada probabilidad, son los primeros, y así de forma consecutiva. Esta orden permite hacer una línea de corte, para poder priorizar a los riesgos que quedan por encima de esta línea. Esto no significa que los riesgos que quedan por debajo, no se van a evaluar, al contrario, son los que se analizan una y otra vez hasta poder eliminarlo. Los riesgos priorizados o de primer orden, son los que el jefe de proyecto trabajará de inmediato en su reducción y control.

Realizando el orden y aplicando la línea de corte, la tabla quedaría de la siguiente forma.

CAPÍTULO 2: PLANIFICACIÓN DEL MÓDULO BQG

Riesgos	Categoría	Probabilidad	Impacto	PRSGR
1. Una mala planificación y estimación del tiempo, que después no se pueda entregar el producto en el tiempo estimado.	RP	80 %	1	Realizar una planificación real del proyecto, y evaluar y controlar las actividades que se realizan, apoyándonos en una de las herramientas que existen.
Cambios en el proceso del negocio, o sea cambios en decisiones del ámbito del proyecto.	RN	80%	1	Permitir a los desarrolladores del sistema la integración con el cliente del mismo, y establecer una serie de preguntas previas, que permitan al desarrollador entender lo que realmente necesita el usuario.
El cambio constante de personal.	RT	80%	1	Definir bien el método de captación de personal, especificarles la necesidad de que se necesitan personas consagradas y responsables. Además darles las condiciones favorables que les permita desarrollar su trabajo, con gusto.
La motivación del equipo de trabajo, para realizar las tareas que se les asigne.	RN	70%	2	Realizar conferencias, mensuales, con el objetivo de ir mostrando los avances que se han ido logrando en cada módulo.
Las actividades extra-docentes, inciden directamente en el	RN	60%	2	Trabajar horas extra, las noches, para ir adelantando el trabajo y que estas actividades no caigan de sorpresa

CAPÍTULO 2: PLANIFICACIÓN DEL MÓDULO BQG

tiempo de realización del proyecto. Atrasando el trabajo a realizar, provocando poco aprovechamiento del proyecto.				
Línea de Corte				
Indecisión de los clientes, no saben que es lo que quieren, o no se saben explicar.	RN	50%	2	Preparar a lo analistas, para que las preguntas que les realicen a los clientes sean precisas, claras y brinde la información requerida.
Los fallos eléctricos que se pueden ocasionar, trayendo consigo, la descomposición de las máquinas.	RT	30%	3	Todo lo que se vaya realizando debe tener una copia en otro ordenador, así como debe estar en el repositorio de información que debe tener el proyecto.
Error en la asignación de los recursos humanos a las tareas, así como la máquina que se necesita.	RT	20%	3	Se debe priorizar al role que esté en el momento atrasado o que se necesite que esté a tiempo completo en las máquinas.

Una vez que se ha desarrollado el plan RSGR y el proyecto ha comenzado, comienzan los procedimientos de reducción y supervisión del riesgo como ya se ha mencionado antes, la reducción del riesgo es una actividad para evitar problemas La supervisión del riesgo es una actividad de seguimiento del proyecto con tres objetivos principales; valorar cuando un riesgo previsto ocurre; asegurarse de que los procedimientos para evitar el riesgo definidos se están aplicando apropiadamente; y recoger información

que pueda emplearse en el futuro para analizar riesgos. En muchos casos, los problemas que ocurren durante un proyecto pueden afectar a más de un riesgo. Otro trabajo de la supervisión de riesgos es intentar determinar el "origen" (qué riesgos ocasionaron tal problema) a lo largo de todo el proyecto. (BARZANALLANA 2007)

Plan de contingencia

Después de analizar el riesgo y evaluar su probabilidad de ocurrir y su impacto, se realiza un plan de reducción y supervisión de los mismos. Existen riesgos que no se pueden evitar, por mucho esfuerzo que se le dedique a su reducción. Por esta razón, se realiza un plan de contingencia, que permite darle soluciones a los riesgos en caso de que ocurran.

Para realizar el plan de contingencia, se puede agregar una columna a la tabla anterior, o sea después del plan de reducción y supervisión de la gestión de los riesgos, pero para una mejor organización del proyecto, es mejor realizar una nueva tabla, que se denomine, plan de contingencia, que contenga dos columnas.

Una columna con el nombre y la descripción del riesgo, y en la otra columna, especificar qué medidas se tomarán, en caso de que este riesgo se haga realidad.

Riesgo	Plan de Contingencia
Una mala planificación y estimación del tiempo, que después no se pueda entregar el producto en el tiempo estimado.	Re-planificar, estimar un tiempo de entrega, más real, que permita a los desarrolladores aprovecharlo, y siempre analizando que, los estudiantes tienen clase docentes y alguna que otra actividad. Por tanto se propone que la que la estimación se realice cada un mes. Ajustando los datos según lo que se ha realizado y lo que falta por realizar.
Cambios en el proceso del negocio, o sea cambios en decisiones del ámbito del proyecto.	Estar preparado para que este riesgo ocurra es lo primero, después, tener bien claras las ideas y dedicarle todo el tiempo necesario a esta actividad que es tan sumamente importante para todo el proceso de desarrollo.

CAPÍTULO 2: PLANIFICACIÓN DEL MÓDULO BQG

El cambio constante de personal.	Realizar una reunión el mismo día que entre la persona, para darle una visión de lo que se quiere en el proyecto, y ponerlo en adaptación con otro desarrollador para que lo actualice.
La motivación del equipo de trabajo, para realizar las tareas que se les asigne.	En caso de que este riesgo suceda, se le informará al equipo todos los avances que se han logrado, y se realizara un trabajo, explicándoles la necesidad de lo que se está desarrollando para el país
Las actividades extra-docentes, inciden directamente en el tiempo de realización del proyecto. Atrasando el trabajo a realizar, provocando poco aprovechamiento del proyecto.	Re-planificar, dándole más tiempo de realización a la actividad que se vea afectada por este riesgo. Informar a los jefes del proyecto.
Indecisión de los clientes, no saben que es lo que quieren, o no se saben explicar.	Realizar preguntas claras, y tener más contacto con los clientes, todos los necesarios. Explicarles de qué forma se puede ayudar con la realización de un software que implemente algunas de las actividades que ellos realizan. Ser insistentes hasta lograr lo deseado.
Los fallos eléctricos que se pueden ocasionar, trayendo consigo, la descomposición de las máquinas.	Si este riesgo ocurre, se debe tener la información en un repositorio, que utilizará el proyecto, en este caso, en el Subversion. Si esta información no está guardada y se pierde, de igual forma se le proporciona la máquina en tiempo completo al desarrollador y se le estima el tiempo que debe estar terminada la actividad, como una meta, para que lo realice lo más rápido posible.
Error en la asignación de los recursos humanos a las tareas, así como la máquina que se necesita.	En caso de que este riesgo se haga realidad, se revalora el caso, y se reajusta la planificación de estos recursos.

En este capítulo, se aprecia, el procedimiento propuesto, para la realización de una planificación en el grupo de desarrollo GeHos. El planificador debe tener en cuenta tres aspectos fundamentales antes de

iniciar el proceso de desarrollo del proyecto; qué tiempo durará, cuántas personas necesitará y cuánto esfuerzo tomará su realización. Además, debe prevenir los riesgos que pueden estar implicados en el proceso de desarrollo y los recursos de hardware y software que va a necesitar.

Para ayudar a realizar las estimaciones, se cuenta con el enunciado del ámbito del proyecto, y con las técnicas siguientes; modelos y herramientas de estimación. Los modelos brindan el número de personas-mes requeridas para la implementación de las funciones o para cada etapa de la ingeniería del software. Las herramientas efectúan los modelos, facilitando su realización.

Una persona con un buen sentido común de la realización de un proyecto de software, analiza los riesgos que se pueden presentar. En muchos casos esto se realiza informal y superficialmente. En este capítulo, se propone una guía para el análisis y la gestión de los riesgos, para tenerlos archivados y poder trabajar con ellos de un modo más fácil, para su reducción y supervisión.

El tiempo utilizado para la identificación, reducción y supervisión de los riesgos, es muy importante y vale la pena. Por muchos motivos específicos; pues así se disminuyen los trastornos durante el proceso del proyecto, los que brinda una mayor confianza para paliar los problemas antes de que ocurran.

CAPÍTULO 3: PLANIFICACIÓN TEMPORAL Y SEGUIMIENTO DEL PROYECTO

Después de haber realizado un modelo de proceso adecuado en la planificación, identificar las tareas de ingeniería del software que se van a llevar a cabo, después de estimar la cantidad de trabajo y el número de personas necesarias, establecer un plan de fechas de entregas del producto, donde se han considerado los riesgos; ahora es el momento de unir todo lo realizado. Luego de crear una red de tareas, se asigna la responsabilidad para cada tarea, asegurando de que se hagan en el tiempo definido.

La planificación consiste en diseñar un futuro deseable y seleccionar o crear las formas de lograrlo, hasta donde sea posible. Por lo tanto, al planificar, construye la **secuencia** de tareas con **lógica** necesaria, y la asignación de **recursos** necesarios para alcanzar el objetivo del proyecto en un **tiempo** óptimo.

La disponibilidad de recursos hace, que la secuencia de tareas pueda variar en el tiempo; dependiendo de los recursos con que se dispongan. Por lo que, al momento de planificar, hay que considerar, las tareas y los recursos; con el mismo grado de importancia.

La planificación temporal y el seguimiento de proyecto son muy importantes, principalmente en el proyecto GeHos, donde las tareas se realizan en paralelo, y el resultado del trabajo desarrollado durante una tarea, puede tener un gran efecto en el trabajo a realizar en otras tareas. Además es casi imposible evaluar el progreso del proyecto sin una previa planificación detalla.

3.1 Objetivo

La planificación temporal tiene como objetivo definir todas las tareas del proyecto, construir una red que describa sus interdependencias, identificar las tareas que son críticas dentro de la red y después hacerles un seguimiento para determinar de inmediato posibles retrasos.

3.2 Conceptos básicos

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.

La planificación evoluciona con el tiempo: durante las primeras etapas se desarrolla una *planificación temporal macroscópica* y a medida que el proyecto va progresando se refina obteniéndose una *planificación temporal detallada*

3.3 Principios de la planificación temporal

La planificación temporal de proyectos de software tiene como principios:

Comportamiento: Descomposición del proyecto en un número manejable de tareas.

Interdependencia: Se debe determinar las dependencias de cada tarea.

Asignación de tiempo: A cada tarea se le asigna un cierto número de unidades de trabajo, una fecha de inicio y otra finalización.

Validación del esfuerzo: A medida que se realiza la asignación de tiempo, el gestor del proyecto se tiene que asegurar de que los técnicos necesarios estarán disponibles en cada momento.

Responsabilidades definidas: Cada tarea que se programe debe asignarse un miembro específico del proyecto.

Resultados definidos: El resultado de cada tarea, normalmente un producto, será definido. Los productos se combinan generalmente en entregas.

Hitos definidos o Sucesos: Todas las tareas, grupos de tareas, deben asociarse con algún hito del proyecto. Se considera un hito cuando se ha revisado la calidad de uno o más productos y se han aceptado.

3.4 Métodos de Planificación temporal

La planificación temporal de un proyecto de software, no difiere mucho de la de cualquier otro esfuerzo de desarrollo de multitarea. Se pueden utilizar las técnicas y herramientas generales de planificación temporal de proyectos para el desarrollo de software, con pequeñas modificaciones; entre ellas se pueden

citar la técnica de Evaluación y de Revisión de programas, y el método de la Ruta Crítica y el Diagrama de GANTT los que serán tratados más adelante.

La **Técnica de Evaluación y Revisión de Programas (PERT)** y el **Método de la Ruta Crítica (CPM)** son dos métodos de la planificación temporal de proyectos que pueden aplicarse al desarrollo de proyectos informáticos. Ambas técnicas desarrollan una descripción de la red de tareas del proyecto, es decir, una representación gráfica de las tareas que deben realizarse desde el principio hasta el final del proyecto.

MICE (Minimum Cost Expediting), Aceleración del proyecto a coste mínimo o PERT Coste: Es una de las variantes de CPM, pero introduciendo la relación que existe entre coste y duración de una actividad. De esta forma se obtiene la programación de proyectos a coste mínimo.

Método de ROY: Desarrollado en Europa entre 1958 y 1961 por un grupo de ingenieros. Similar a los métodos PERT y CPM, pero permite establecer las redes sin utilizar actividades ficticias e iniciar los cálculos sin la construcción de la red.

Método GERT (*Graphical Evaluation & Review Technique*): El método GERT extiende la incertidumbre en la duración de las actividades a la propia programación, permitiendo considerar un número mayor de situaciones del proyecto que otros métodos. Las actividades precedentes de cada uno pueden ser de naturaleza determinante o probabilística.

- Otros métodos:

Métodos de secuencia mínima irreducible para programas de mantenimiento.

PEP (*Program Evaluation Procedure*) desarrollado por las Fuerzas Aéreas de EEUU.

PERT – Recursos: Aplicable cuando existen limitaciones en los recursos.

PERT (Program Evaluation & Review Technique):

- Creado para proyectos del programa de defensa del gobierno norteamericano entre 1958 y 1959.
- Se utiliza para controlar la ejecución de proyectos con gran número de actividades que implican investigación, desarrollo y pruebas.

CPM (Critical Path Method):

- Desarrollado para dos empresas americanas entre 1956 y 1958.
- Se utiliza en proyectos en los que nunca hay poca incertidumbre en las estimaciones.

Red de tareas: Representación mediante una estructura en red de las tareas e hitos del proyecto.

Diagrama de Gantt: Representación gráfica de las tareas sobre una escala de tiempos.

3.4.1 Red de tareas

Para lograr una planificación temporal de un proyecto, el planificador debe distribuir un conjunto de tareas en las diferentes etapas del proceso de desarrollo. Un conjunto de tareas es una suma de actividades de la ingeniería del software, hitos y entregas, que deben realizarse, para ejecutar el sistema.

Este conjunto de tareas requiere de mucha disciplina para lograr una aceptación en la calidad del software. Aunque, no se debe asignar al equipo de desarrollo una innecesaria cantidad de trabajos.

Las tareas generales del proyecto definen una planificación temporal a grandes rasgos. Esta planificación debe ser puntualizada, para crear una planificación temporal detallada del proyecto. La puntualización no es más que la descomposición del conjunto de tareas un conjunto de subtareas; donde se especifican las actividades que se realizan dentro de este conjunto.

3.4.1.1 Definir una red de tareas

Las tareas y subtareas individuales tienen interdependencias basadas en su secuencia. Además, cuando hay más de una persona implicada en un proyecto de ingeniería del software, es las actividades de desarrollo y las tareas se realizan en paralelo. En este caso, las tareas concurrentes deben coordinarse de manera que estén finalizadas cuando tareas posteriores requieran su(s) resultado(s).

Red de tareas o red de actividades: es una representación gráfica del flujo de tareas de un proyecto. Representa las tareas que deben ejecutarse en paralelo y las que deben llevarse a cabo en secuencia debido a una dependencia respecto a la tarea o tareas anteriores.

Existen 3 tipos básicos de técnicas para construir el Diagrama de Redes del Proyecto:

- PDM (Precedence Diagramming Method),

- CDM (Conditional Diagramming Methods) y
- ADM (Arrow Diagramming Method)

Técnicas de Secuenciación: PDM

- Método de Diagramación de Precedencias; consiste en construir un DPR utilizando nodos para representar las actividades y conectándolas con flechas que representan las dependencias. Es además el método más utilizado.

Si una actividad A precede a otra B, existen cuatro *tipos de relaciones de dependencia*.

- Acabar-para-empezar: la actividad A debe concluir antes de poder comenzar la B.
- Acabar-para-acabar: la actividad A debe haber concluido antes de poder concluir también la B.
- Empezar-para-empezar: la actividad A debe comenzar antes de poder concluir la B (no usada en software).

Técnicas de secuenciación ADM y CDM

ADM: Método de Diagramación por Flechas:

- Construir un Diagrama de Redes utilizando flechas para representar las actividades y nodos para indicar las dependencias entre actividades.

3.4.2 Diagrama de GANTT

Los cronogramas de barras o “gráficos de Gantt” fueron concebidos por el ingeniero norteamericano Henry L. Gantt, uno de los precursores de la ingeniería industrial contemporánea de Taylor.

Una de las responsabilidades más habituales del jefe del proyecto es informar sobre el avance del proyecto a sus superiores. Los gráficos Gantt suelen utilizarse para mostrar el avance de los proyectos, en virtud de que pueden compararse de forma conveniente la planificación original con el desarrollo real. Para informar del avance del proyecto se tiene que ampliar las convecciones propias del gráfico de Gantt. Si una tarea ha sido completada, su barra correspondiente aparecerá más oscura. Si ha sido completada solo parcialmente, la parte proporcional de la barra estará más oscura. El porcentaje de barra oscurecida

debería corresponder al porcentaje de tarea completa. Las barras más claras simbolizan tareas que no han sido empezadas. A continuación, se trazara una línea vertical perpendicular al eje horizontal y que cortará a éste en la fecha del día. Entonces, se puede evaluar el avance del proyecto.

La herramienta utilizada en la planificación del módulo BQG, Microsoft Office Project, brinda el diagrama de gantt, representando los recursos asignados por tareas, el tiempo de duración de cada tarea, datos específicos de la misma, y la dependencias de subtareas. **Ver Anexo 3.**

3.5 Método PERT: Principios básicos

PERT (Técnica de Revisión y Evaluación de Programas sus siglas en Inglés Program Evaluating and Review Technique), comúnmente abreviada como PERT, es un modelo para la administración y gestión de proyectos inventado en 1958 por la Oficina de Proyectos Especiales de la Marina de Guerra del Departamento de Defensa de los EE.UU. como parte del proyecto Polaris (UGM-27 Polaris - misil lanzado desde submarino diseñado por la marina estadounidense durante la Guerra Fría.) Este proyecto fue una respuesta directa a la crisis del Sputnik (El **Programa Sputnik** fue una serie de misiones espaciales no tripuladas lanzadas por la Unión Soviética a finales de los años 1950 para demostrar la viabilidad de los satélites artificiales en órbita terrestre. El nombre "Sputnik" ("Спутник") viene del ruso y su significado es "satélite" o "camarada viejo").

PERT es básicamente un método para analizar las tareas involucradas en completar un proyecto dado, especialmente el tiempo para completar cada tarea, e identificar el tiempo mínimo necesario para completar el proyecto total.

Este modelo de proyecto fue el primero de su tipo, un consuelo para la administración científica, fundada por el fordismo (es el modo de producción en cadena que llevó a cabo Henry Ford) y el taylorismo (es un método para aumentar la productividad y evitar el control del desarrollador en tiempos de producción, iniciado por Frederick W. Taylor). A pesar de que cada compañía tiene su propio modelo de proyectos, todos se basan en PERT de algún modo.

La parte más famosa de PERT son las Redes PERT, diagramas de líneas de tiempo que se interconectan. PERT está diseñado para proyectos de gran escala, que se ejecutan de una vez, complejos y no rutinarios.

Redes PERT

PERT permite planificar y controlar el desarrollo de un proyecto. Las redes PERT, trabajan con tiempos probabilísticos. Normalmente para desarrollar un proyecto específico lo primero que se hace es determinar, en una reunión multidisciplinaria, cuales son las actividades que se deberá ejecutar para llevar a feliz término el proyecto, cuál es la precedencia entre ellas y cuál será la duración esperada de cada una.

Para definir la precedencia entre actividades se requiere de una cierta cuota de experiencia profesional en el área, en proyectos afines.

Duración de una Actividad

Para estimar la duración esperada de cada actividad es también deseable tener experiencia previa en la realización de tareas similares. En planificación y programación de proyectos se estima que la duración esperada de una actividad es una variable aleatoria de distribución de probabilidad Beta Unimodal” de parámetros (a, m, b) donde:

t_a = Se define como el tiempo optimista al menor tiempo que puede durar una actividad.

t_m = Es el tiempo más probable que podría durar una actividad.

(Este corresponde al tiempo CPM, asumiendo que los cálculos son exactos).

t_b = Éste es el tiempo pesimista, o el mayor tiempo que puede durar una actividad.

t_e = Corresponde al tiempo esperado para una actividad.

El valor (o tiempo) esperado en esta distribución se expresa en la siguiente fórmula:

$$t_e = \frac{t_a + 4t_m + t_b}{6}, \text{ cuya variabilidad está dada por:}$$

$$\sigma^2 = \left(\frac{t_b - t_a}{6} \right)^2, \text{ y una desviación estándar:}$$

$$\sigma = \frac{t_b - t_a}{6}$$

En un dibujo PERT se puede distinguir nodos y arcos. Los nodos representan instantes en el tiempo. Específicamente, representan el instante de inicio de una o varias actividades y simultáneamente el instante de término de otras varias actividades. Los arcos por su parte representan las actividades, tienen un nodo inicial y otro de término donde llega en punta de flecha. Asociada a cada arco está la duración esperada de la actividad. Más información de un diagrama de actividades es representar éstas con una valoración de complejidad para minimizar el efecto de cuello de botella.

Dibujo PERT

Existen dos metodologías aceptadas realizar un dibujo PERT, la de “Actividad en el Arco” y las de “Actividad en el Nodo”, siendo ésta última la más utilizada en la actualidad en atención a que es la que usan la mayoría de las aplicaciones computacionales especialistas en este tema.

Cada nodo contiene la siguiente información sobre la actividad:

- Nombre de la actividad
- Duración esperada de la actividad (t)
- Tiempo de inicio más temprano (ES = Earliest Start)
- Tiempo de término más temprano (EF = Earliest Finish)
- Tiempo de inicio más tardío (LS = Latest Start)
- Tiempo de término más tardío (LF = Latest Finish)
- Holgura de la Actividad (H)

Por convención los arcos se dibujan siempre con orientación hacia la derecha, hacia el nodo de término del proyecto, nunca retrocediendo. El dibujo PERT se comienza en el nodo de inicio del proyecto. A partir de él se dibujan las actividades que no tienen actividades precedentes, o sea, aquellas que no tienen que esperar que otras actividades terminen para poder ellas iniciarse. A continuación se dibujan las restantes actividades cuidando de respetar la precedencia entre ellas. Al terminar el dibujo preliminar, existirán varios nodos ciegos, nodos terminales a los que llegan aquellas actividades que no son predecesoras de ninguna otra, es decir aquellas que no influyen en la fecha de inicio de ninguna otra, éstas son las actividades terminales y concurren por lo tanto al nodo de término del proyecto.

Cálculo de los tiempos de inicio y término más tempranos

El tiempo de inicio más temprano “ES” (Earliest Start) y de término más temprano “EF” (Earliest finish) para cada actividad del proyecto, se calculan desde el nodo de inicio hacia el nodo de término del proyecto según la siguiente relación:

$$EF = ES + t$$

Donde (t) es el tiempo esperado de duración de la actividad y donde ES queda definida según la siguiente regla:

- Regla del tiempo de inicio más temprano:

El tiempo de inicio más temprano, ES, de una actividad específica, es igual al mayor de los tiempos EF de todas las actividades que la preceden directamente.

El tiempo de inicio más temprano de las actividades que comienzan en el nodo de inicio del proyecto es cero (0).

Duración esperada del proyecto

La duración esperada del proyecto (T) es igual al mayor de los tiempos EF de todas las actividades que desembocan en el nodo de término del proyecto.

Cálculo de los tiempos de inicio y término más tardíos

El tiempo de inicio más tardío “LS” (Latest Start) y de término más tardío “LF” (Latest finish) para cada actividad del proyecto, se calculan desde el nodo de término retrocediendo hacia el nodo de inicio del proyecto según la siguiente relación:

$$LS = LF - t$$

Donde (t) es el tiempo esperado de duración de la actividad y donde LF queda definida según la siguiente regla:

- Regla del tiempo de término más tardío:

El tiempo de término más tardío, LF, de una actividad específica, es igual al menor de los tiempos LS de todas las actividades que comienzan exactamente después de ella.

El tiempo de término más tardío de las actividades que terminan en el nodo de término del proyecto es igual a la duración esperada del proyecto (T).

Holguras, actividades críticas y rutas críticas

- Holgura

Se denomina holgura de una actividad, al tiempo que tiene ésta disponible para, ya sea, atrasarse en su fecha de inicio, o bien alargarse en su tiempo esperado de ejecución, sin que ello provoque retraso alguno en la fecha de término del proyecto.

La holgura de una actividad se calcula de la siguiente forma:

$$H = LF - EF, \text{ o bien, } H = LS - ES$$

- Actividades críticas

Se denomina actividades críticas a aquellas actividades cuya holgura es nula y que por lo tanto, si se retrasan en su fecha de inicio o se alargan en su ejecución más allá de su duración esperada, provocarán un retraso exactamente igual en tiempo en la fecha de término del proyecto.

- Rutas críticas

Se denomina rutas críticas a los caminos continuos entre el nodo de inicio y el nodo de término del proyecto, cuyos arcos componentes son todas actividades críticas.

Las rutas críticas se nombran por la secuencia de actividades críticas que la componen o bien por la secuencia de nodos por los que atraviesa.

Un proyecto puede tener más de una ruta crítica, pero al menos tendrá siempre una.

Variabilidad de la duración de un proyecto

La duración esperada del proyecto (T) es una variable aleatoria proveniente de la suma de otras variables aleatorias, las duraciones esperadas de las actividades de la o las rutas críticas del proyecto y por lo tanto su variabilidad dependerá de la variabilidad de todas las actividades críticas del proyecto. Se tiene entonces que la varianza y la desviación estándar de la duración esperada del proyecto está dada por:

$$\sigma_t^2 = \sum \text{Varianza de todas las actividades del Proyecto}$$

Cálculo de probabilidades

Asumiendo que la duración esperada de una actividad es una variable aleatoria independiente, que la duración esperada del proyecto es una variable aleatoria de distribución aproximadamente normal. Se pueden calcular algunas probabilidades haciendo uso de una tabla de distribución normal, tomando en consideración las siguientes relaciones:

La probabilidad de que el proyecto se termine antes de una duración dada t_0 está dada por:

$P \{T \leq t_0\} = P \{Z \leq z_0\}$, donde z_0 es el valor de entrada a una tabla de distribución normal y que se calcula según:

$$z_0 = \frac{t_0 - T}{\sigma_T}$$

(PERT 2006)

3.6 Método de la ruta crítica

El método de la ruta crítica fue inventado por la corporación DuPont (empresa multinacional de origen estadounidense (DuPont 2007)) y es comúnmente abreviado como CPM por las siglas en inglés de Critical Path Method. En administración y gestión de proyectos, una ruta crítica es la secuencia de los elementos terminales de la red de proyectos con la mayor duración entre ellos, determinando el tiempo más corto para completar el proyecto. La duración de la ruta crítica determina la duración del proyecto entero. Cualquier retraso en un elemento de la ruta crítica afecta la fecha de término planeada del proyecto, y se dice que no hay holgura en la ruta crítica.

Un proyecto puede tener varias rutas críticas paralelas. Una ruta paralela adicional a través de la red con las duraciones totales menos cortas que la ruta crítica es llamada una sub-ruta crítica.

Originalmente, el método de la ruta crítica consideró solamente dependencias entre los elementos terminales. Un concepto relacionado es la cadena crítica, la cual agrega dependencias de recursos. Cada recurso depende del manejador en el momento donde la ruta crítica se presente.

A diferencia de la técnica de revisión y evaluación de programas (PERT), el método de la ruta crítica usa tiempos ciertos (reales o determinísticos). Sin embargo, la elaboración de un proyecto en base a redes CPM y PERT son similares y consisten en:

Identificar todas las actividades que involucra el proyecto, lo que significa, determinar relaciones de precedencia, tiempos técnicos para cada una de las actividades.

Construir una red con base en nodos y actividades (o arcos, según el método más usado), que implican el proyecto.

Analizar los cálculos específicos, identificando las rutas críticas y las holguras de los proyectos.

En términos prácticos, la ruta crítica se interpreta como la dimensión máxima que puede durar el proyecto y las diferencias con las otras rutas que no sean la crítica, se denominan tiempos de holgura.

3.7 Propuesta

El objetivo de este trabajo es que se realicen las actividades, especificadas en cada capítulo, en los restantes módulos del proyecto Gestión Hospitalaria. La técnica de evaluación y revisión del programa y el método de la ruta crítica, se han implementado en la herramienta Microsoft Office Project. Project es una herramienta factible, hace más práctica éstas técnicas y está disponible en la universidad.

La planificación temporal es la culminación de una actividad de planificación, componente primordial de la dirección de proyectos software. Cuando se combinan métodos de estimación y análisis de riesgo, la planificación temporal se convierte en un mapa de carreteras a seguir por el gestor del proyecto.

La planificación temporal empieza con la descomposición del proceso. Las características del proyecto se emplean para adaptar un conjunto de tareas apropiado al trabajo a realizar. Una red de tareas muestra todas las tareas de ingeniería, sus dependencias con otras tareas y sus duraciones previstas.

La red de tareas se usa para calcular el camino crítico del proyecto, un gráfico de tiempo e información diversa del proyecto. El planificador del proyecto puede seguir y controlar todos los pasos del proceso de software usando la planificación temporal como guía.

3.8 Ventajas de la planificación

Existen muchas ventajas para realizar la planificación, que deben estimular a los directivos en todos los niveles de cualquier organización. Entre las cuales se pueden mencionar:

La planificación requiere de actividades en orden y propósito, enfocándose todas las actividades hacia los resultados deseados, lográndose una efectiva secuencia de los esfuerzos. Señala además la necesidad de cambios posteriores, ayudando al jefe del módulo, a visualizar las futuras posibilidades. Contesta a las preguntas ¿y que pasa si?, estas preguntas le permiten al planificador, divisar los posibles planes de contingencia. Proporciona una base para el control que se ejecuta, garantizando que la planificación brinde los resultados deseados. Estimula la realización; el hecho de poner los pensamientos en el papel y formular un plan proporciona al que hace los planes la orientación y el impulso de realizar los objetivos.

En resumen el planificador del Proyecto de Software tiene que estimar tres cosas antes de que comience el proyecto: cuánto durara, cuánto esfuerzo requerirá y cuántas personas estarán implicadas. Además el planificador debe predecir los recursos de hardware y software que va a requerir y el riesgo implicado. El conjunto de la técnica PERT y el método del camino crítico (CPM) proporcionan herramientas cuantitativas que permiten determinar: 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 y los márgenes de tiempo que se dan para cada tarea o actividad individual y que no impliquen un retraso del proyecto.

CONCLUSIONES

1. Se realizó un estudio del estado del arte de la planificación, en la ingeniería del software. Obteniendo como resultado, que no es muy utilizada en grandes empresas desarrolladoras de software. Aunque existen algunas que si lo han realizado, pero no de la forma adecuada.
2. En la Universidad de las Ciencias Informáticas, los proyectos llevan a cabo la planificación de forma ineficiente lo que conlleva al atraso de la entrega del software.
3. Se llevó a cabo una investigación de las diferentes herramientas que existen. En la mayoría de los casos se utilizan herramientas diseñadas por el mismo equipo de desarrollo, para su mejor entendimiento.
4. Se efectuó un estudio minucioso, para obtener un procedimiento de planificación, que sea claro sencillo y profundo. Además, se definieron los riesgos que pueden afectar al proyecto, obteniendo una lista de riesgos con sus métodos de reducción y supervisión, evaluando su impacto y su probabilidad de ocurrencia.
5. Se definió un método de estimación, fácil, y que se puede realizar en todas las etapas del proyecto, dándole un control a lo estimado y verificando el tiempo de duración del proyecto, evaluando los factores ambientales y técnicos que influyen en el proceso de desarrollo del sistema.
6. Se demostró con un ejemplo en el módulo Bloque Quirúrgico General, que puede realizarse la planificación enfocada en el procedimiento propuesto.

RECOMENDACIONES

Por la importancia de sus funciones, en un equipo de desarrollo de software, el planificador, realiza algunas importantes recomendaciones, a la dirección del proyecto Gestión Hospitalaria, y a la dirección de la facultad, con el objetivo de extender el trabajo realizado a todos los desarrolladores e implementar el procedimiento propuesto:

Como primera recomendación y objetivo principal de este trabajo se exhorta a desarrollar el procedimiento propuesto para una planificación detallada y prolongada, en los restantes módulos del equipo de desarrollo GeHos, para definir la estimación general del mismo.

Definir a inicios del proyecto los diferentes roles que existen, especialmente el de planificador, explicándoles las funciones que debe realizar y su objetivo dentro del proceso de desarrollo de software.

REFERENCIAS BIBLIOGRÁFICAS

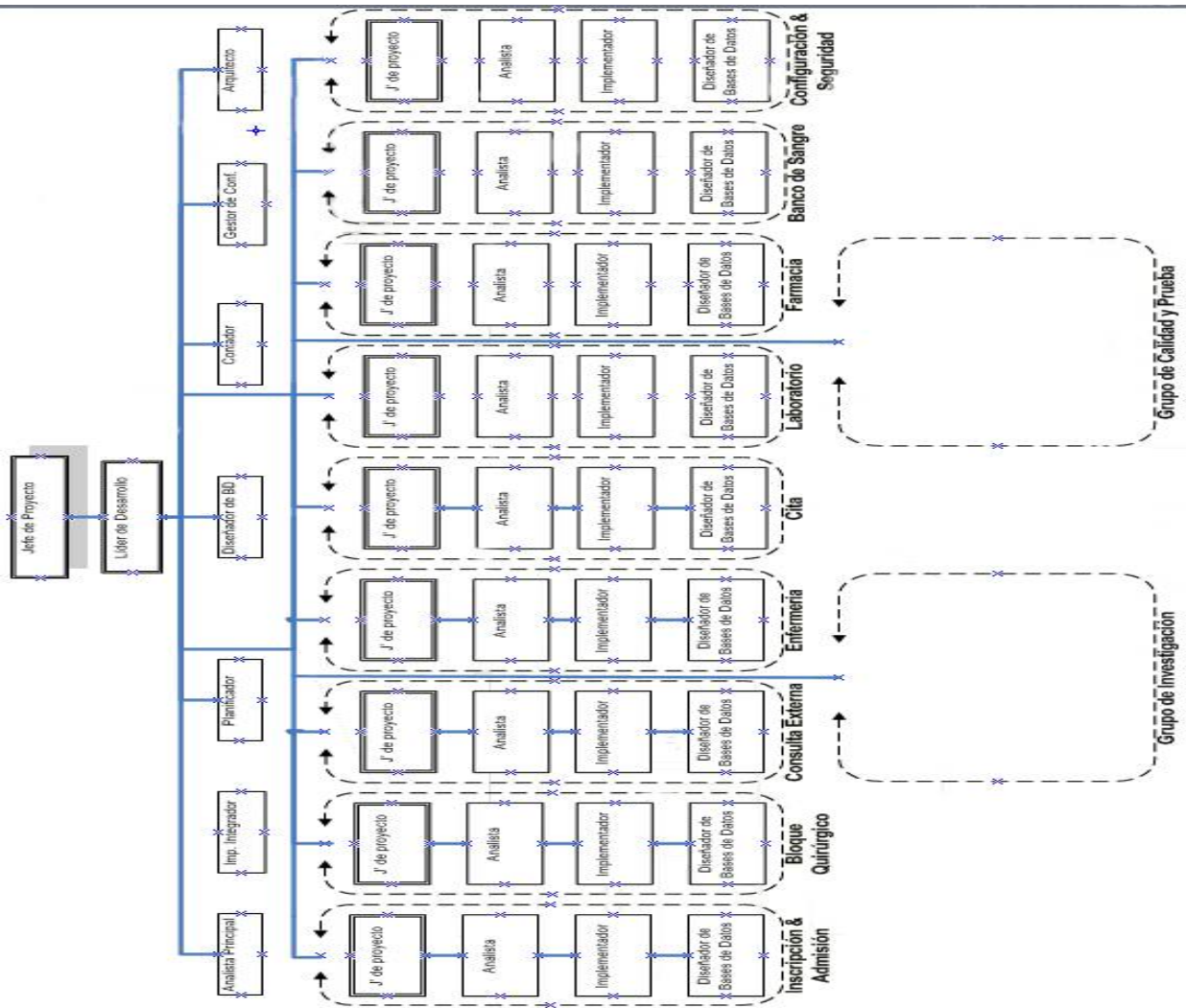
- ARMAS, E. F. Y. R. Planificación, 2006. [Disponible en:
<http://www.monografias.com/trabajos34/planificacion/planificacion.shtml>
- BARZANALLANA, R. M. Y. A. Reducción, supervisión y gestión del riesgo., 2007. [2007]. Disponible en:
http://www.wikilearning.com/reduccion_supervision_y_gestion_del_riesgo-wkccp-3620-13.htm
- COCOMO. 2007. [Disponible en: <http://es.wikipedia.org/wiki/COCOMO>
- DuPont. 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/DuPont>
- LOMA-OSORIO, G. F. Capítulo 1.3.El proyecto en el marco de la planificación del desarrollo., 1ra parte].
Disponible en:
http://www.dpi.upv.es/nueva/tesis/tesis_ferrero/Cap-1-3-Proyecto_y_Planificacion_Desarrollo.pdf
- MICROSOFT. Encarta, 1993-2006. [Disponible en:
Para diagrama de flujos y presentaciones. 2007. [Disponible en:
<http://www.navegapolis.net/content/blogcategory/0/85/10/10/>
- PERT. 2006. (TUTOS). Disponible en: <http://es.wikipedia.org/wiki/PERT>
- Subversion. Disponible en: <http://es.wikipedia.org/wiki/SVN>
- ToDoList. 2007. [Disponible en: <http://www.navegapolis.net/content/blogcategory/0/85/10/10/>
- Trac: Gestión de proyectos y de configuración integrados. 2006. [Disponible en:
<http://www.navegapolis.net/content/view/359/87/>
- TUTOS. 18.07.2006 [2007]. Disponible en: <http://www.navegapolis.net/content/view/385/87/>

BIBLIOGRAFÍA

1. ARMAS, E. F. Y. R. Planificación, 2006. [Disponible en: <http://www.monografias.com/trabajos34/planificacion/planificacion.shtml>]
2. BARZANALLANA, R. M. Y. A. Reducción, supervisión y gestión del riesgo., 2007. [2007]. Disponible en: http://www.wikilearning.com/reduccion_supervision_y_gestion_del_riesgo-wkccp-3620-13.htm
3. COCOMO. 2007. [Disponible en: <http://es.wikipedia.org/wiki/COCOMO>]
4. Dotproject. 2006. [2007]. Disponible en: <http://www.desarrolloweb.com/scripts/dotproyect-gestion-proyectos-php.html>
5. DuPont. 2007. [2007]. Disponible en: <http://es.wikipedia.org/wiki/DuPont>
6. Gestión de riesgos. 2007]. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=1784>
7. HERNÁNDEZ, S. E. B. Métricas de estimación de tamaño. Puntos de casos de uso, 2002.
8. HILLSON, D. Describiendo el riesgo: Cuántos detalles?, 2004. [2007]. Disponible en: <http://www.allpm.com/allpmnewsletter/August2004/09S%20Describiendo%20el%20riesgo%20-%20cuantos%20detalles.pdf>
9. JUAN M. PIKATZA, I. U. L., FRANCISCO J. SOBRADO, JUAN J. GARCÍA. Modelo jerárquico de roles para organizaciones de pequeño tamaño., 2007]. Disponible en: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-88/paper4.pdf>
10. LOMA-OSORIO, G. F. Capítulo 1.3. el proyecto en el marco de la planificación del desarrollo., 1ra parte]. Disponible en: http://www.dpi.upv.es/nueva/tesis/tesis_ferrero/Cap-I-3-Proyecto_y_Planificacion_Desarrollo.pdf
11. MICROSOFT. Encarta, 1993-2006.
12. PREESMAN, R. S. Ingeniería del software, un enfoque práctico. Quinta. La habana, 2005. 77-127 p.
13. Para diagrama de flujos y presentaciones. 2007. [Disponible en: <http://www.navegapolis.net/content/blogcategory/0/85/10/10/>]
14. PERT. 2006. [2006]. Disponible en: <http://es.wikipedia.org/wiki/PERT>
15. Planificación temporal, el método PERT. Disponible en: <http://www.uhu.es/eyda.marin/apuntes/gesempre/tema9>
16. RAFAEL MENÉNDEZ, A. B. Gestión de riesgos del software. Disponible en: http://www.wikilearning.com/gestion_de_riesgos_en_ingenieria_del_software-wkc-3620.htm
17. Subversion. Disponible en: <http://es.wikipedia.org/wiki/SVN>
18. THEMPRA. Sistema de un control de versiones, 2007. [2007]. Disponible en: <http://labloguera.net/blogs/themptra/archive/2007/04/10/sistemas-de-control-de-versiones.aspx>
19. ToDoList. 2007. [Disponible en: <http://www.navegapolis.net/content/blogcategory/0/85/10/10/>]

20. Trac: Gestión de proyectos y de configuración integrados. 2006. [Disponible en:
<http://www.navegapolis.net/content/view/359/87/>
21. TUTOS. 18.07.2006 [2007]. Disponible en: <http://www.navegapolis.net/content/view/385/87/>
22. VALER, J. I. La gerencia de riesgos - Factor crítico, 2002. [2007]. Disponible en:
http://www.willydev.net/descargas/WillyDev_GerenciadeRiesgosFactorCriticodeExito.pdf

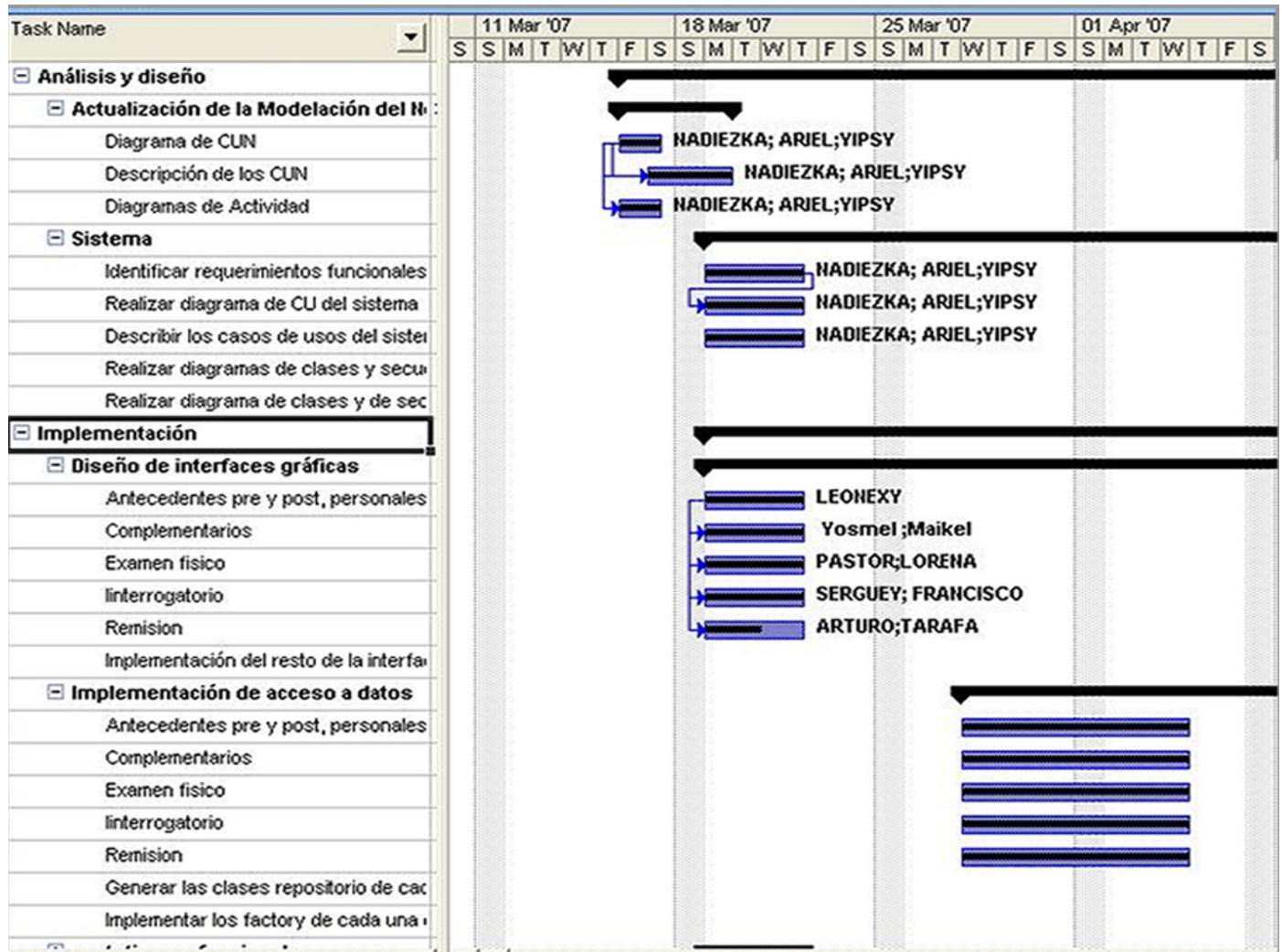
ANEXO 1



ANEXO 2

COMPONENTES		RENDIMIENTO	SOPORTE	COSTE	PLANIFICACIÓN TEMPORAL
CATEGORÍA					
CATASTRÓFICA	1	Dejar de cumplir los requisitos provocaría el fallo de la misión		Malos resultados en un aumento de costes y retrasos de la planificación temporal con gastos que superan las £500.000	
	2	Degradación significativa para no alcanzar el rendimiento técnico	El software no responde o no admite soporte	Recortes financieros significativos, presupuestos excedidos	Fecha de entrega inalcanzable
CRÍTICA	1	Dejar de cumplir los requisitos degradaría el rendimiento del sistema hasta donde el éxito de la misión es cuestionable		Malos resultados en retrasos operativos y/o aumento de coste con un valor esperado de £100.000 a £500.000	
	2	Alguna reducción en el rendimiento técnico	Pequeños retrasos en modificaciones de software	Algunos recortes de los recursos financieros, posibles excesos del presupuesto	Posibles retrasos en la fecha de entrega
MARGINAL	1	Dejar de cumplir los requisitos provocaría la degradación de la misión secundaria		Los costes, impactos y/o retrasos recuperables de la planificación temporal con un valor estimado de £1.000 a £100.000	
	2	De mínima a pequeña reducción en el rendimiento técnico	El soporte del software responde	Recursos financieros suficientes	Planificación temporal realista, alcanzable
DESPRECIABLE	1	Dejar de cumplir los requisitos provocaría inconvenientes o impactos no operativos		Los errores provocan impactos mínimos en el coste y/o planificación temporal con un valor esperado de menos de £1.000	
	2	No hay reducción en el rendimiento técnico	Software fácil de dar soporte	Posible superávit de presupuesto	Fecha de entrega fácilmente alcanzable

ANEXO 3



GLOSARIO DE TÉRMINOS

Gestor: Persona de una empresa que participa en la administración de la misma.

Hardware: Conjuntos de componentes que integran la parte material de una computadora. Componentes físicos de una tecnología.

IBM: International Business Machines o IBM, conocida coloquialmente como el Gigante Azul, es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911, pero lleva operando desde 1888.

IDE: Un entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE') es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Microsoft: Es una empresa multinacional americana fundada en 1975 dedicada al sector de las tecnologías de la información con sede en Redmond, Washington, Estados Unidos. Es mayormente conocida por sus sistemas operativos Windows y aplicaciones ofimáticas como Microsoft Office.

Proyecto: Un proyecto es un esfuerzo temporal emprendido para crear un producto o un servicio único. Así, el resultado final buscado puede diferir con la misión de la organización que la emprende, ya que el proyecto tiene determinado específicamente un plazo y el esfuerzo es temporal.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Linux: Sistema Operativo abierto y libre creado por el finlandés Linus Torvalds.

Megabyte (MB): Unidad de medida de cantidad de datos informáticos.