

UNIVERSIDAD
DON BOSCO



**METODOLOGÍA PARA LA DETERMINACIÓN DE COSTOS EN LA
REALIZACIÓN DE SISTEMAS DE INFORMACIÓN**

**TRABAJO DE GRADUACIÓN
PREPARADO PARA LA FACULTAD DE INGENIERÍA
PARA OPTAR AL GRADO DE
INGENIERO EN CIENCIAS
DE LA COMPUTACIÓN.**

POR

**RENE ARTURO ANGULO ARRIAZA.
MARIO ENRIQUE CONTRERAS ACUÑA.**

FEBRERO DE 1997

SAN SALVADOR,

EL SALVADOR,

CENTRO AMÉRICA

UNIVERSIDAD
DON BOSCO

RECTOR
ING. FEDERICO MIGUEL HUGUET.

SECRETARIO GENERAL
PADRE PEDRO JOSE GARCIA CASTRO.

DECANO DE LA FACULTAD DE INGENIERIA
ING. CARLOS ALBERTO GUTIERREZ.

ASESOR DE TRABAJO DE GRADUACION:
LIC. RENE HECTOR MARTINEZ.

JURADO EXAMINADOR


ING. NELSON TESORERO.
LIC. ROBERTO RIVAS BAYONA

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERIA

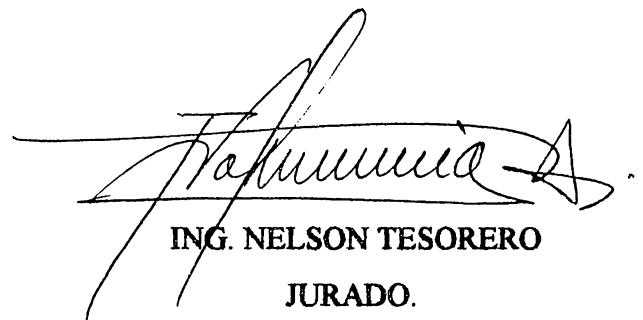
DEPARTAMENTO DE INGENIERIA EN SISTEMAS.

JURADO EVALUADOR DEL TRABAJO DE GRADUACION:
“METODOLOGÍA PARA LA DETERMINACIÓN DE COSTOS EN LA
REALIZACIÓN DE SISTEMAS DE INFORMACIÓN”



LIC. ROBERTO RIVAS BAYONA

JURADO.



ING. NELSON TESORERO

JURADO.



LIC. RENE HECTOR MARTINEZ.

ASESOR.

A DIOS TODO PODEROSO:
POR LA OPORTUNIDAD, TALENTO Y SABIDURIA QUE ME HA DADO
PARA CORONAR ESTA CARRERA.

A MIS PADRES:
ENRIQUE Y BERTHA, PORQUE ESTE LOGRO ES DE UDS., POR TODOS
SUS SACRIFICIOS, ESFUERZOS Y COMPRESION EN TODO MOMENTO
Y A QUIENES LES GUARDO UN PROFUNDO AMOR, RESPETO Y
ADMIRACION.

A MIS HERMANAS:
CAROLINA Y ANA RUTH, POR BRINDARME TODO SU CARIÑO Y
APOYO, GRACIAS.

A MI ABUELITA:
MAMATINA, POR TODO SU AMOR Y APOYO, DIOS LA BENDIGA.

A MI COMPAÑERO DE TRABAJO:
RENE Y SU FAMILIA, POR SU AMISTAD Y APOYO PARA FINALIZAR
ESTE TRABAJO

A TODOS AQUELLOS QUE DE UNA U OTRA FORMA ME AYUDARON
GRACIAS.

MARIO

A DIOS TODO PODEROS Y LA SANTISIMA VIRGEN:
POR DARME LA SABIDURIA NECESARIA PARA LA CULMINACION DE
MI CARRERA.

A MIS PADRES:
POR SER UN EJEMPLO VIVO DE AMOR, SABIDURIA Y COMPRESION.
Y A QUIENES LOS APRECIO PROFUNDAMENTE.

A MI ESPOSA:
POR TODO SU APOYO Y COMPRESION EN ESTOS CINCO AÑOS DE MI
CARRERA

A MIS HERMANOS:
WILFREDO Y VERONICA, PORQUE ME APOYARON DIRECTA O
INDIRECTAMENTE, CON LAS PALABRAS ADECUADAS DE ANIMO EN EL
MOMENTO PRECISO.

A MI COMPAÑERO DE TESIS:
MARIO, POR TODA LA AMISTAD, APOYO Y COMPRESION QUE ME
BRINDO, Y A SU FAMILIA POR TODAS SUS ORACIONES.

Y A TODAS LAS PERSONAS QUE DE UNA U OTRA FORMA ME
APOYARON..... GRACIAS.

RENE.

Y UN AGRADECIMIENTO ESPECIAL:
AL JURADO EXAMINADOR; LIC. ROBERTO RIVAS E ING. NELSON
TESORERO, POR ACEPTAR SER PARTE DE NUESTRO TRABAJO A
TRAVEZ DE SU EVALUACION Y CORRECCION
Y DE MANERA ESPECIAL AL
LIC. RENE MARTINEZ, QUIEN SE DESEMPEÑO COMO UN AUTENTICO
GUIA PARA LA REALIZACION DEL MISMO.

MARIO Y RENE.

INDICE.

Tema	Pág.
Introducción	I.
Indice	II.

CAPITULO I.

1.- Conceptos	1.
1.1.- Ciclo de vida del Sistema.	1.
a) Estudio de factibilidad.....	1.
b) Análisis de Requerimientos.....	4.
c) Diseño.....	9.
d) Codificación.....	14.
e) Prueba del Software.....	20.
f) Mantenimiento del software.....	21.
1.2.- Sistemas de Información	23.
1.2.1.- Definición de los datos, Información y sistemas de información.....	24
a) Datos.....	24.
b) Información.....	25.
c) Sistemas de Información.....	25.
1.3.- Reingeniería	26.
1.4.- Programación de proyectos con PERT-CPM	28.
1.5.- Caso de Estudio	32.
1.5.1.- Proceso de inscripción a: "Scientific American".....	32.
a) Descripción del sistema.....	32.
b) La solución del programa.....	34.
c) La perspectiva de la solución de	

programación económica.....	37.
-----------------------------	-----

CAPITULO II.

2.1.- Modelo Básico de COCOMO	43.
2.1.1.- Definiciones	43.
2.1.2.- Mes hombre versus Estimación en Efectivo.....	46.
2.1.3.- Desarrollo de esfuerzo y horario	47.
2.1.4.- Las ecuaciones que COCOMO utiliza	47.
2.2.- Ejemplo de estimación	49.
2.3.- El Banco Nacional Hunt, Sistema EFT	58.

CAPITULO III.

3.1.- Modelo Intermedio de COCOMO.....	61.
3.1.1.- Introducción	61.
3.1.2.- Desarrollo de la estimación del esfuerzo en el sistema.....	63.
a) Ecuaciones de escala nominal.....	63.
b) Factores de multiplicación en el desarrollo de sistemas.....	64.
3.1.3.- Ejemplo de costeo: " Sistema de comunicación de Microprocesadores".....	70.
a) Análisis de sensibilidad.....	73.
3.2.- Rentabilidad del proyecto	77.
3.2.1.- Evaluación Económica	77.
a) Valor presente neto.....	77.

b) Tasa Interna de rendimiento	78.
3.3.- Análisis de riesgo	79.
a) El riesgo medido en los flujos	81.
b) Fuentes de riesgo en Ingeniería de software	82.

CAPITULO IV.

Investigaciones y Conclusiones.

4.1.- Introducción	84.
4.2.- Objetivos	84.
4.3.- Espacio Muestral	85.
4.4.- Elección del Tipo de muestra	85.
4.5.- Tamaño de la muestra	85.
4.6.- Método de recolección de Datos	85.
4.7.- Formato de la Entrevista realizada	86.
4.8.- Resultados de la entrevistas dirigida	87.

CAPITULO V.

Metodología para la determinación de Costos en la Realización de Sistemas de Información.

5.1.- Introducción	94.
5.2.- Pasos de la Metodología	97.
I. Determinación de la cantidad de Módulos, entidades y relaciones del proyecto	97.
II. Determinación de la cantidad de meses; que se tardará en desarrollar el sistema	98.
III. Análisis comparativa de los tiempos reales	

con los tiempos estimados	99.
IV. Identificación de las variables en el estudio de factibilidad	100.
V. Establecimiento de las variables en el Análisis de requerimientos	101.
VI. Determinación de las variables en la fase de Diseño	103.
VII. Identificación de las variables en la fase de codificación	104.
VIII. Establecimiento de las variables en la fase de prueba	104.
IX. Determinación de las variables en la fase de implementación.	105.
X. Identificación de las variables en la fase de mantenimiento	106.
5.3.- Ejemplo de Costeo:	
"Realización de un sistema de Administración, financiero y Contable"	107.
5.3.1.- Introducción	107.
5.3.2.- Pasos de la metodología	108.
5.3.3.- Valores comparativos	116.
5.3.4.- Conclusiones	116.
5.3.5.- Resumen de fórmulas	119.
Anexos	122.
Tabla 2.3 Distribución de Actividades del proyecto por fases: Modo de encadenamiento.	123.
Tabla 2.4 Distribución de Actividades por fases: Modi Semi-Capturado	124.

Tabla 2.5 Distribución de Actividades

por fases: Modo Organico	125.
Diagrama Gannt	126.
Bibliografía	127.

INTRODUCCIÓN

Este documento contiene un estudio investigativo que se desarrolla bajo el título **METODOLOGÍA PARA LA DETERMINACIÓN DE COSTOS EN LA REALIZACIÓN DE SISTEMAS DE INFORMACIÓN**. Esta integrado por cinco capítulos:

En el primer capítulo se realiza un estudio conceptual del ciclo de vida del sistema de información, conceptualización de sistema de información y las técnicas de REINGENIERIA y redes PERT.

En el segundo capítulo, trata sobre el modelo básico de COCOMO, enfatizando el trabajo sobre todo en el diseño del sistema y la estimación de esfuerzo en el mantenimiento del software.

En el tercer capítulo se estudian metodologías que consideran líneas códigos (COCOMO intermedio) que consideran el desarrollo de una mejor estimación del costo del proyecto y se realiza un análisis de riesgo identificando las áreas de problemas potenciales.

En el cuarto capítulo, se considera el avance a lenguajes de cuarta generación (4Gls) y sus respectivas herramientas y se desarrolla una metodología considerando estos criterios. Finalmente en el capítulo cinco, se presenta la metodología diseñada para estimar los costos en la realización de sistemas de información, evaluando cada una de las fases del ciclo de vida de un sistema, facilitando la determinación si un proyecto es factible o no.

LOS AUTORES.

CAPITULO I.

1.- CONCEPTOS.

1.1.- CICLO DE VIDA DEL SISTEMA.

El ciclo de vida del sistema posee las siguientes fases:

- a) Estudio de factibilidad del sistema.
- b) Análisis de requerimientos.
- c) Diseño.
- d) Codificación.
- e) Prueba del Software.
- f) Mantenimiento del Software.

a) ESTUDIO DE FACTIBILIDAD DEL SISTEMA.

Las etapas de los sistemas de investigación, frecuentemente requieren de un estudio preliminar llamado " Factibilidad de estudio ", que consiste en investigar la información que el usuario necesita.

Los resultados de este estudio son formalizados en un reporte escrito, el cual incluye, especificaciones preliminares y un plan de desarrollo para el sistema propuesto. Este reporte es presentado a la gerencia para la firma, para ser firmado, en caso de ser aprobado, antes

de comenzar a desarrollarlo, en caso de que, la gerencia apruebe las recomendaciones de la factibilidad del estudio , las etapas del análisis del sistema pueden comenzar a evaluar las alternativas del sistema y de proponer el más factible. El estudio de factibilidad se puede dividir de la siguiente forma:

- Factibilidad de tiempo de respuesta.
- Factibilidad Económica.
- Factibilidad Técnica.
- Factibilidad Política.

Factibilidad Operativa.

FACTIBILIDAD DE TIEMPO DE RESPUESTA.

Se utiliza, para determinar si el sistema podrá o no habilitarse en el tiempo previsto. La factibilidad de tiempo de respuesta hace referencia a dos aspectos importantes:

- Si la respuesta de los sistemas no es la prevista, se dice,
- Si la finalización del sistema se da después de lo necesitado.

En ambos casos se dice que el sistema es infactible.

Con este estudio de factibilidad, podemos establecer, si un sistema solventa oportunamente una necesidad de la empresa.

FACTIBILIDAD ECONÓMICA.

En esta parte, la pregunta que debe responder es: "¿Un sistema provee una solución efectiva en relación al costo?"; para poder determinar la respuesta, tiene que realizarse una aproximación en base a costos y beneficios.

La determinación adecuada de los costos, nos presenta ciertas ventajas:

- Visualiza mejor la alternativa menos cara y que soluciona adecuadamente el problema.
- Se determina puntos de reducción de costos, que puedan realizarse sin necesidad de afectar en gran medida otros aspectos del proyecto.
- Muestra los aspectos cualitativos y cuantitativos que va a resolver el proyecto, bajo el criterio cuantificado del dinero que ahorra.

FACTIBILIDAD TÉCNICA.

El estudio de Factibilidad Técnica, trata de verificar la tecnología a utilizar en el proyecto. Con la

factibilidad técnica, se puede determinar, si un proyecto puede o no ser realizado con la tecnología actual, que la empresa posee. Al complementar con el estudio económico, podemos determinar, si se tiene que invertir en una nueva plataforma, ya sea HARDWARE Y/O SOFTWARE; y si esta inversión es rentable.

☑ FACTIBILIDAD POLÍTICA.

La Factibilidad Política, determina, si la solución o proyecto propuesto es conveniente en el entorno social, cultural y político, en el cual se desarrolla la empresa o departamento.

☑ FACTIBILIDAD OPERATIVA.

La factibilidad operativa u operacional, es la habilidad de la administración, empleados, etc. Para operar, usar y soportar un sistema propuesto. De tal forma, que en esta parte se determina si el recurso humano esta apto para adaptarse a un nuevo proyecto; así, se puede establecer otros parámetros relativos a estos, que determinan el costo en capacitar al personal para que utilice el nuevo sistema.

b) ANÁLISIS DE REQUERIMIENTOS.

La determinación de los requerimientos, es el estudio del sistema actual del negocio a fin de encontrar

¿ Cómo trabaja ? y ¿ Dónde debe mejorarse ?. Los estudios del sistema, son el resultado de una evaluación para conocer cómo funcionan los métodos actuales y si son necesarios o posibles algunos ajustes; se elaboran preguntas en relación con los sistemas manuales y los computarizados.

Un requerimiento, es una característica que debe incluirse en un nuevo sistema y puede consistir, en una forma de captar o procesar los datos, producir información , controlar una actividad de negocio o dar apoyo a la gerencia; por lo tanto, la determinación de los requerimientos, significa estudiar el sistema existente y recopilar los datos en relación con éste.

Dado que los analistas de sistema, no trabajan, como gerentes o empleados en los departamentos para usuario (como mercadotecnia, compras, producción ó contabilidad), no tienen los mismos conocimientos sobre hechos y datos, como los gerentes y usuarios de esas áreas; por lo tanto, un paso inicial en la investigación es entender la situación actual de la empresa o el departamento.

El estudio de requerimientos, posee las siguientes partes:

- 1.Requerimientos básicos.
- 2.Requerimientos de las transacciones de los usuario.
- 3.Requerimientos de decisión de los usuarios.

4.Requerimientos para toda la empresa.

b.1.-REQUERIMIENTOS BÁSICOS

Los analistas estructuran su investigación y buscan respuestas a las siguientes preguntas principales:

- ¿Cuál es el proceso básico ?.
- ¿ Qué datos se utilizan o se producen durante este proceso ?.
- ¿ Cuáles son los limites impuestos por tiempo y cantidad de trabajo ?.
- ¿ Qué controles de rendimiento se utilizan ?.

Las respuestas a estas preguntas se detallan a continuación:

¿ CUAL ES EL PROCESO BASICO ?

Se empezará con lo básico, se harán aquellas preguntas que proporcionen , los datos fundamentales y las descripciones del sistema. Las preguntas que se pueden realizar son:

- ¿Cuál es el propósito de esta actividad ?.
- ¿ Cuáles son los pasos que se realizan ?.
- ¿ Donde se realizan ?.
- ¿ Quién los ejecuta ?.
- ¿ Cuánto tiempo consumen ?.
- ¿ Con que frecuencia se realizan ?.

- ¿ Quién utiliza la información resultante ?.

¿ QUE DATOS SE UTILIZAN O SE PRODUCEN DURANTE ESTE PROCESO ?.

El analista necesita encontrar, los datos que se deben utilizar para realizar cada actividad.

La mayor parte de las transacciones de negocios, producen también, información que sirve a los gerentes, cuando evalúan a los empleados, al negocio y el rendimiento de los sistemas, y puede ser útil en otro contexto para el gerente y analista.

¿ CUALES SON LOS LIMITES IMPUESTOS POR TIEMPO Y CANTIDAD DE TRABAJO ?.

La frecuencia de las actividades del negocio, varía enormemente; por ejemplo, algunas actividades, como el pago de impuestos, se presentan unas cuantas veces al año, mientras que, pagar a los empleados, es una actividad semanal; por lo tanto, el analista debe aprender con que frecuencia se repite la actividad. Cuando el analista la conoce, puede usarla de guía para encontrar muchas preguntas adicionales e importantes y determinar la razón de la frecuencia y su efecto en las actividades del negocio.

¿ QUE CONTROLES DE RENDIMIENTO SE UTILIZAN ?

En las situaciones de negocios bien controladas, ya sea, por la gerencia o el control de procesos,

determinar, dónde se ha realizado una actividad en forma apropiada, puede, no tener problema; pero, durante la etapa de análisis , el analista debe examinar los métodos de control, respondiendo a las siguientes preguntas:

- ¿ Existen normas específicas de rendimiento ?.
- ¿Quién compara el rendimiento contra estas normas?.
- ¿ Cómo se descubren los errores ?.
- ¿ Cómo se manejan éstos ?.
- ¿ Y si son excesivos ?.

Encontrar controles débiles o faltantes, es un descubrimiento importante en cualquier investigación de sistemas.

b.2.-REQUERIMIENTOS DE LAS TRANSACCIONES DE LOS USUARIOS.

En esta parte, se debe determinar el nivel de transacción, es decir, la forma en que se captan, procesan y almacenan los datos.

b.3.-REQUERIMIENTOS DE DECISIÓN DE LOS USUARIOS.

Las decisiones, a diferencia de las actividades de transacción, pueden no seguir un procedimiento específico. Las rutinas no son tan claras y los controles pueden ser muy vagos. Las decisiones se hacen, al integrar información en forma tal, que los gerentes puedan saber qué acciones llevar a cabo. Los sistemas de

toma de decisiones pueden enfocarse hacia el pasado, el presente o el futuro.

b.4.-REQUERIMIENTOS PARA TODA LA EMPRESA.

En los negocios , los departamentos dependen entre si, para, proporcionar servicios, manufacturar productos y satisfacer a los clientes; por lo tanto, el trabajo que realiza un departamento afecta a los otros. Cuando los analistas estudian los sistemas, para un departamento, también, deben tomar en cuenta las implicaciones que haya en otros departamentos, que interactúan con el sistema que se investiga. Es responsabilidad de los analistas, identificar las dependencias entre departamentos y determinar cómo pueden afectar al proyecto.

c) DISEÑO

Para algunos Ingenieros en sistemas, el diseño, es el primer paso en la fase de desarrollo de sistemas. Esta etapa es definida como: "El proceso de aplicación de varias técnicas y principios, para el propósito de definición de ideas, un proceso ó un sistema, con suficiente detalle para permitir la realización física".

La meta de un diseñador, es producir un modelo o representación de una entidad que más tarde será construida. De forma física, el proceso por el cuál, el modelo se desarrolla, combina :

- Intuición y juicios basados en la experiencia en entidades similares, que anteriormente se crearon,
- Un conjunto de principios que guían al camino, en el cuál el modelo se desarrolla,
- Un conjunto de criterios que pueden ser juzgados, y
- Un proceso de iteración que guía la última representación diseñada.

La fase de diseño se divide en:

1. Fase de desarrollo y diseño de software.
2. El proceso de diseño.
3. Diseño fundamental.

c.1.- Fase de desarrollo y diseño de software.

Una fase de desarrollo, debe ser instituida sin hacer caso de ningún paradigma aplicado en la ingeniería de software.

Una vez los requerimientos del software, han sido establecidos, la fase de desarrollo comprende tres pasos diferentes :

Diseño, generación de código (manual o automático) y prueba.

Cada paso, transforma información en una manera que resulta finalmente, la validación del software de computadora.

El diseño de arquitectura, define, la relación de elementos de la estructura mayor del programa. El diseño de procedimiento, transforma estructuralmente los elementos dentro de una descripción de proceso de software. El código fuente, es, generado y probado, de tal forma que se da la integración y validación del software.

c.2.- El proceso de diseño.

El diseño de software, es un proceso, por el cuál, los requerimientos son trasladados a una representación del software. Desde el punto de vista gerencial del proyecto, el diseño de software es dirigido en dos pasos :

- El diseño detallado se enfoca sobre los refinamientos de la estructura de los datos y
- La representación algorítmica para el software.

c.3.-Diseño fundamental.

Un conjunto de conceptos fundamentales de diseño de software se ha desarrollado sobre las tres pasadas décadas. Los pasos del diseño fundamental son:

- Refinamiento.
- Arquitectura de software.
- Estructura de un programa.
- Estructura de datos.
- Procedimiento de software.

- Modularidad.
- Abstracción.
- Información oculta.

REFINAMIENTO.

El paso de refinamiento, es de las primeras estrategias de diseño propuesta por Niklaus Wirth.

La arquitectura de un programa, es desarrollado por niveles de refinamiento, exitosamente de un procedimiento detallado. Una jerarquía, es desarrollada por descomposición de una información macroscópica, hasta que se ha alcanzado la programación.

ARQUITECTURA DE SOFTWARE.

La arquitectura de software, se basa, en dos importantes características, de un programa de computadora:

- La estructura jerárquica de los componentes del procedimiento (módulos).
- Los datos estructurados.

La arquitectura de software es derivada, por medio de una partición del proceso que infieren la solución del sistema.

☑ ESTRUCTURA DE UN PROGRAMA.

La estructura de un programa, representa la organización de los componentes del mismo é implica una jerarquía de control.

☑ ESTRUCTURA DE DATOS.

La estructura de datos, es una representación de la relación lógica, entre los elementos individuales de datos.

☑ PROCEDIMIENTO DE SOFTWARE.

La estructura de programas, define jerárquicamente controles sin considerar la secuencia de procesamiento y decisiones. Los procesos deben proveer una especificación precisa de procesamiento, incluyendo, la secuencia de evento, puntos exactos de decisión, operaciones repetidas y aún organización de datos.

☑ MODULARIDAD.

El concepto de modularidad en software de computadoras, ha sido expuesto por casi cuatro décadas.

Se ha establecido que la "modularidad", es el atributo simple de software que permite que sea manejable intelectualmente.

☑ ABSTRACCION.

Cuando se considera, una solución modular para un problema, muchos niveles de abstracción, dan una solución amplia del ambiente del problema. Un nivel bajo de abstracción, toma orientación procedimental.

☑ INFORMACIÓN OCULTA.

El concepto de modularidad, guía cada diseño de software, para una pregunta fundamental:

¿ como se puede descomponer, una solución de software, para obtener los mejores conjuntos de módulos ?.

Los módulos deben ser especificados y diseñados de tal forma que la información (procedimientos y datos) contengan dentro de si, un módulo que este inaccesible para otros, que no necesitan tal información.

d) CODIFICACIÓN.**d.1.- LENGUAJES DE PROGRAMACIÓN Y CODIFICACIÓN.****d.1.1.- Lenguajes de programación.**

Todos los pasos de ingeniería de software, establecen la programación, como su objetivo final, de tal forma que

se pueda representar la traducción del software, en una forma, que, puede ser entendido por la computadora. Finalmente se alcanza el paso de codificación como un proceso que transforma el diseño en una programación.

La mayoría de aplicaciones, diseñadas todavía permanecen atrás del alcance de los de cuarta generación, de tal forma que en la actualidad, se usan lenguajes artificiales como ADA, FORTRAN, PASCAL, C, COBOL ó LENGUAJE ENSAMBLADOR.

Cuando se considera como un paso en el proceso de ingeniería de software, la codificación, es una secuencia natural de diseño. Aún las características de los lenguajes de programación y estilos de codificación pueden profundamente afectar la calidad de software así como su mantenimiento.

Existe un proceso llamado "procesos de traducción", el cual es considerado como el paso de la codificación, que traduce una representación detallada del diseño del software, dentro de la realización del lenguaje de programación.

El proceso de traducción continua cuando una compilación acepta el código fuente como entrada y produce dependencia de la máquina, la instrucción actual maneja la clave lógica en la unidad de procesamiento central.

a) LENGUAJES DE TERCERA GENERACIÓN.

Los lenguajes de tercera generación llamados: MODERNOS o LENGUAJES DE PROGRAMACIÓN ESTRUCTURADA, son caracterizados por procedimientos firmes y con capacidad de estructuración de datos. Los lenguajes en esta clase pueden ser divididos en dos amplias categorías: Lenguajes de alto orden de propósito general y Lenguajes especializados.

a.1. LENGUAJES DE ALTO ORDEN DE PROPÓSITO GENERAL

Entre estos tipos de lenguajes tenemos:

- **Algol**, ha servido como un modelo para otros lenguajes en esta categoría. Son descendientes, PL/1, PASCAL, MODULA-2, C Y ADA, son adoptados como lenguajes con potencial para aplicaciones de amplio espectro (para usar en ingeniería, ciencia, comercios y otras áreas de aplicación).
- **PL/1**, fue el primer lenguaje de amplio espectro, desarrollado con un amplio rango de aplicaciones futuras en diferentes áreas. PL/1 provee soporte para ingeniería convencional, científica y aplicaciones de negocios.
- **PASCAL**, es un lenguaje de programación moderna desarrollado en 1970 para enseñar técnicas modernas

(programación estructurada en el desarrollo de software).

- **MODULA-2**, es una evolución de PASCAL y una posible alternativa para el lenguaje de programación ADA. MODULA-2 maneja implementación directa de futuros diseños como una información oculta. Pero, por otra parte, las aplicaciones en la industria son limitadas.
- **EL LENGUAJE DE PROGRAMACIÓN C**, fue originalmente desarrollado como un lenguaje para implementación de sistemas operativos. C contiene futuras herramientas que pueden ser consideradas flexiblemente.

3.2. LENGUAJES ESPECIALIZADOS.

Los lenguajes especializados, son caracterizados por su forma sintáctica inusual, que han sido establecidos para aplicaciones diferentes.

Los lenguajes especializados proveen ventajas y desventajas, porque se han establecido para diseñar la dirección de una aplicación específica.

Por otra parte, existen los que se conocen como lenguajes de cuarta generación. En la década pasada, los lenguajes de cuarta generación (4GLs), han levantado el nivel de abstracción en el desarrollo de sistemas.

Los Lenguajes de cuarta generación, como los artificiales, contienen una sintaxis diferente para control y la representación de datos. Dentro de los existe una sofisticada clase de 4Gls, llamado generadores de programas. Un generador de programa facilita al usuario a crear completamente programas de tercera generación usando pocas declaraciones.

d.1.2.- CODIFICACION.

ESTILOS DE CODIFICACIÓN.

Los códigos deben ser entendibles. El Estilo de codificar abarca una filosófica codificación que sea simple y clara de comprender.

DOCUMENTACIÓN DE CÓDIGO.

La documentación interna del código fuente, comienza con la sección de identificación (variables y etiquetas) de nombres, continuando con la localización y composición de comentarios, es decir, la organización visual del programa. Seleccionar nombres de identificadores significativos es crucial para entenderlos.

La habilidad para expresar comentarios del lenguaje natural como parte del listado de código fuente, es proveído por todo lenguaje de programación de propósito

general. Sin embargo, ciertas preguntas surgen en esta parte:

- ¿ Cuántos comentarios son suficientes ?
- ¿ Donde deben colocarse los comentarios ?
- ¿Hacen un flujo lógico los comentarios oscuros?
- ¿ Pueden los comentarios guiar al lector ?

Hay pocas respuestas definitivas acerca de estas preguntas. Pero una cosa es clara, los software deben contener documentación interna. Los comentarios proveen un desarrollo significativo de comunicación con otros lectores del código fuente.

☑DECLARACIÓN DE DATOS.

La complejidad y organización de la estructura de datos se definen durante el diseño de este paso. El estilo de la declaración de datos se establece cuando el código es generado. Un número de normas relativamente simple pueden ser establecidas para hacer datos más entendibles y de simple mantenimiento.

El orden de declaración de datos podría ser estandarizado aún si el lenguaje de programación no tiene requerimientos de mandato asignados.

✓ CONSTRUCCIÓN DE LA DECLARACIÓN.

La construcción del flujo lógico del software se establece durante el diseño y las declaraciones individuales son parte del paso de codificación.

✓ ENTRADA/SALIDA.

El estilo de entrada y salida es establecido durante el análisis de requerimientos de software y diseño.

3) PRUEBA DEL SOFTWARE.

La prueba del software, es un elemento crítico de la calidad del mismo, y representa la última revisión de especificación, diseño y codificación.

✓ FUNDAMENTOS DE LA PRUEBA DE SOFTWARE.

Durante la fase de definición y desarrollo, el ingeniero trata de formar el software, de un concepto abstracto, a una implementación tangible. Se trata entonces, de crear una serie de pruebas que intenten demoler el software que se ha construido. En realidad, la prueba es un paso en el proceso de ingeniería de software que puede ser visto, como destructivo, más que constructivo.

Los desarrolladores de software son, personas constructivas por naturaleza. La Prueba requiere que el desarrollador descarte preconcebidamente la noción de la "corrección" del desarrollador de software, y , elimine el conflicto de intereses que ocurre cuando los errores son descubiertos.

Los objetivos de una prueba de software son:

- La Prueba es un proceso de ejecución de un programa con el intento de encontrar un error.
- Una buena prueba es la que tiene alta probabilidad de encontrar todavía un error.
- Una prueba exitosa es la que descubre errores que todavía no han sido descubiertos.

Los objetivos sobre todo implican un cambio dramático. El objetivo, entonces, es diseñar una prueba, que sistemáticamente descubra diferentes clases de error y hacerla con la mínima cantidad de tiempo y esfuerzo.

MANTENIMIENTO DE SOFTWARE.

El mantenimiento de software, se ha caracterizado, como la "montaña de hielo" : se espera que sea inmediatamente visible, todo lo que este ahí. Realmente, se sabe que, en la mayoría de problemas potenciales y costos, los usuarios ven superficialmente.

El mantenimiento de un software, existente puede sobrepasar el 60% de todo el esfuerzo consumido, por el desarrollo de la organización. El porcentaje continúa ascendiendo a medida que es producido el software.

Las actividades en la fase de mantenimiento se pueden clasificar de la siguiente manera:

- La primera actividad de mantenimiento, ocurre, porque es irrazonable, asumir que la prueba del software, podría descubrir todos los errores latentes. En sistemas de software grandes, durante el uso de algunos programas grandes, los errores pueden ocurrir y ser reportados, al desarrollador. El proceso que incluye diagnóstico y corrección de uno o más errores es llamado "mantenimiento correctivo".
- La segunda actividad, que contribuye a definir el mantenimiento, ocurre por los cambios rápidos, encontrados en cada aspecto de la computación. Generaciones nuevas de hardware, parecen ser anunciadas en un ciclo de 36 meses; los sistemas de operaciones nuevos y equipo periférico y otros elementos del sistema, son actualizados frecuentemente o modificados.
- La tercera actividad, puede ser aplicada para la definición del mantenimiento, ocurre, cuando un paquete

de software es exitoso. Este éxito, depende de como se usa el software, de las recomendaciones para nuevas capacidades, modificaciones para funciones existentes y realces en general, que son recibidas de los usuarios.

Esta actividad cuenta para la mayoría de todo el esfuerzo consumido en el mantenimiento del software.

- La cuarta actividad del mantenimiento, ocurre, cuando el software es cambiado para mejorar mantenimientos futuros o rehabilitar o proveer mejores bases para realces futuros. A menudo, es llamado "mantenimiento preventivo", esta actividad es todavía relativamente rara en el mundo del software.

1.2.- SISTEMAS DE INFORMACIÓN.

El notable crecimiento experimentado, por las instituciones sociales, tanto en su tamaño como en su alcance, exige el desarrollo de sistemas de información amplios y complejos. Se requiere información, que refleje, no sólo las actividades internas de la organización, sino, también la actitud de la competencia, los intereses ambientales y sociales y las tendencias políticas y económicas. Esta información, debe indicar no únicamente lo que ha ocurrido o está ocurriendo, sino, lo que es más importante, lo que va a ocurrir. Es preciso conocer todo aquello que pueda contribuir a la planeación, operación y controles efectivos de las actividades de la organización.

La información se requiere por igual, en la industria privada, en las dependencias gubernamentales, en el ejército, en las instituciones educativas, en los hospitales y en muchas otras organizaciones.

Para llenar las necesidades de información, de una empresa compleja, es preciso diseñar un sistema mediante el cual, se pueda recopilar toda clase de datos y procesarlos de tal manera, que pueda obtenerse la información requerida para tomar decisiones a todos los niveles.

Para diseñar y poner en operación, los sofisticados sistemas de información, que exige nuestro complejo mundo actual, primero es preciso entender debidamente los conceptos básicos de datos é información.

..2.1.- DEFINICIÓN DE LOS DATOS, INFORMACION Y SISTEMAS DE INFORMACION.

a) DATOS.

Los datos son hechos aislados y en bruto, los cuales, situados en un contexto significativo, mediante una o varias operaciones de procesamiento, permiten obtener reducciones relacionadas con la evaluación e identificación de personas, eventos y objetos.

La finalidad básica al recopilar y procesar los datos es producir información.

b) INFORMACIÓN.

La información se distingue substancialmente de los datos, en que, estos últimos son mensajes en bruto y no evaluados, mientras, que la primera significa un aumento de conocimientos, obtenido por el receptor, mediante la coordinación apropiada, de los elementos de los datos con las variables de un problema. La información, es la adición u el procesamiento de los datos, que puede proporcionar un conocimiento o bien el entendimiento de ciertos factores.

c) SISTEMAS DE INFORMACIÓN.

Todas las organizaciones cuentan con alguna clase de sistema de información, aunque, no se trate mas, que, de un archivador y de un pequeño número de cuentas en el libro mayor. Sin embargo, para contar con un sistema funcional de información que satisfaga diversas necesidades, todos los datos medibles, deben ser organizados de manera, que, sea fácil registrarlos, almacenarlos, procesarlos, recuperarlos y comunicarlos, según, lo requieran los usuarios. El concepto de sistema de información se puede definir como:

“ Un conjunto sistemático y formal de componentes, capaz de realizar operaciones de procesamiento de datos, con los siguientes propósitos:

- Llenar las necesidades de procesamiento de datos correspondientes a los aspectos legales y otros, de las transacciones dentro de la organización.
- Proporcionar la información a los administradores, en apoyo de las actividades de planeación, control y toma de decisiones.
- Producir una gran variedad de informes, según se requiera, para los grupos externos."

1.3.- REINGENIERIA.

La implementación de reingeniería de procesos de negocios, requiere una claridad de los objetivos de la organización y del conocimiento y manejo del comportamiento organizacional.

La reingeniería, promueve el cambio y el desarrollo, lleva a cambios en su cultura. Todos los cambios por naturaleza humana, tienden a generar conflictos, así, surge la necesidad, que para implementarla sean guiados por un líder, con el suficiente poder de conocimiento y persuasión, para guiar a sus equipos de trabajo, que sea un agente motivador, con gran propensión a la innovación y el cambio.

Existen diferentes definiciones de reingeniería, de tal forma que se puede decir que Reingeniería es:

- La revisión fundamental y el rediseño radical de los procesos de negocios, para alcanzar mejoras espectaculares, en medida críticas y contemporáneas de rendimiento. Mejoras en lo que respecta a costos, calidad, servicio y rapidez¹.
- Es el método mediante el cual, una organización puede lograr un cambio radical de rendimiento, medido por el costo, tiempo de ciclo, servicio y calidad, mediante la aplicación de varias herramientas y técnicas, enfocadas en el negocio como una serie de procesos del producto principal del negocio, orientados hacia el cliente, en lugar de una serie de funciones organizacionales².
- Es una revolución, que demuestra, que las personas pueden crear cambios organizacionales radicales, desde adentro y pueden hacerlo mientras obtienen éxitos financieros³

Por lo cual, el concepto que se percibe es:

“ La reingeniería, es el rediseño de los procesos de los negocios, mediante la aplicación de técnicas y

Hammer Michael y James Champy, Reingeniería. Editorial Norma. Colombia, 1994
pág. 2

Jobanson Henry J. Reingeniería de Procesos de negocios. Editorial Limusa 1994.
pág. 31

Maestría en Administración y dirección de Empresas. Made UCA.

herramientas, con el objetivo de alcanzar mejoras en costos, calidad, servicio y rapidez, para brindar una mayor satisfacción al cliente y obtener ventajas competitivas.

Los objetivos que se deben cumplir cuando se realiza un proceso de reingeniería son:

- Buscar instituir equipos de procesos, que reúnan a los empleados relevantes, contando a través de las jerarquías funcionales.
- Colocar los intereses del cliente en el centro de la organización.
- Rediseñar los procesos desde el punto de vista del cliente, trabajando hacia atrás, para encontrar los procesos mas eficientes de manejar los requerimientos exigidos por él.

1.4.- PROGRAMACION DE PROYECTOS CON PERT- CPM

Un proyecto, define, una combinación de actividades interrelacionadas, que deben ejecutarse en un cierto orden, antes que el trabajo completo pueda terminarse. Las actividades están interrelacionadas en una secuencia lógica, en el sentido, que algunas de ellas, no pueden comenzar, hasta que otras se hayan terminado. Una **actividad** en un proyecto, es un trabajo que requiere tiempo y recursos para

su terminación. En general, un proyecto, es un esfuerzo de un solo período; esto es, la misma sucesión de actividades puede repetirse en el futuro.

En el pasado, la programación de un proyecto (en el tiempo), se hizo con poca planeación. La mejor herramienta conocida de "planeación", era el diagrama de Gantt, el cual especifica los tiempos de inicio y terminación de cada actividad, en una escala de tiempo horizontal. Su desventaja, es que, la interdependencia entre las diferentes actividades (la cual controla principalmente el progreso del proyecto) no puede determinarse a partir del diagrama de barras.

Las complejidades crecientes, de los proyectos actuales, han exigido técnicas de planeación más sistemáticas y más efectivas, con el objeto de optimizar, la eficiencia en la ejecución del proyecto.

Aquí, la eficiencia implica, efectuar la mayor reducción, en el tiempo requerido, para terminar el proyecto, mientras se toma en cuenta la factibilidad económica de la utilización de los recursos disponibles.

La administración de proyectos, ha evolucionado como un nuevo campo, con el desarrollo de dos técnicas analíticas para la planeación, programación y control de proyectos. Tales son, el **Método de Ruta Crítica (CPM)** y la **Técnica de Evaluación y Revisión de Proyectos (PERT)**. Las dos técnicas fueron desarrolladas, por dos grupos diferentes casi

simultáneamente (1956-1958). El CPM fue desarrollado primero, por, E.I. du Pont de Nemours & Company, como una aplicación a los proyectos de construcción y, posteriormente, se extendió a un estado más avanzado, por Mauchy Associates. El PERT (Project Evaluation and Review Technique), por otra parte, fue desarrollado para la Marina de Estados Unidos, por una organización consultora, con el fin de programar las actividades de investigación y desarrollo, para el programa de misiles Polaris.

Los métodos PERT Y CPM están básicamente orientados en el tiempo, en el sentido, que ambos llevan a la determinación de un programa de tiempo. Aunque, los dos métodos fueron desarrollados casi independientemente, ambos son asombrosamente similares. Quizá, la diferencia más importante, es que, originalmente las estimaciones en el tiempo para las actividades, se supusieron determinantes en CPM y probables en PERT. Ahora PERT y CPM comprenden realmente una técnica y las diferencias, si existe alguna, son únicamente históricas. En adelante, ambas se denominarán técnicas de "programación de proyectos".

La programación de proyectos por PERT-CPM consiste en 3 fases básicas : **planeación, programación y control.**

- La fase de planeación, se inicia, descomponiendo el proyecto en actividades distintas. Las estimaciones de tiempo, para estas actividades, se determinan luego, y se

simultáneamente (1956-1958). El CPM fue desarrollado primero, por, E.I. du Pont de Nemours & Company, como una aplicación a los proyectos de construcción y, posteriormente, se extendió a un estado más avanzado, por Mauchy Associates. El PERT (Project Evaluation and Review Technique), por otra parte, fue desarrollado para la Marina de Estados Unidos, por una organización consultora, con el fin de programar las actividades de investigación y desarrollo, para el programa de misiles Polaris.

Los métodos PERT Y CPM están básicamente orientados en el tiempo, en el sentido, que ambos llevan a la determinación de un programa de tiempo. Aunque, los dos métodos fueron desarrollados casi independientemente, ambos son asombrosamente similares. Quizá, la diferencia más importante, es que, originalmente las estimaciones en el tiempo para las actividades, se supusieron determinantes en CPM y probables en PERT. Ahora PERT y CPM comprenden realmente una técnica y las diferencias, si existe alguna, son únicamente históricas. En adelante, ambas se denominarán técnicas de "programación de proyectos".

La programación de proyectos por PERT-CPM consiste en 3 fases básicas : **planeación, programación y control.**

- La fase de planeación, se inicia, descomponiendo el proyecto en actividades distintas. Las estimaciones de tiempo, para estas actividades, se determinan luego, y se

construye un diagrama de red (o de flechas), donde cada uno de sus arcos (flechas), representa una actividad. El diagrama de flechas completo, da una representación gráfica de las interdependencias, entre las actividades del proyecto. La construcción del diagrama de flechas como una fase de planeación, tiene la ventaja, de estudiar los diferentes trabajos en detalle, sugiriendo, quizá, mejoras antes de que el proyecto realmente se ejecute.

- El objetivo de la fase de programación, es construir un diagrama de tiempo, que muestre los tiempos de iniciación y terminación para cada actividad, así, como, su relación con otras actividades críticas (en función) que requieren atención especial, si el proyecto se debe terminar oportunamente. Para las actividades no criticas, el programa debe mostrar los tiempos de holgura, que pueden utilizarse, cuando tales actividades se demoran, o cuando se deben usar eficientemente recursos limitados.
- La fase final en la administración de proyectos, es la de control. Esto incluye, el uso del diagrama de flechas y la gráfica de tiempo para hacer reportes periódicos del progreso. La red puede, por consiguiente, actualizarse y analizarse y si es necesario determinar un nuevo programa para la parte restante del proyecto.

1.5.- CASO DE ESTUDIO.

1.5.1.- PROCESO DE INSCRIPCIÓN A "SCIENTIFIC AMERICAN"

En el año de 1960, la revista "SCIENTIFIC AMERICAN" fue fundada con una fuerte expansión de números de subscriptores, la que produjo una sobre carga de trabajo en el proceso de subscripción de datos, y un lento, incomodo y poco confiable sistema manual, para el cumplimiento de la subscripción.

El sistema utilizado se resume así:

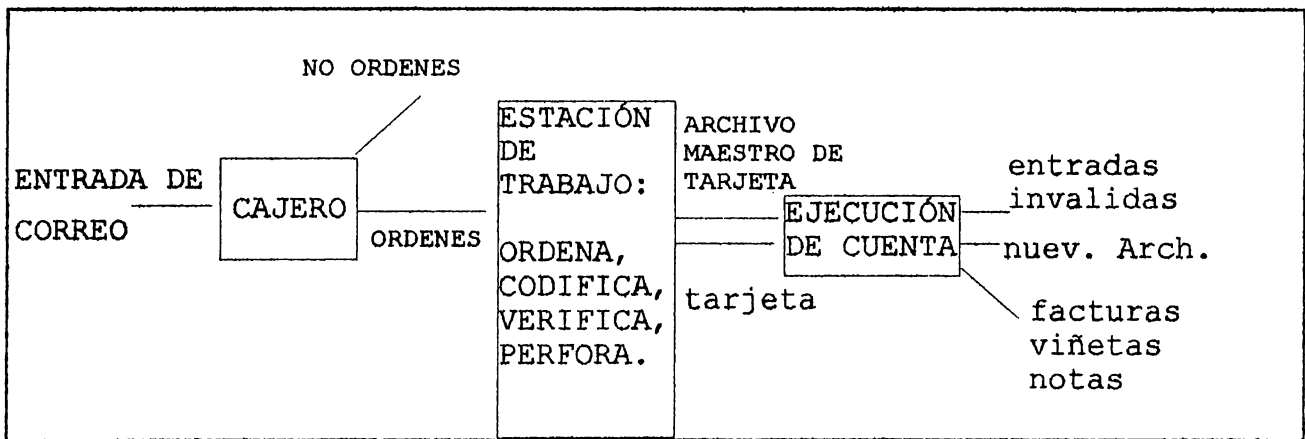


Figura 1.1

a) DESCRIPCIÓN DEL SISTEMA.

El Primer paso, es la entrada del correo a los cajeros, en donde se paga en efectivo, cheques o money orders, en esta parte son ordenados para el siguiente paso. El resto de ítems, son divididos en órdenes y no

órdenes (quejas, cambios de direcciones, etc.). en esta etapa, los ítems, son manualmente contados y totalizados para servir como un control para los siguientes pasos.

Los pasos restantes son realizados por operadores con una secuencia de estaciones de trabajo, en el "Departamento de cumplimiento de las estaciones". Estas personas ordenan las operaciones por tipos de transacciones (nuevas, continuas, regalos), codificados por entradas, y pasados para el proceso de perforación de tarjetas. Después, comienza la perforación, el orden en que están procesados por equipo de unidad - registro (tarjetas, tabuladores, impresiones y ordenamientos) para producir un conjunto inicial de cuentas y viñetas de correo.

La tarjeta suscrita, será entonces manualmente insertada en un archivo maestro perforado, el cual, podría ser periódicamente procesado, para, producir viñetas de correo mensuales, notificaciones de renovación anual, etc.

Esta labor intensiva del sistema, se vuelve incrementalmente lenta, cara, poco confiable e incapaz de mantenerse los tiempos máximos de respuesta en el proceso de suscripción. El retraso resultante, causa un incremento en el número de quejas de los clientes, motivo por el cual, los Administradores de "SCIENTIFIC

AMERICAN", deciden automatizar el proceso de suscripción.

b) LA SOLUCIÓN DEL PROGRAMA.

"SCIENTIFIC AMERICAN", contrató una empresa para el desarrollo de Software, y así poder desarrollar, un sistema computarizado, que solviente sus necesidades. Las empresas de desarrollo de software, usualmente enfocan los trabajos desde la perspectiva de programación. Sus conocimientos previos y experiencia, son centralizados, sobre las soluciones derivadas de programación desde la clara formulación de los problemas.

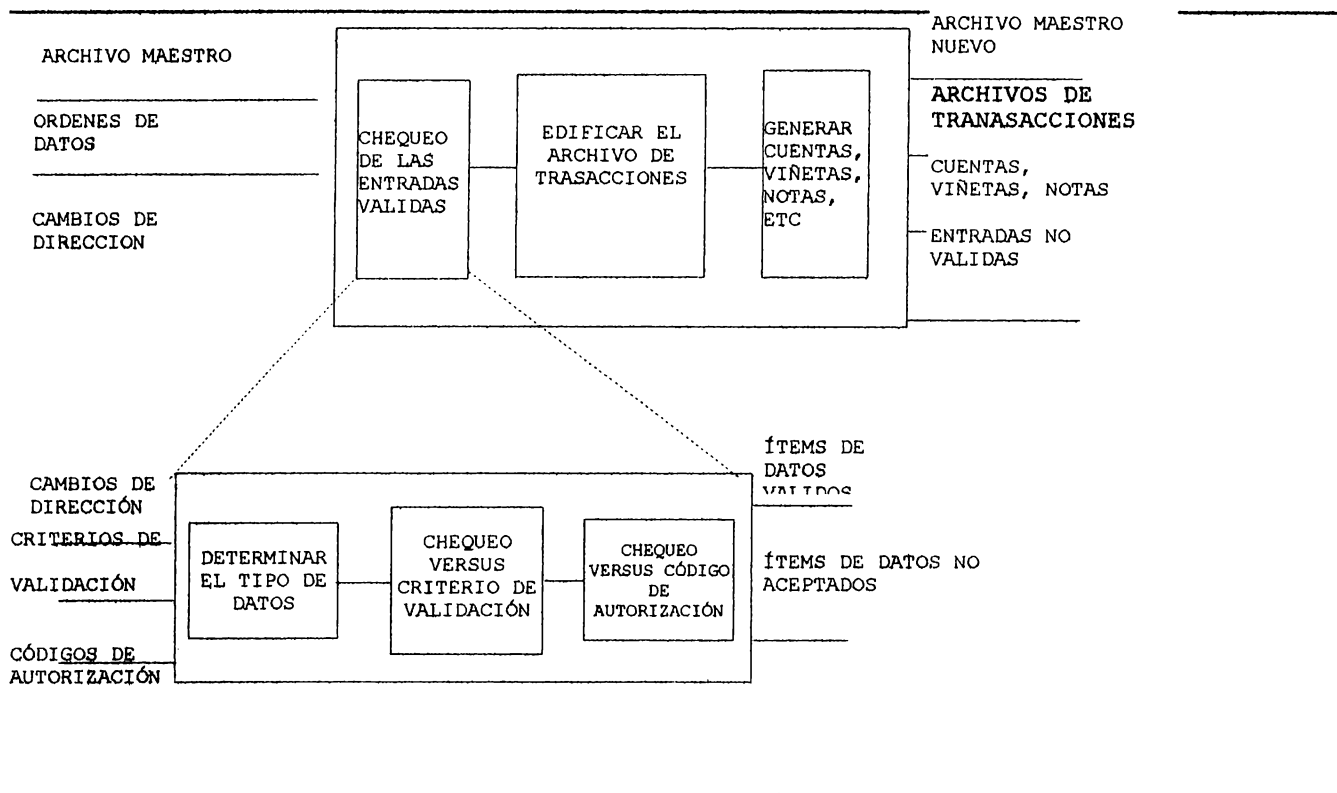


Figura 1.2

Así, en la situación actual de la empresa, los programadores analizaron la situación, identificaron la parte que tenía la solución de programación y aplicaron dicha solución.

De tal forma, que, la parte derecha de la primera figura (Procesamiento y actualización del archivo maestro) se mira como, uno de los puntos de comienzo en la situación actual sobre la programación estructurada TOP-DOWN, de refinamiento Step-Wise. Como se ve en la figura 1.2, el paso siguiente en el refinamiento StepWise, es refinar la función de alto nivel (Proceso de datos) en una secuencia de funciones de bajo-nivel (chequeo de validación de datos, etc.), y para continuar refinando funciones, hacia niveles mas bajos (Determinar el tipo de datos, etc.).

El desarrollo de la solución de programación, por la empresa contratada, es mostrada en la figura 1.3. Reemplazando, el paso de ejecución de cuenta, con un procesador programado IBM 360/30 a lo largo de las líneas de la figura siguiente:

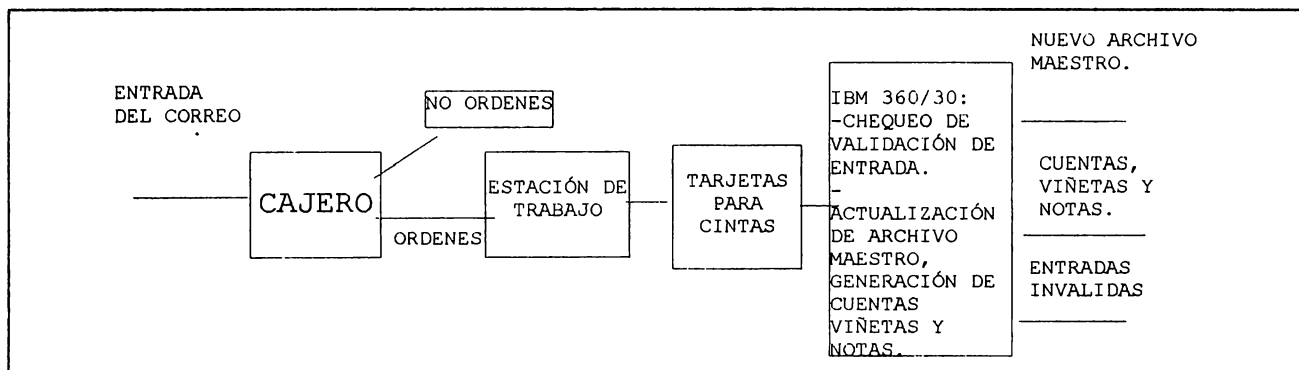


Figura 1.3

☑ LOS RESULTADOS DE LA SOLUCIÓN DE PROGRAMACIÓN.

Los resultados principales de esta solución, fueron los siguientes:

- Los costos se incrementaron.
- La puntualidad y calidad del servicio bajo.
- Mas personal fue requerido.
- La moral de los empleados bajó.
- EL cambio de personal aumento.

La razón principal, para estos resultados, fue que, esta solución omitió algunos elementos operacionales que eran claves en la solución del problema. Los factores omitidos, contenían procedimientos significativos e implicaciones económicas. Por ejemplo, los errores insignificantes de entrada, podrían ocasionar, que el programa rechace la introducción de procesos de entrada, y detener completamente la ejecución de actualización. Cuando esto sucede, los listados de transacciones de entradas, tienen que ser buscados y manualmente revisados de errores. La corrección de estos errores, complica estos procesos incómodos de actualización de archivos de transacción. En el antiguo sistema de ejecución de cuentas, eran usualmente descubiertos y arreglados en el momento por los operadores, quienes tuvieron caras experiencias por esperar y arreglarlos.

Para tratar de evitar, los posibles errores de tiempos consumidos por esperas, la empresa, impuso niveles adicionales de controles sobre los pasos de procesamientos manuales, guiados por la ejecución de computadoras. Adicionalmente, se realizaron chequeos para evitar problemas, como por ejemplo, la renovación de suscripciones inexistentes. Se desarrollaron hojas de control, para manejar condiciones de excepción, las cuales se tornaron mas complicadas. Todo esto produjo la adición de personal y reducción de la moral del staff.

c) LA PERSPECTIVA DE LA SOLUCIÓN DE PROGRAMACIÓN ECONÓMICA.

Afortunadamente, a este punto, la empresa empleo una persona, la cual es capaz de acceder el problema desde una perspectiva de programación - económica. Esta persona analizó todo el sistema, y le dio una orientación hacia el usuario con un acceso TOP-DOWN.

Las consideraciones involucradas siguen la siguiente secuencia de preguntas:

- ¿ Qué objetivos intenta satisfacer el usuario ?
- ¿ Qué decisiones de control, se deben hacer que afectan estos objetivos?
- ¿Qué ítems afectan nuestro rango de selección?

los criterios, relacionados a las variables de solución involucradas en las soluciones candidatas?

- ¿ Qué decisiones, proporcionan los resultados mas satisfactorios con respecto al criterio que se tiene establecido ?

Basado en las respuestas de las preguntas, la persona a desarrollar la solución, de programación - económica, fue capaz de definir y desarrollar una solución para las necesidades de la empresa de costo - efectivo. A continuación esta un resumen de las respuestas que el determinó:

- ¿ Qué objetivos intenta satisfacer el usuario ?

Para la empresa, el objetivo principal, es incrementar la velocidad y puntualidad en el cumplimiento de la subscripción, y para reducir los costos, el nivel de staff, cambios y las quejas del cliente. Respondiendo a estas preguntas, también involucró al staff y analizó los datos sobre fuentes de costos primarios, errores, demoras y frustraciones.

- ¿ Qué decisiones de control se deben hacer que afectan estos objetivos?

Para la empresa, esta decisión incluye, no solo una decisión de computadora, sino, también, la decisión para usar cajas postales separadas para diferentes tipos de

ordenes, transferencias ordenadas de trabajo, ordenamientos para los servicios postales, etc.

- ¿Qué criterios se podrían usar, para evaluar las soluciones candidatas ? ¿ Cómo son los valores de los criterios, relacionados a las variables de solución involucradas en las soluciones candidatas?

Criterios importantes para la empresa, incluyeron el costo del procesamiento de suscripción, el tiempo requerido para responder a los cambios de suscriptores, rangos de errores y niveles de personal de staff y rangos de cambios de personal. Con esta respuesta se encuentra, que el procesamiento manual fue el mayor lugar del problema, y estas cargas de trabajo de procesamiento, fue insuficiente para justificar un IBM 360/30 dedicado.

- ¿ Qué decisiones proporciona los resultados mas satisfactorios con respecto al criterio que se tiene establecido ?

La decisión hecha por la empresa, fue, para implementar el sistema resumido en la figura 1.4. Números de cajas de oficinas postales, se separaron para ser usadas en diferentes entradas de ordenes, así, cumplieron un ordenamiento básico de las entradas de la oficina postal. El primer caso ejecutado por la empresa, fue la apertura de sobres y entradas de ordenes por un empleado, usando una minicomputadora base dedicado, como terminal inteligente. Esta terminal, fue programada,

para permitir validaciones interactivas de entradas; las entradas serán chequeadas por la minicomputadora, dentro de la terminal e inmediatamente mandada para correcciones por el empleado en la orden de entrada. Las entradas fueron optimizadas, así, las transacciones mas frecuentes (Orden standard de renovación) requerían solo 21 llaves de golpe. Los resultados de las entradas, son registrados en una cinta de cassette, los que son tomados diariamente por un servicio de oficina para su procesamiento.

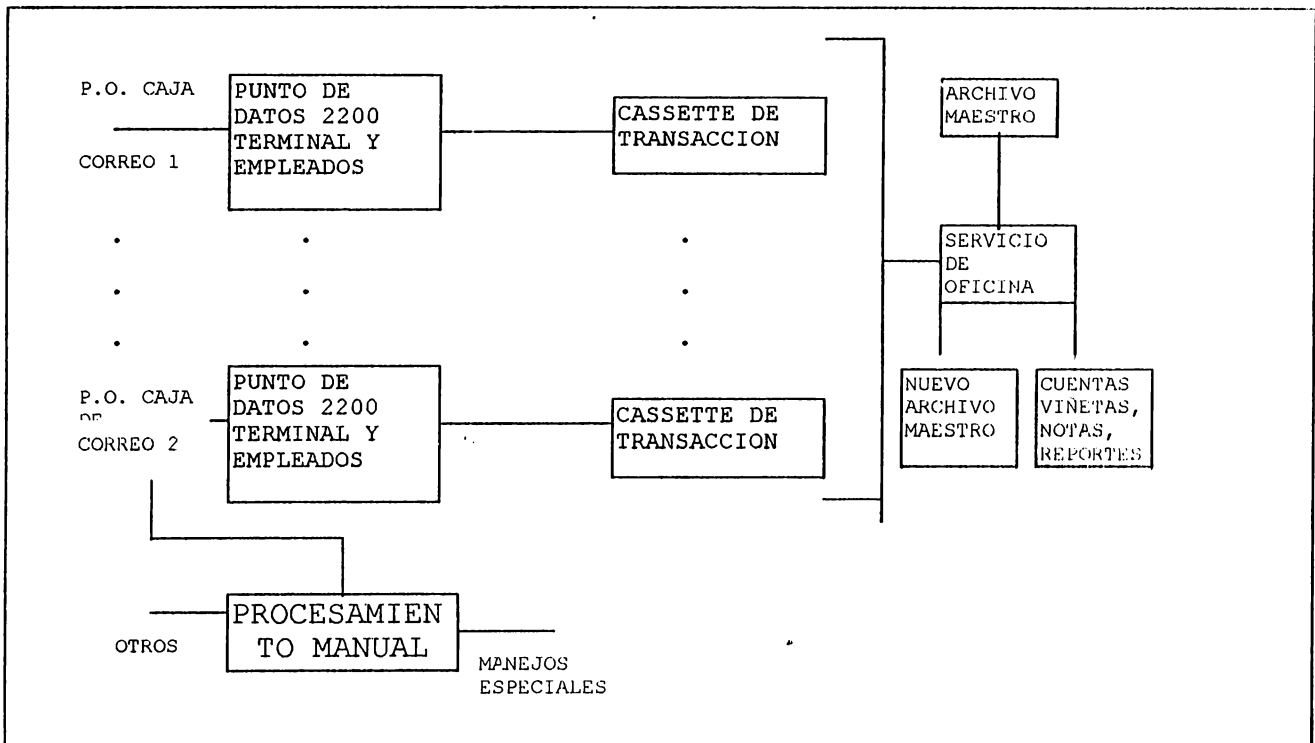


Figura 1.4

EL RESULTADO DEL ACCESO DE LA PROGRAMACION ECONOMICA.

La secuencia de preguntas por superficiales, constituye la esencia del acceso de análisis, orientado a sistemas económicos. El análisis llevo a:

- Instalar 5 terminales de minicomputadoras inteligentes, de \$10,000, para eliminar el IBM 360/30 recuperando aproximadamente \$9,000 por mes.
- Un servicio de oficina, para el procesamiento de datos, el que fue contratado por aproximadamente \$7,000 al mes.
- El sistema resultante, requiere una reducción significativa en el personal, así, la empresa fue capaz de amortizar su inversión en terminales, para el final del primer año de operación.

Al final del año, el resultado fue el siguiente:

- El número de trabajadores fue reducido de 40 a menos de 20.
- Los errores de empleados, fueron reducidos, a una pequeña fracción de el número de inevitables "errores al subscribir" (por ejemplo, duplicar ordenes o pagos).
- Todas las transacciones virtuales, fueron manejadas en menos de una semana, comparado con los frecuentes retrasos de mas de un mes, en los previos sistemas.
- El nuevo sistema, manipula un 33% de incremento de la carga de trabajo.
- Existieron respuestas subjetivas de los empleados que indicaron un incremento en el sentimiento de la satisfacción laboral. La orden en la entrada de paso individual, les dio mas significado al trabajo realizado, con mas oportunidad para

ejecutar sus propios juicios, en el manejo de problemas y excepciones.

Así, el acceso a la programación - económica, guió a una solución satisfactoria o sobrepasó los objetivos mejorados de la empresa, donde la solución pura de programación, solamente sucedió en hacer malas soluciones para "Scientific American".

En general, el valor principal del acceso de programación para "Scientific American", fue, que permitió el análisis para ver el problema desde mas de una perspectiva, llegando a la siguiente consideración:

"Den a un hombre un martillo, y comenzará a ver al mundo como un grupo de clavos".

Esto es lo que pasa con la primera solución para Scientific American.

El programador - analista, identifica la parte del problema que fue de programación y comenzó a dar, una solución de programación, sin considerar otros problemas que no involucran la programación.

CAPITULO II.

2.1.- MODELO BASICO DE COCOMO.

2.1.1.- DEFINICIONES.

El modelo básico COCOMO, es bueno para una estimación de costos preliminar del proyecto, pero esta posición, esta limitada porque existen factores que afectan de forma significativa al proyecto (hardware obligado, personal experimentado y calificado, etc.).

Los parámetros que este método utiliza son:

- El principal parámetro utilizado, es el número de instrucciones fuentes entregadas (DSI : Delivered Source Instruction) para el proyecto. La palabra *DELIVERADO*, es un término generalmente interpretado, para excluir software de soportes de prueba no deliverados, los cuales son manejadores de prueba. Sin embargo, si dichos manejadores son desarrollados con algunos cuidados, como, el hecho, de poseer sus propias revisiones, planes de prueba, documentación, etc. entonces pueden ser considerados como software deliverados.
- Las *INSTRUCCIONES FUENTE*, este término, incluye todas las instrucciones creadas por el personal del proyecto y procesadas en un código máquina por alguna combinación de pre-procesadores, compiladores y ensambladores. Esto excluye tarjetas de observaciones entre otras. Estas

instrucciones incluyen lenguajes de control de trabajo, formato de instrucciones de declaraciones de datos. Las instrucciones están definidas como líneas de código. Así, una línea contiene 2 o mas declaraciones fuentes contadas como una instrucción, unas cinco líneas de declaración de datos cuentan como cinco instrucciones.

- El período de desarrollo, cubierto por la estimación de costos de COCOMO, comienza con la fase de diseño del producto y finaliza con la fase de integración y prueba. La estimación de costos de COCOMO abarca las actividades indicada del software en la estructura de análisis de trabajo (WBS). Así, por ejemplo, la estimación de desarrollo, cubre el esfuerzo de administración y documentación, pero excluye algunos esfuerzos, los cuales toman lugar, durante el período de desarrollo, tal como, la capacitación del usuario, planeación de instalación y planes de conversión.
- La Estimación de COCOMO, cubre las labores de cargos directos sobre el proyecto. Así, se incluye el manejo de proyectos y librerías de programas, pero, excluye el centro de computo, el departamento de personal, secretarias, alta gerencia, consejería, etc.
- Un hombre-mes, consiste de 152 horas laborales. Este dato, fue encontrado para ser consistente con la experiencia en la practica, con el tiempo del mes promedio libre de días festivos, vacaciones y enfermedades. Para convertir este valor estimado de

hombre-mes, a otras unidades, se debe usar la siguiente conversión:

Hombre-hora: *152.

Hombre-día : *19.

hombre-año : *12.

- La estimación de COCOMO, asume, que el proyecto será bien administrado por el desarrollador y el cliente. Por ejemplo, la suma de improducciones en tiempo lento, esta mantenido por el administrados y el cliente.
- COCOMO asume, que la especificación de requerimientos, no sufre cambios substanciales, después de las fases de planeación y requerimientos. Algunos refinamientos y re-interpretaciones serán inevitables, pero, cualquier modificación significativa o adiciones, podrían causar una revisión de la estimación del costo.
- El modelo detallado de COCOMO, asume, que la influencia de el manejador de costos del software es fase dependiente de otras. El modelo básico y el intermedio no lo hacen, excepto para la diserción entre el desarrollo y el mantenimiento.
- El costo de cada fase, incluye todos los costos incurridos durante la misma, de actualización de la integración, planes de prueba, así como la aceptación de dichos planes son incluidos, en el detalle del costo de la fase de diseño.

Este proceso, enfatiza los siguientes pasos:

- Una definición y validación cuidadosa de la especificación de requerimientos, por un número relativamente pequeño de personas, para priorizar el trabajo sobre todo el diseño del sistema.
- Una definición y validación cuidadosa del diseño del sistema.
- Detallar el diseño, codificación y unidades de procedimientos de pruebas, para un gran grupo de programadores en paralelo, trabajando en un diseño prototipo; muchas veces con respecto al desarrollo incremental planificado.
- La integración y prueba de cada incremento, basado sobre una medida significativa de una planeación de prueba temprana y la eliminación de casi todas las unidades internas.

2.1.2.- MES HOMBRE VERSUS ESTIMACION EN EFECTIVO.

El método de COCOMO, no trata de estimar los costos de las labores en efectivo, debido a las grandes variaciones entre organizaciones (Planes de pensión, rentas de oficinas, márgenes de ganancias, etc.) y además, porque, el valor cuantitativo de MES-HOMBRE, es mas estable en relación con el rango de inflación y la fluctuación de la moneda.

2.1.3.- DESARROLLO DE ESFUERZO Y HORARIO.

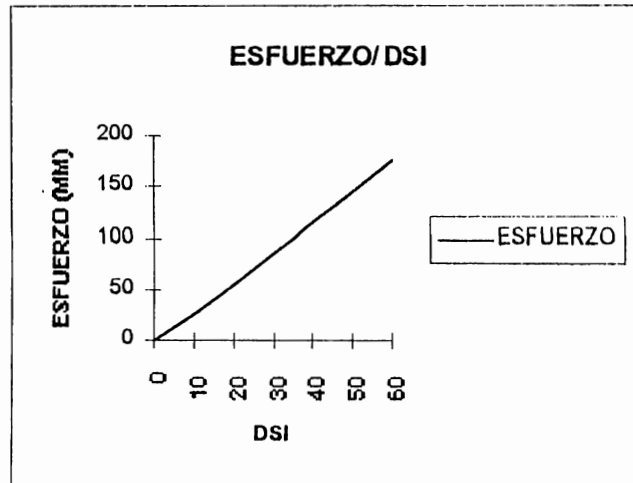
Para poder determinar una adecuada distribución de esfuerzo y horario, se tiene que establecer, lo suficiente de el ciclo de vida del software, para darse una idea de que se esta estimando.

2.1.4.- LAS ECUACIONES QUE COCOMO UTILIZA.

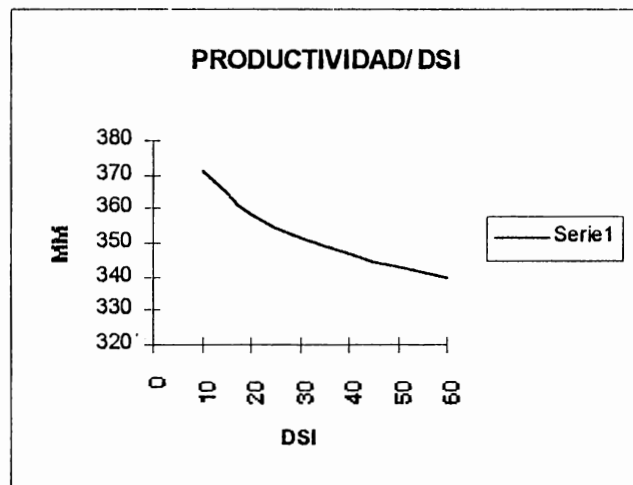
La ecuación del esfuerzo básico, para un proyecto de modo orgánico es:

$$MM = 2.4 (KDSI)^{1.05} \quad (\text{Ecuación : 2.1})$$

El valor cuantitativo KDSI, es el número de miles de instrucciones fuentes deliberadas en el producto de software. Este valor MM, es el número de meses hombres estimado para el desarrollo de cada fase del software en el ciclo de vida. Un gráfico de la estimación de productividad y esfuerzo con unidades DSI/MM, esta presentado a continuación:



GRAFICA 2.1



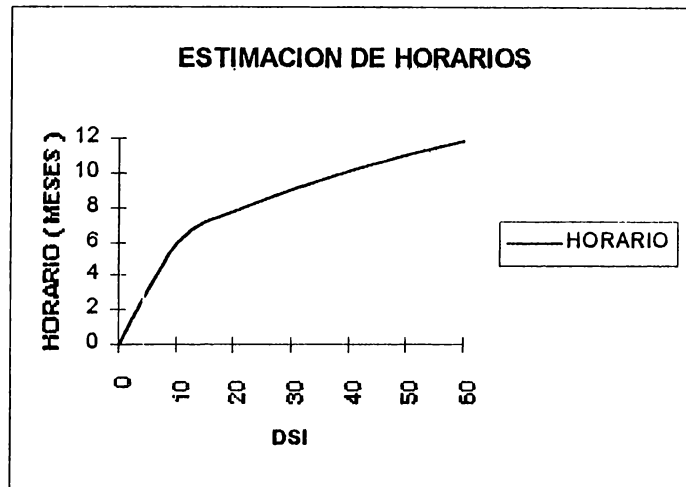
GRAFICA #2.2

La ecuación básica de horarios para un proyecto de software de modo-orgánico es:

$$TDEV = 2.5 (MM)^{0.38} \text{ (Formula 2.2)}$$

El valor cuantitativo TDEV, es el número de meses estimado para el desarrollo del software, usando las mismas definiciones y asunciones como para la ecuación

básica de esfuerzo. Un gráfico de la estimación dado por la ecuación básica de horario, se presenta a continuación:



GRAFICA 2.3

2.2.- EJEMPLO DE ESTIMACION.

Du Bridge Chemical Inc., una compañía de productos químicos, esta planeando desarrollar un nuevo programa de computadora para llevar el control de la materia prima. Este programa será desarrollado por un equipo de programadores y analistas de la empresa. Un estudio inicial ha determinado, que el tamaño del proyecto será de aproximadamente 32,000 líneas de instrucciones deliberadas (32KDSI). Desde la ecuación básica, se estimaron las siguientes características del proyecto:

ESFUERZO: $MM = 2.4(32)^{1.05} = 91$ HOMBRES-MES.
 PRODUCTIVIDAD: $(32,000DSI) / (91MM) = 352$ DSI/MM.

HORARIO: $TDEV = 2.5(91)^{0.38} = 14$ MESES.

STAFFING PROMEDIO: 91 HOMBRES-MES/14MESES = 6.5FSP.

El valor cuantitativo FSP, colocado para el tiempo-completo del personal del software, es una medida del número equivalente de personas trabajando en el proyecto a un tiempo determinado.

☑ PERFILES DEL PROYECTO.

La tabla 2.1 muestra los perfiles similares para un conjunto de tamaños standard de modos orgánicos de proyectos:

PEQUEÑOS: 2,000DSI.

INTERMEDIOS: 8,000DSI.

MEDIO: 32,000DSI.

GRANDE: 128,000DSI.

TAMAÑO DEL PRODUCTO		ESFUERZO	PRODUCTIVIDAD AD	HORARIOS MESES	STAFFING PROMEDIO
PEQUEÑO.	2KDSI	5.0 MM	400 DSI/MM	4.6	1.1 FSP
INTERMEDIO	8KDSI	21.3 MM	376 DSI/MM	8.0	2.7 FSP
MEDIO	32KDSI	91.0 MM	352 DSI/MM	14.0	6.5 FSP
GRANDE	128KDSI	392.0 MM	327 DSI/MM	24.0	16.0 FSP

TABLA 2.1

Se puede ver, que un proyecto pequeño es esencialmente el trabajo de una persona, mientras que un proyecto grande requiere un promedio de 16 personas.

☑ FASE DE DISTRIBUCION.

La tabla 2.2, presenta la distribución de porcentaje del esfuerzo básico del software, y horarios dentro de las fases del desarrollo del ciclo de un sistema. La fase de distribución, esta definida en función del tamaño del producto. Los proyectos grandes, requieren relativamente mas tiempo y esfuerzo para realizar las actividades de prueba e integración (y son hábiles para comprimir la parte de programación de componentes en paralelo). Los proyectos pequeños tienen una mayor uniformidad, planos de distribución de labores a través del desarrollo del ciclo, y tienen mas recursos dedicados para otras fases como la integración y prueba.

Como un ejemplo de el uso de esta tabla, considérese el tamaño medio (32KDSI) para la empresa Bridge Chemical Inc, en el proyecto de inventario de materia prima. En este proyecto se determinaran las ecuaciones de horario y esfuerzo como COCOMO Básico, el cual produce un requerimiento en el esfuerzo, superior de 91MM y mas de cuatro meses en su horario. Supongamos que necesitamos conocer un tiempo promedio, esfuerzo y un número total promedio del tiempo-completo que equivale a las persona, que trabajan en el proyecto (FSP), que se necesitan para la fase de programación.

En la tabla 2.3 vemos que la fase de programación requiere 62% del esfuerzo total o:

$$(0.62)(91MM) = 56MM.$$

La fase de programación requiere un 55% del horario total, o

$$(0.55)14 = 7.7 \text{ meses.}$$

El número promedio de personas (FSP) requerido para la fase de programación es:

$$56MM / 7.7 \text{ meses} = 7.3 \text{ FSP.}$$

FASE	TAMAÑO DEL PRODUCTO			
	PEQUEÑO (2KDSI)	INTERMEDIO (8KDSI)	MEDIO (32KDSI)	GRANDE (128KDSI)
<i>ESFUERZO.</i> PLANES Y REQUERIMIENTOS.	6%	6%	6%	6%
DISEÑO DEL PRODUCTO	16	16	16	16
PROGRAMACION	66	65	62	59
DISEÑO DETALLADO	26	25	24	23
CODIFICACION Y PRUEBA DE UNIDADES.	42	40	38	36
INTEGRACION Y PRUEBA	16	19	22	25
TOTAL	100%	100%	100%	100%
<i>HORARIOS.</i> PLANES Y REQUERIMIENTOS.	10%	11%	12%	13%
DISEÑO DEL PRODUCTO.	19	19	19	19
PROGRAMACION	63	59	55	51
INTEGRACION Y PRUEBA	18	22	26	30
TOTAL	100%	100%	100%	100%

Tabla 2.2

ESTIMACION DEL ESFUERZO EN EL MANTENIMIENTO DEL SOFTWARE.

La estimación en la fase de mantenimiento, es calculada en un término cuantitativo llamado "Número de Cambio Anual" (ACT), este valor, es la fracción de las

instrucciones fuentes del producto, las cuales son cambiadas durante un típico año, de ajustes o modificaciones.

Por ejemplo, en Du Bridge Chemical, el software necesito 4,000DSI adicionales, y 2,400DSI modificadas durante su primer año de mantenimiento. La fórmula para encontrar el ACT es:

$$ACT = (4000+2400)/32000 = 6400/32000 = 0.2.$$

La ecuación de COCOMO para estimar el esfuerzo anual de mantenimiento MM_{AM} es:

$$MM_{AM} = 1.0 (ACT) (MM)_D (FORMULA 2.3)$$

Utilizando un tiempo de trabajo de persona (FSP) contribuidas en 12 MM en un año, pueden ser fácilmente calculadas como:

$$FSP_M = (MM)_{AM} / 12 (Esfuerzo en la modificación).$$

Por ejemplo, considere el mantenimiento de las 32KDSI del producto diseñado con:

$$ACT = 0.2.$$

$$MM_{AM} = 0.2(91) = 18MM.$$

Y el esfuerzo:

$$FSP_M = 18/12 = 15.$$

Se conocen algunas técnicas básicas para la estimación de todo esfuerzo, costo y horario, que son distribuidos a través de varias fases del proyecto. En adición a esta información, a menudo se encuentra necesario, estimar, como el esfuerzo es distribuido entre las primeras actividades del ciclo de vida de un software: diseño, programación, prueba, planeación, etc.

Por ejemplo:

- En la preparación de planes de personal para el proyecto de software, se necesita conocer que talentos especiales el proyecto necesitara y a que tiempo.
- En la gerencia del proyecto, se necesita ser capaz de trabajar fuera de las asignaciones individuales y responsabilidades con los posibles miembros del equipo.
- En la organización del proyecto, se necesita llevarla al limite, particularmente en grandes proyectos, la estructura organizacional necesitará ser modificada varias veces durante el ciclo de vida de un software como el modelo de cambios de actividades del proyecto.

☑ DISTRIBUCION DE ACTIVIDADES POR FASES.

Las tablas 2.3⁴, 2.4 y 2.5, presentan, como el proyecto de software se esfuerza en cada fase del ciclo de vida, que esta distribuido entre las ocho actividades principales:

- Análisis de requerimientos.
- Diseño de producto.
- Programación.
- Prueba.

⁴ Las tablas 2.3, 2.4 y 2.5 están definidas en el Anexo de este documento

- Verificación y validación.
- Funciones de la oficina de proyectos.
- Configuración y calidad de seguridad.
- Manuales.

Estas diferentes tablas, muestran los datos para cada diferente Modo (Encadenamiento, semi-capturado y orgánico).

En la tabla 2.3 y 2.5, se puede ver, la distribución de esfuerzo para varias actividades con el tamaño del proyecto. El tamaño normal del proyecto se determina con los siguientes elementos :

- S : pequeño (2KDSI)
- I : intermedio (8KDSI)
- M : mediano (32KDSI)
- L : grande (128KDSI)
- VL : muy grande (512KDSI)

La distribución de actividades, para proyectos de otros tamaños, son obtenidos por interpolación lineal dentro de las tablas 2.3, 2.4 y 2.5.

Para explicar la tabla 2.3 , tiene que ubicarse primero, los números de la columna izquierda; esta columna indica, que, en un proyecto pequeño (2KDSI) , el 8% del total de esfuerzo puede ser requerido para completar el plan y la fase de requerimientos. De esta

parte (8%) , 50% (o 4% del proyecto total) podría ser dedicado a la ejecución de tareas en el análisis de requerimientos (Analizando el sistema existente, se determinan las necesidades del usuario, integración, documentación é iteración de requerimientos). Los otros 12% de la parte del 8% (o 1% del total del proyecto) puede ser dedicado para la ejecución de la tarea de diseño del producto. Durante el plan y fase de requerimientos: se investiga lo básico del hardware y software , ejecutando sensiblemente y analizando el riesgo para asegurar que los requerimientos sean factibles. Otro 2% de la parte del 8% podría ser dedicada para la tarea de programación. En la próxima columna de esta tabla, se puede ver que la distribución de actividad, es algo diferente para un proyecto intermedio (8KDSI) durante el plan y la fase de requerimiento. Para el proyecto intermedio, solo el 48% del esfuerzo en el plan y la fase de requerimientos, es dedicada al análisis de los mismos, mientras, que, el 13% es dedicado al diseño del producto. La razón de esta diferencia, es que en proyectos grandes, se tiende a necesitar más esfuerzo para investigar la factibilidad de los requerimientos, para la ejecución del diseño del producto que se hacen para los proyectos pequeños. Las otras partes de la misma tabla 2.3, tienen una interpretación similar.

Como un ejemplo, del uso de la tabla 2.3, se calculó la distribución de actividades para la fase de diseño del

producto, de un proyecto muy grande (512 KDSI) en el Modo de Encadenamiento. El desarrollo total de esfuerzo y horario para un proyecto muy grande de Modo de Encadenamiento es de 6420 MM y 41 meses, respectivamente.

El correspondiente esfuerzo y porcentaje de horario para la fase de diseño del producto es de 18% y 38% . Así el proyecto podría requerir:

$$(6420 \text{ MM}) (0.18) = 1156 \text{ MM}$$

$$(41 \text{ meses}) (0.38) = 15.6 \text{ meses, para}$$

completar la fase de diseño de producto. Esto significa, que un promedio de $1156/15.6 = 74$ personas (FSP) están en trabajo durante dicha fase.

De la tabla 2.3, se pueden inferir la duración y las actividades de las 74 personas.

Usando las sub-columnas de la derecha (v_1) de la columna de diseño del producto en la tabla 2.3, se ve que el 10% de las 74 personas (aproximadamente 7 personas), podrían ejecutar las funciones de análisis de requerimientos; 42% de las 74 personas, o 31 personas, podrían ejecutar las funciones de diseño del producto. El número de personas involucradas en cada una de las ocho actividades es dada a continuación :

- Análisis de requerimientos	=	7 FSP
- Diseño de producto	=	31 FSP
- Programación	=	10 FSP
- Planeación de la prueba	=	6 FSP

- Se espera que el proyecto sea un caso normal con respecto a los que manejan los costos (personal, ambiente, etc.).

Este último descubrimiento, significa que puede razonarse usando el modelo de COCOMO básico , determinando estimaciones sobre el ciclo de vida de un producto.

Primero se usará la ecuación del modo de encadenamiento, para determinar los requerimientos del desarrollo de productos en esfuerzo y horario, de tal forma que:

$$\begin{aligned} \text{Esfuerzo} &= 3.6 (80)^{1.8} && = 692 \text{ MM} \\ \text{Horario} &= 2.5 (692)^{0.32} && = 20 \text{ meses} \end{aligned}$$

Es de notar, que si un Mes-Hombre podría costar \$ 6000, el total de costos de las labores para desarrollar el sistema inicial podría ser \$ 4,152,000.

El producto de 30 KDSI, está exactamente a la mitad del camino entre el producto mediano normal (32 KDSI) y el producto grande normal (128 KDSI), así, se puede interpolar parcialmente entre estas columnas en las porciones del modo de encadenamiento. Esto significa, que el esfuerzo requerido para la planeación y fase de requerimientos del proyecto mediano-grande es 8% del total de esfuerzo de desarrollo, o (692) (0.08) = 55

MM, similarmente, se puede determinar el esfuerzo estimado, horario y nivel de FSP para cada fase.

CAPITULO III.

3.1.- MODELO INTERMEDIO DE COCOMO.

3.1.1.- INTRODUCCIÓN.

El modelo Básico de COCOMO, estima el esfuerzo requerido para desarrollar un producto de Software (Sistema), utilizando simplemente una variable (número de instrucciones fuentes) y tres modos de desarrollo de sistemas. Este modelo, explica bastante bien la variación del costo del proyecto. Sin embargo, este valor, solo es efectivo para hacer usado como una primera etapa de la definición del sistema.

En este capítulo, se presenta el Modelo Intermedio de COCOMO, el cual, es una extensión del anterior, en él, el valor del costo es mas detallado, en base a la definición del Sistema. Para esta parte del modelo, existen muchos factores candidatos a considerar, en el desarrollo de una mejor estimación del costo del proyecto. Estudios que se han llevado a cabo encuentran algunos factores, tales como:

- Tipo de aplicaciones.
- Tipos de entradas y salidas.
- Promedio de experiencia de los analistas y programadores.
- Configuración del hardware.
- Número de personas involucradas en las actividades del proyecto.
- Controles.
- Lenguajes de programación, etc.

El conjunto resultante de factores o atributos de manejadores de costo, usados en este Modelo, son agrupados en las siguientes categorías:

Atributos del producto.

- RELY: Integridad requerida para el Sistema.
- DATA: Tamaño de la base de datos.
- CPLX: Complejidad del Producto.

Atributos de la computadora.

- TIME: Tiempo crítico de ejecución.
- STOR: Almacenamiento crítico principal.
- VIRT: Inconstancia de la máquina virtual.
- TURN: Tiempo de retorno de la computadora.

Atributos del Personal.

- ACAP: Capacidad del Analista.
- AEXP: Experiencia en las aplicaciones.
- PCAP: Capacidad de programación.
- VEXP: Experiencia en la Máquina virtual.
- LEXP: Experiencia en el lenguaje de Programación.

Atributos del proyecto.

- MODP: Prácticas Modernas de programación.
- TOOL: Uso de herramientas en el producto.
- SCED: Horario requerido en el desarrollo.

Cada uno de estos atributos, son controlados por factores de multiplicación, que estiman el efecto de estos atributos sobre el esfuerzo en el desarrollo del sistema. Estos factores de multiplicación son aplicados para la estimación del esfuerzo en el desarrollo nominal de COCOMO, para obtener, una estimación refinada de el esfuerzo en el desarrollo del sistema. Un proceso similar es sacado para determinar una estimación refinada del esfuerzo, en el mantenimiento del sistema.

3.1.2.-DESARROLLO DE LA ESTIMACIÓN DEL ESFUERZO EN EL SISTEMA.

a) Ecuaciones de escala nominal.

La estimación de esfuerzo, en el desarrollo del sistema de el Modelo Intermedio COCOMO, comienza por la

generación de la estimación del esfuerzo nominal, usando ecuaciones fijas, como las usadas en el Modelo Básico de COCOMO. Esta estimación nominal, es entonces, ajustada por la aplicación de multiplicadores de esfuerzo, que se determinan en base al rango de proyectos, con respecto a los 15 atributos mencionados anteriormente.

El factor de escala del Modelo Intermedio COCOMO (exponentes), para los tres Modos de Desarrollo, son los mismos que en el Modelo Básico, diferenciándose de estos, en la variación de los coeficientes. La tabla 3.1 presenta la estimación de las ecuaciones del esfuerzo nominal, usadas para el Modelo Intermedio.

La gráfica 3.1 presenta una comparación del esfuerzo nominal en el Modelo Intermedio, estimado en los tres modos.

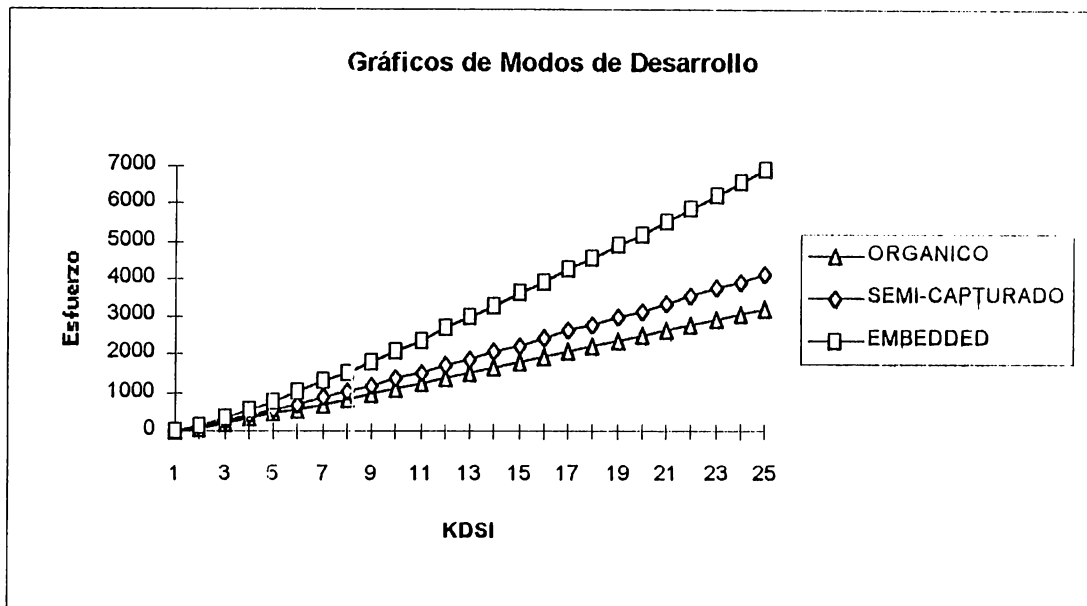
MODO DE DESARROLLO	ECUACIÓN NOMINAL DE ESFUERZO
Orgánico.	$(MM)_{NOM}=3.2 (KDSI)^{1.08}$
Semi-Capturado.	$(MM)_{NOM}=3.0 (KDSI)^{1.12}$
Embedded.	$(MM)_{NOM}=2.8 (KDSI)^{1.20}$

Tabla 3.1. Estimación para las ecuaciones para el esfuerzo nominal en el Modelo Intermedio COCOMO.

b) Factores de multiplicación en el desarrollo de sistemas.

La tabla 3.2, presenta factores de multiplicación para el desarrollo de sistemas desde el Modelo Intermedio. Cada atributo de los manejadores de costos,

tiene un conjunto de multiplicadores, los que son claves para un grupo de rangos de proyectos de cada atributo. La escala de rango básico para cada atributo esta detallado en la tabla 3.3.



Gráfica 3.1

Para exponer el uso de la tabla 3.2 y 3.3, considérese el desarrollo de un sistema de 32KDSI, desarrollado como un producto Semi-Capturado⁵, el cual tiene diferentes requerimientos para su realización, porque va de lo natural al uso esperado en forma operacional. La ecuación del esfuerzo nominal, para el modo Semi-Capturado, es mostrado en la tabla 3.1, la que estima que $3.0(32)^{1.12} = 146\text{MM}$, son requeridos para desarrollar un producto Semi-Capturado de 32KDSI, independientemente de cualquier consideración que el producto adquiera.

⁵ Semi-capturado, es una de las tres formas en las que COCOMO agrupa el desarrollo de sistemas.

MANEJADORES DE COSTOS	RANGOS					
	MUYBAJO	BAJO	NOMINAL	ALTO	MUYGRANDE	EXTRA GRAND
Atributos del producto						
RELY Puntualidad requerida	0.75	0.88	1.00	1.15	1.40	
DATA Tamaño de la Base de Datos		0.94	1.00	1.08	1.16	
CPLX Complejidad del Producto	0.70	0.85	1.00	1.15	1.30	1.6
Atributos de la computadora						
TIME Tiempo de ejecución crítico			1.00	1.11	1.30	1.6
STOR Almacenamiento principal crítico			1.00	1.06	1.21	1.5
VIRT Inconsistencia en la máquina virtual.		0.87	1.00	1.15	1.30	
TURN Tiempo de retomo de la computadora		0.87	1.00	1.07	1.15	
Atributos del personal.						
ACAP Capacidad de los analistas.	1.46	1.19	1.00	0.86	0.71	
AEXP Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	
PCAP Capacidad de los programadores.	1.42	1.17	1.00	0.86	0.70	
VEXP Experiencia en la máquina virtual.	1.21	1.10	1.00	0.90		
LEXP Experiencia en el lenguaje de programación	1.14	1.07	1.00	0.95		
Atributos del proyecto						
MCDP Uso de prácticas de programación modulares.	1.24	1.1	1	0.91	0.82	
TOOLS Uso de herramientas en el sistema.	1.24	1.10	1.00	0.91	0.83	
SCEDI Esfuerzo requerido en el desarrollo	1.23	1.08	1.00	1.04	1.10	

Tabla 3.2 Factores de esfuerzo en el desarrollo de Sistemas.

A continuación se muestra la estimación de ajustes para el mismo sistema, en diferentes consideraciones:

1. Un prototipo de 32KDSI o Modelo de demostración de factibilidad de un lenguaje natural o sistemas de activación por voz, pueden tener requerimientos de puntualidad muy bajos, si estos se intentarán utilizar en la producción; y los efectos de las fallas en los sistemas son simples inconvenientes, para que los desarrolladores los corrijan (véase la tabla 3.3). La estimación de esfuerzo ajustada, de la tabla 3.2, para

un producto de esta naturaleza es: $(146\text{MM}) (0.75) = 110\text{MM}$.

2. Un Modelo de planeación de gran-rango o Modelo de estimación de pronósticos, podrían tener un requerimiento relativamente bajo, en la puntualidad del sistema, porque, su impacto operacional es generalmente de gran-rango, y mas fallas pueden incurirse en niveles bajos, fácilmente el usuario puede recuperar pérdidas. La estimación de esfuerzo ajustada, quedaría: $(146\text{MM}) (0.88) = 128\text{MM}$.
3. Un sistema de información gerencial de 32KDSI, o sistemas de control de inventario es representativo de grandes clases de sistemas, con requerimientos de puntualidad nominales. El efecto de una falla en sistemas semejantes, es muchas veces importante, pero generalmente recuperable sin extremas sanciones. Los inajustes, están hechos para rangos nominales; así, el esfuerzo estimado para estos sistemas restantes en 146MM.
4. Unas transacciones, automatización de créditos o sistemas de control de distribución de energía eléctrica de 32KDSI, pueden tener un requerimiento de alta puntualidad. El efecto de fallas, en los sistemas pueden ser pérdidas financieras mayores o inconvenientes humanos masivos. La estimación del esfuerzo ajustado es: $(146\text{MM}) (1.15) = 168\text{MM}$.

5. Una comando y control de sistemas militares, o un sistema de control nuclear de 32KDSI, podrían tener muy-altos requerimientos de puntualidad. Aquí, el efecto de fallos en el sistema, podrían ocasionar la pérdida de vidas humanas. El ajuste de esfuerzo requerido para llevar a cabo la puntualidad deseada es: $(146MM) (1.15) = 168MM$.

	RANGOS					
MANEJADORES DE COSTOS	MUY BAJO	BAJO	NOMINAL	ALTO	MUY GRANDE	EXTRA GRANDE
Atributos del producto						
RELY	Efectos de Inconvenientes descuidados	Bajo, facilidad en recuperar perdidas	Moderado, facilidad en recuperar perdidas	Altas perdidas en financiamiento	Riesgos en vidas humanas	
DATA		$DBytes < 10$ DSI Prog.	$10 \leq D < 100$ P	$100 \leq D < 1000$ P	$D \geq 1000$ P	
CPLX			VER TABLA 3.4			
Atributos de la computadora						
TIME			< 50% Utilizado en la ejecución del tiempo avilitado.	70%	85%	95%
STOR			< 50% Utilizado en el almacenamiento avilitado.	70%	85%	95%
VIRT			Mayores cambios cada 12 meses, y menores cada mes.	Mayores: 2 meses Menores: 2 semanas.	Mayores: 2 meses Menores: 1 semana.	Mayores: 2 meses. Menores: 2 días.
TURN		Interactivo	Promedio de retomo < 4 horas.	4 - 12 horas	> 12 horas	
Atributos del personal.						
ACAP	15° percentil	30° percentil	55° percentil	75° percentil	90° percentil	
AEXP	< 4 meses de experiencia	1 año	3 años	6 años	12 años	
PCAP	15° percentil	30° percentil	55° percentil	75° percentil	90° percentil	
VEXP	< 1 mes de experiencia	4 meses	1 años	3 años		
LEXP	< 1 mes de experiencia	4 meses	1 años	3 años		
Atributos del proyecto.						
MODP	No usa heramientas básicas de microprocesadores.	Se comienza el uso de heramientas básicas de microprocesadores.	Poco uso de heramientas básicas de microprocesadores.	Uso general de heramientas básicas de microprocesadores.	Uso de rutinas completas de heramientas básicas de microprocesadores.	
TOOLS		Mini heramientas básicas.	Midi / Maxi heramientas básicas.	Heramientas de máxima programación.	Adición de requerimientos, diseño y administración.	

Tabla 3.3. Factores de esfuerzo en el desarrollo de sistemas.

La razón por la cual se toman mas meses hombres para desarrollar un sistema con altos requerimientos de puntualidad, son resumidos en la tabla 3.3., esta tabla, indica la típica fuente de reducción o adición de esfuerzos en cada fase de desarrollo, como una función de los niveles requeridos de puntualidad. Así, por ejemplo, en la integración y prueba de un sistema, con muy bajos requerimientos de puntualidad, la tabla 3.2, indica que menos esfuerzo es dedicado al procedimiento de prueba, (las pruebas de requerimientos, calidad de seguridad, administración de la configuración, pruebas de esfuerzo, pruebas sobre-nominales y detalles de la documentación de la estructura). Por otra parte, un producto con "muy-altos" requerimientos de puntualidad, requerirán mas que el esfuerzo usual, para estas actividades, mas esfuerzos se agregan para negocios con un agente independiente de validación y verificación.

Un programa para calcular impuestos sobre la renta, basados en la honradez de ecuaciones de impuesto, podrían dar un rango "muy-bajo" o "bajo", desde la escala de "Operaciones Computacionales", dependiendo esto, de la naturaleza de la ecuación. Si el cálculo del impuesto sobre la renta, involucró una interpolación en un conjunto de tablas de impuestos, este pudo tener un rango nominal.

c) Distribución de fase de actividad, esfuerzo y horario.

El Modelo Intermedio, utiliza algunas estimaciones relacionales para el desarrollo de horarios y fase de distribución de actividades, como las que se usan en el Modelo Básico. Esto es:

- EL horario de desarrollo estimado, TDEV, es calculado desde el Modelo Intermedio, estimando el esfuerzo de desarrollo en MM, usando la ecuación de la tabla 3.1.
- El porcentaje de distribución del esfuerzo y horario por cada fase es obtenido como una función del tamaño de modo y producto, utilizando las tablas 2.3, 2.4 y 2.5).
- El porcentaje de distribución y esfuerzo por actividades y fase es obtenido como una función del tamaño de Modo y producto, usando las tablas de la 2.3 a la 2.5.

3.1.3.- EJEMPLO DE COSTEO:

"SISTEMA DE COMUNICACIÓN DE MICROPROCESADORES".

Supóngase, que se realizará una negociación, con "Megabit Communication Company", sobre el precio en el desarrollo de 10KDSI, el que se considera un producto de modo incrustado para ejecutar funciones de procesamiento de comunicaciones, sobre un microprocesador comercial. La

ecuación del esfuerzo nominal, para la estimación en dicho modo (Tabla 3.1), requiere $2.8(10)^{1.28} = 44\text{MM}$ para desarrollarlo, y ahora se desea determinar el efecto de varias facciones no-nominales a nuestro proyecto, en el desarrollo global de esfuerzo y costo. Por ejemplo, los sistemas de procesamiento de información, generalmente tienen rangos muy altos sobre la escala de complejidad, pero, el plan, es utilizar personal de análisis y programación con alta capacidad en el ramo, los cuales pueden balancear la tendencia a incrementar costos debido a la complejidad. Sin embargo, se tomará un costo promedio de personal de \$56,000 por mes por personas, (incluyendo horas extras), y se desea también descubrir, cuanto se requiere en efectivo para desarrollarlo.

MANEJADOR DE COSTOS	SITUACIÓN	RANGO	MULTIPLICACIÓN DE ESFUERZO
RELY	Uso local del sistema. La recuperación de problemas no son serios.	Nominal.	1.00
DATA	20,000bytes.	Bajo.	0.94
CPLX	Procesamiento de comunicación.	Muy alto.	1.30
TIME	Se usará el 70% de habilitación del tiempo.	Alto.	1.11
STOR	45K de 64K de almacenaje. (70%)	Alto.	1.05
VIRT	Basado sobre un hardware de un microprocesador comercial.	Nominal.	1.00
			CONTINUA....

ACAP	Analista experimentado.	Alto.	0.55
AEXP	Tres años.	Nominal.	1.00
PCAP	Programadores experimentados. CONTINUA	Alto.	0.91
VEXP	6 meses.	Bajo.	1.10
LEXP	12 meses.	Nominal.	1.00
MODP	Utilización de varias técnicas en un año.	Alto.	0.91
TOOL	Niveles de herramientas de microprocesadores básicos.	Bajo.	1.10
SCED	9 meses	Nominal.	1.00
Factor de ajuste del esfuerzo (Factores de esfuerzo del producto).			1.17

Tabla 3.4. Situación actual del sistema.

El análisis de la situación, con respecto a cada uno de los manejadores de costo, es resumido en la tabla 3.4, junto con el resultado del rango de manejadores de costos (de la tabla 3.3) y los factores de multiplicación correspondientes de cada esfuerzo (de la tabla 3.2). Se nota que el 1.30 de incremento es debido a la alta complejidad, esto es balanceado por el 0.86 de reducciones ocasionadas por la alta capacidad de los analistas y programadores. No obstante, cuando se multiplican todos los factores del esfuerzo, el resultado global de ajuste en este es 1.17, representando un 17% de incremento sobre el esfuerzo nominal. El estimado de ajuste resultante, en el desarrollo del esfuerzo y costo del sistema es:

MESES HOMBRE: (44MM) (1.17) = 51MM.

EFFECTIVO : (51MM) (\$6,000) = \$306,000.

a) Análisis de Sensibilidad.

El Modelo Intermedio COCOMO, capacita para realizar análisis de sensibilidad, con respecto a los atributos de manejadores de costos, los cuales, permiten estimar el efecto sobre los costos en el desarrollo del programa cuando se realicen cambios en los niveles de manejadores de costos, para así estimar los cambios en el rango de cada uno.

Por ejemplo, supongamos que se tuvo una opción para contratar personal del proyecto cotizado, a menos costo, pero menos capacitados. Con esta opción, el costo en personal podría ser de \$5,000 por mes, en lugar de \$6,000 por mes, y la capacidad de analistas (ACAP) y programadores (PCAP) pueden ser nominales, en lugar de alta. Estos rangos corresponden a factores de multiplicación de 1.00 para cada uno, en el desarrollo del esfuerzo del sistema, mas que el obtenido con un rango alto de 0.86 (de la tabla 3.2). Esto significa, que la estimación de COCOMO, puede ser ajustada para eliminar un factor de reducción de 0.86; las modificaciones resultantes serían:

FACTOR DE AJUSTE EN EL ESFUERZO: 1.58

MESES HOMBRE: (44MM)(1.58) = 70MM.

$$\text{EFECTIVO} = (70\text{MM}) (\$5,000/\text{MM}) = \$35,000$$

De esta forma, se puede determinar la ventaja de utilizar mas personal capacitado, excediéndose por \$44,000 el efecto de "altos salarios" que podrían pagarse.

Como otro ejemplo, considérese que para \$10,000, Megabit podría comprarse 96K de memoria para los microprocesadores a utilizar, mas que los 64K estipulados. Esto, puede cambiar el almacenamiento principal forzando al (45K / 96K) 47%. El factor de multiplicación resultante puede ser de 1.00 a 1.06, y la estimación en la modificación es:

FACTOR DE AJUSTE EN EL ESFUERZO: 1.10.

MESES HOMBRE : (44MM) (110) = 48MM.

EFECTIVO : (48MM) (\$6,000) = \$288,000.

Esto da como resultado, un valor de salvamento de \$18,000, el cual compensa la inversión de \$10,000 en los costos del hardware. Así, cuando se realiza una negociación con Megabit, se puede proporcionar esta opción como una forma de reducir los costos del sistema, partiendo de los valores de salvamento.

Finalmente, supóngase que conviene un nuevo circuito microprocesador, el cual es equivalente al hardware seleccionado, a un precio de \$20,000 menos de la cantidad necesitada. Sin embargo, este microprocesador tiene una

mínima patente de soporte de herramientas: los compiladores, ensambladores y cargadores son primitivos e inconfiables. Como resultado, las herramientas están entre muy bajo y bajo, con un factor de multiplicación del esfuerzo de 1.24. El costo del desarrollo modificado se estimaría de la siguiente forma:

Factor de ajuste = 1.32.

Meses hombre = (44MM)(1.32) = 58MM.

Efectivo = (58MM)(\$6,000) = \$348,000.

El agregar al costo del sistema \$42,000 no justifican la compra del microprocesador.

3.2.- RENTABILIDA DEL PROYECTO.

Para poder realizar una evaluación económica de un proyecto, es decir, si este es rentable o no, deben considerarse los siguientes conceptos:

HORIZONTE Y PERIODO: Desde el punto de vista económico, la vida de un proyecto es el tiempo durante el cual producirá cobros y pagos; a este tiempo se le denomina **horizonte** de la inversión y puede o no coincidir con la vida del proyecto desde el punto de vista técnico. Por supuesto, generalmente el horizonte no se conoce exactamente a priori, pero es indispensable una estimación del mismo. Se ha de tener presente que el fin del horizonte de un proyecto no coincida, en general, con el fin de la

vida de la empresa, por lo cual hay que considerar con cuidado las condiciones en este instante. Para el caso, período es la división del horizonte en igual duración (cómo un año, un mes, etc.), y generalmente se tratan todos los cobros y pagos de un período como si tuviera lugar en el instante final del mismo.

MOVIMIENTOS DE FONDOS: El calendario de cobros y pagos determina el movimiento de fondos, que es la base para evaluar la rentabilidad de cualquier proyecto de inversión. En definitiva, el movimiento de fondos es una lista de valores, uno por período, que son la diferencia (positiva o negativa) entre el total de cobros y el total de pagos correspondientes al período de que se trate.

DIMENSION DE UN PROYECTO: La dimensión de un proyecto se considera como la cantidad máxima de fondos que requiere, es decir, el valor más negativo de la curva acumulada de movimientos de fondos.

RENTABILIDAD: El proyecto es rentable si el valor de los rendimientos que proporciona es superior al de los recursos que utiliza.

VALOR ACTUAL NETO (VAN): Este valor, es el resultado de llevar todas las inversiones y recuperaciones a un período de tiempo, el cual debe ser el presente.

TASA INTERNA DE RENTABILIDAD (TIR): Es el valor de "i" que anula al VAN.

3.2.1. EVALUACION ECONOMICA.

Para realizar una evaluación Económica de un Proyecto se pueden considerara el Valor Presente Neto o La Tasa Interna de Rendimiento, estas dos evaluación toman en cuenta el valor del dinero a través del tiempo.

a) VALOR PRESENTE NETO.

El valor actual neto o VAN se calcula de la siguiente forma:

$$VAN = \sum_{t=0}^T \frac{S_t}{(1+i)^t}.$$

En donde:

S_t : Fondo.

T : Horizonte.

I : Interés. En la fórmula deben considerarse que la inversiones de los proyectos son negativas.

Este valor se encuentra de sumar los flujos descontados en el presente y restar la inversión inicial que equivale a comparar todas las ganancias esperadas contra todos los desembolsos necesarios para producir esa ganancia, en términos de su valor equivalente en este momento o tiempo cero. Es claro que para aceptar un

proyecto las ganancias deberán ser mayores que los desembolsos, lo cual dará como resultado que el valor presente neto sea mayor que cero.

Si el VPN (Valor Presente Neto) es igual a cero $VPN=0$, no se aumenta el patrimonio durante el horizonte de planeación estudiado. Ahora bien, si el $VPN \geq 0$, la inversión debe aceptarse, si el $VPN < 0$, debe rechazarse. Y el valor presente neto depende de "i", que es la Tasa Mínima Aceptable de Rendimiento (TMAR).

b) Tasa Interna de Rendimiento (TIR).

Este valor, se llama tasa interna de rendimiento, porque supone que el dinero que se gana año con año, se reinvierte en su totalidad. Es decir, que se trata de la tasa de rendimiento generada en su totalidad en el interior de la empresa por medio de la reinversión.

En base a esto, el criterio de aceptación que emplea el método de la TIR, si ésta es mayor que la TMAR, acéptese la inversión, es decir, si el rendimiento de la empresa es mayor que el mínimo fijado como aceptable, la inversión es económicamente rentable.

Por otra parte, se debe agregar el valor de salvamento (VS), que es considerado como el valor en libros fiscal que tengan los activos al término de la operación.

Existen dos formas básicas para la forma de trabajar con el estado de resultados para obtener los Flujos Netos de Efectivo (FNE):

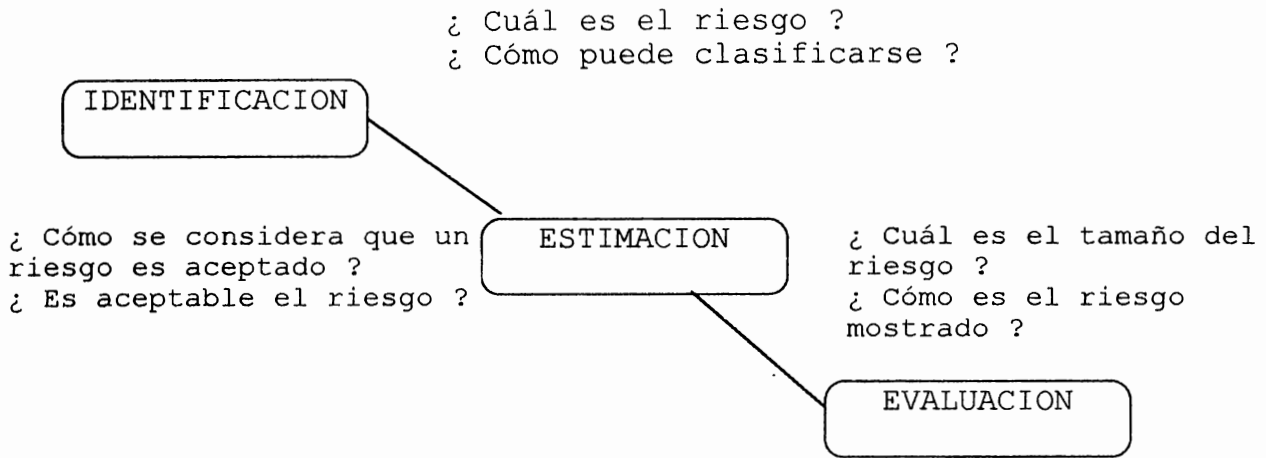
- a) Considerar los FNE del primer años constantes a lo largo del horizonte de planeación.
- b) los efectos inflacionarios sobre los FNE de cada año. Debe tomarse en cuenta que la $TMAR = \text{Tasa de Inflación} + \text{premio al riesgo}$.

3.3.- ANÁLISIS DE RIESGOS.

El análisis de Riesgo identifica las áreas de problemas potenciales, cuantifica los riesgos asociados con estos problemas, asigna los efectos de estos riesgos y genera alternativas de acción para reducir estos riesgos. Aunque, la forma exacta que el análisis de riesgos tomará, depende en gran medida de los requerimientos específicos y prioridades organizacionales, el análisis de riesgos puede dividirse en tres pasos generales:

1. **Identificación** compresiva de los riesgos potenciales usando una metodología estructurada y consistente.
2. Se **Estima** la magnitud de cada riesgo, su consecuencia y la creación de opciones.
3. Se **Evalua** la consecuencia de los riesgos, incluyendo su priorización.

Esto se representa de la siguiente forma:



Quando no existen dudas sobre los sucesos del futuro se puede tener certeza del resultado que producirá una acción. Las decisiones que se toman bajo estas condiciones son decisiones determinísticas o decisiones bajo certeza. En una decisión tomada bajo certeza, el decisor prevee un sólo acontecimiento futuro. Que esta suposición sea cierta o falsa, es otro problema; el hecho es que la decisión se ha tomado, suponiendo un futuro exactamente predecible.

El futuro puede no ser conocido con certeza, sin embargo, si la decisión es tomada suponiendo un futuro exactamente predecible, entonces la decisión se considera tomada bajo certeza. En consecuencia, se dice que las decisiones determinísticas se toman bajo certeza subjetiva, puesto que otros decisores podrían considerar la misma situación como un caso de verdadera

incertidumbre, e incluir toda una serie de resultados posibles en su análisis.

En las situaciones en donde el pronóstico prevé toda una gama de resultados posibles, la decisión de inversión se torna incierta. Cuando todos los resultados se conocen, junto con su probabilidad de ocurrencia, se tiene conocimiento del riesgo que envuelve la decisión. A este tipo de decisión se le conoce como decisión bajo riesgo.

a) EL RIESGO MEDIDO EN LOS FLUJOS.

Para medir el riesgo se puede considerar la probabilidad de alcanzar un VAN (Valor Actual Neto), negativo en cada proyecto.

Para calcular el riesgo se debe :

1. Estimar el VAN representativo de cada proyecto utilizando la tasa libre del país.
2. Estimar para cada VAN, la probabilidad de que el VAN de cada proyecto sea negativo.
3. Determinar el máximo riesgo admisible por la empresa, es decir la máxima probabilidad admisible, de que el VAN sea negativo.
4. Eliminar de la lista de proyectos las que no satisfagan la máxima probabilidad.
5. Seleccionar entre los proyectos restantes aquellos que maximicen el valor Total del VAN, hasta agotar los proyectos con valores actuales netos positivos.

b) FUENTES DE RIESGO EN LA INGENIERIA DE SOFTWARE.

El riesgo involucrado en la ingeniería de Software viene de dos aspectos básicos:

1. Existe el riesgo originado en los negocios y ambientes de Ingeniería de Sistemas, el cual esta al límite del dominio de la ingeniería de software.
2. Hay riesgos que existen entre los procesos y el producto que se generará en la Ingeniería de Software.

Los riesgos de procesos y procedimientos se combinara con los riesgos de los negocios y la Ingeniería de Software para componer o influenciar todos los riesgos de la empresa.

La union de estas fuentes de riesgos forman la totalidad de los riesgos contenidos en el análisis y administración de riesgos en la ingeniería de Software.

En cuanto al Modelo de Cascada, existen riesgos asociados a él. Este Modelo no posee un alto grado poder descriptivo, ya que existen problemas para integrar espacios en los cuales se ubiquen actividades reales de administración de recursos, control de calidad, administración de configuración o verificación y validación. Estas actividades vigilan aspectos que no estan en el Modelo de Cascada, y son de difícil integración a él; esto hace que la administración y control de la empresa sea muy complicada.

Este Modelo carece de cuatro áreas claves:

1. Asume un casi perfecto proceso. Así cuando problemas inevitables ocurren, se tiene dificultades en hacer ajustes necesarios para impedirlos o contenerlos.
2. Fue desarrollado cuando se daban sistemas pequeños con naturaleza Monolítica.
3. No refleja el hecho de que los procesos son parte de otros procesos.
4. No es propiamente para computarizaciones, ya que fue desarrollado cuando la Ingeniería Asistida por computadora no era muy entendida.

CAPITULO IV.

INVESTIGACIONES Y CONCLUSIONES.

4.1.- INTRODUCCION.

La presente investigación, se realizó para obtener información sobre el criterio de desarrollo de Sistemas de Información , desde el punto de vista de " Modelación de Datos ". Considerando que todas las empresas van enfocadas a utilizar herramientas de cuarta generación. Se elaboró una entrevista dirigida, a Jefes ó Analistas de Sistemas de empresas con diferentes rubros (financieras, aseguradoras y de servicio).

4.2.- OBJETIVOS.

Los objetivos de la investigación fueron:

- 1- Verificar si dichas empresas utilizaban una metodología para el desarrollo de un Sistema de Información.
- 2- Determinar en base a que realizan el costo total del proyecto.
- 3- Enfocar un proyecto informático desde el punto de vista " modelo de datos " y no " líneas código ".

4- Determinar e identificar los parámetros para el calculo del costos total y en cada una de las fases del ciclo de vida de un proyecto informático.

4.3.- ESPACIO MUESTRAL.

El espacio muestral de la investigación lo constituirá empresas dedicadas a desarrollar sistemas de información utilizando lenguajes de cuarta generación con sus respectivas herramientas.

4.4.- ELECCION DEL TIPO DE MUESTRA.

La muestra estará representada por individuos que desempeñen cargos de Jefes del departamento de informática y Analistas de sistemas.

4.5.- TAMAÑO DE LA MUESTRA.

Las variables en base a las cuales se ha determinado el número de empresas a examinar corresponden a las empresas que desarrollan sistemas en 4GLs. A nivel nacional.

4.6.- METODO DE RECOLECCION DE DATOS.

El método de recolección de datos a utilizar es el cuestionario dirigido, ya que este permite recoger opiniones, posturas y conductas de las personas entrevistadas en dichas empresas.

4.7.- FORMATO DE LA ENTREVISTA REALIZADA.

Buenos días/tardes, soy estudiante egresado de la Universidad Don Bosco y me encuentro en esta oportunidad, realizando una investigación de campo para el trabajo de tesis, que lleva como título "METODOLOGIA PARA LA DETERMINACION DE COSTOS EN LA REALIZACION DE SISTEMAS DE INFORMACION", por tal motivo solicito su valiosa colaboración y de antemano le doy las gracias.

Nombre: _____

Cargo que desempeña: _____

Empresa: _____

- 1- ¿Utilizan una metodología para el desarrollo de Sistemas de Información?.
- 2- ¿En base a qué clasifican la dimensión del proyecto a desarrollar?.
- 3- ¿Cómo se calcula la duración del proyecto?.
- 4- ¿Qué criterios utiliza para la asignación de personal en las diferentes áreas del proyecto?.
- 5- ¿Cómo se calculan los costos de los Sistemas de Información?.

6- ¿Cuáles son los costos a las modificaciones ó adiciones a los Sistemas de Información, después que el proyecto se ha implementado?.

7- ¿Cómo dimensionaría un proyecto desde el punto de vista "Modelo de Datos" y no "Líneas Código"?.

4.8.- RESULTADOS DE LA ENTREVISTA DIRIGIDA.

Los resultados obtenidos fueron:

- a) La determinación del tamaño del proyecto se realiza en base a la experiencia de las personas involucradas en desarrollo del mismo, así como también en base a la cantidad de tablas relaciones y subsistemas.
- b) La asignación del personal se realiza considerando la experiencia en el uso del lenguaje de programación, manejador de bases de datos o sistema operativo.
- c) La duración de los proyectos se da en base a experiencias anteriores.
- d) No todos desarrollan costeos, ya que dependen de la política de cada empresa.

En forma gráfica los resultados de la entrevista dirigida, los podemos representar de la siguiente manera:

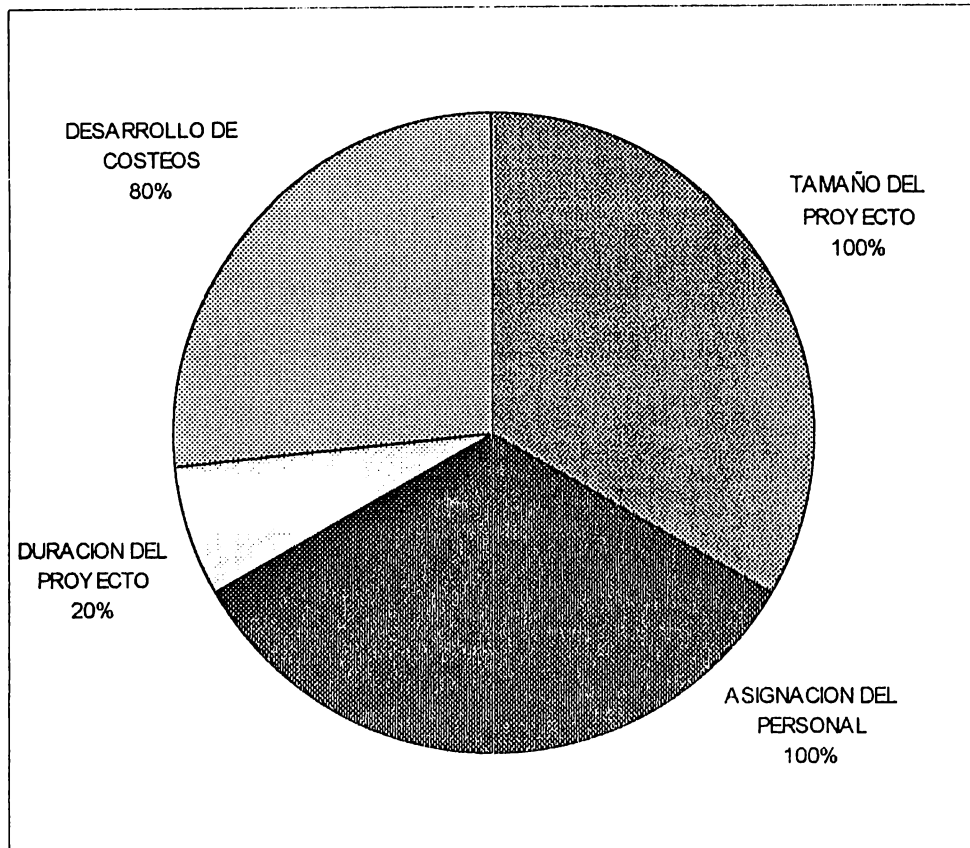


Gráfico 4.1

Considerando los resultados de la encuesta realizada concluimos que :

1. Para determinar el tamaño de un proyecto de Sistemas de Información en base al criterio de Modelo de Datos, se debe considerar los siguientes parámetros :

- Entidades:** Una entidad es algo fundamental de interés para una organización. Una entidad puede ser persona, lugar, cosa, concepto, o evento, real o abstracto. Puede tener un identificado único.

- A mayor cantidad de módulos , se incrementa el número de entidades y relaciones.
- A mayor cantidad de entidades, aumenta la cantidad de módulos y relaciones.

Para encontrar la cantidad de relaciones existentes en un proyecto informático se tiene que:

- Para dos entidades, existen tres posibles relaciones.
- Para tres entidades, existen doce posibles relaciones.
- Para cuatro entidades, existen dieciséis posibles relaciones; etc.

En donde, podemos considerar :

- m = cantidad de módulos.
- n = cantidad de entidades.
- x = cantidad de relaciones.

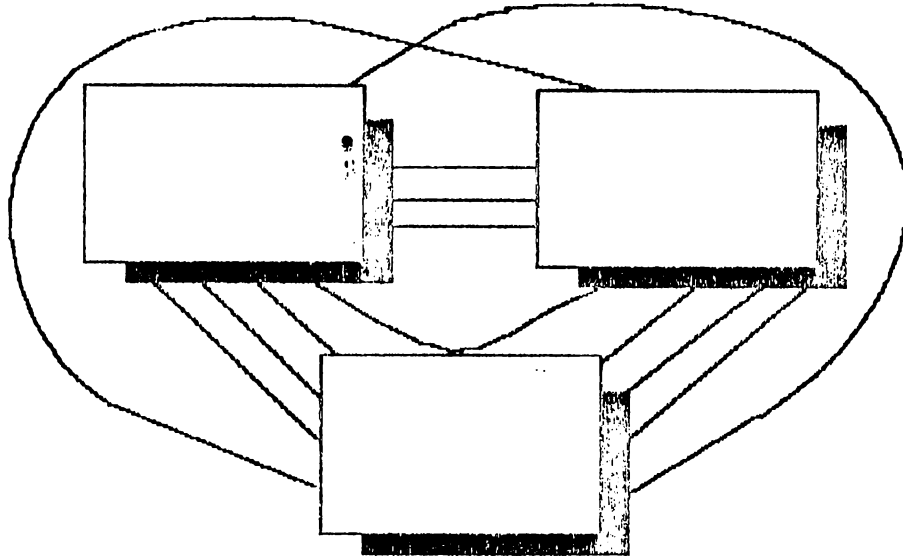
Así tenemos que:

$n = 12.5 \times m$ Fórmula 4.1.

Para $n > 2$, $x = 4n$ Fórmula 4.2.

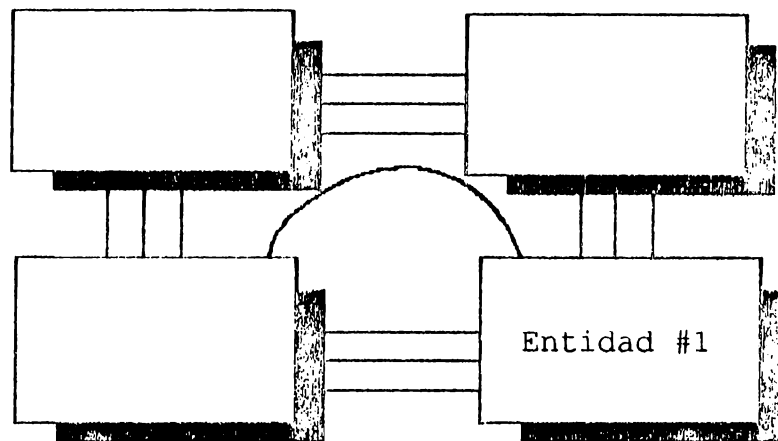
En forma gráfica los podemos representar de la siguiente manera:

Para TRES Entidades tenemos 12 relaciones máximas.



De tal forma que para cada entidad parten cuatro posibles relaciones. Llegando a tener un máximo de $4(3 \text{ Entidades}) = 12 \text{ Relaciones}$.

Para CUATRO Entidades tenemos 16 relaciones.



Para cada entidad existirán un máximo de 4 relaciones, como las que se muestran en la entidad #1, y así sucederá para cada una de las entidades, llegando a tener un máximo de $4(4 \text{ Entidades}) = 16$ relaciones.

Lo anterior se resume en la siguiente tabla:

	PEQUEÑO	MEDIANO	GRANDE	MUY GRANDE
MODULOS MAXIMOS	4	10	32	32⇒
ENTIDAD MAXIMAS	50	125	400	400 ⇒
RELACIONES MAXIMAS	200	500	1600	1600⇒

Tabla 4.1. Cantidad de Módulos, entidades y relaciones Máximas en cada proyecto.

La experiencia, en el uso de herramientas de cuarta generación, para el desarrollo de sistemas de información se clasifica de la siguiente manera:

Alta experiencia: Aquellos Analistas de Sistemas que tienen aproximadamente 6 años en el desarrollo de sistemas.

Media Experiencia: Aquellos Analistas de Sistemas que tienen aproximadamente 4 años en el desarrollo de sistemas.

Baja Experiencia: Aquellos Analistas de Sistemas que tienen aproximadamente 2 años en el desarrollo de sistemas.

Sin Experiencia: Aquellos Analistas de Sistemas que nunca han utilizado herramientas de Cuarta Generación.

Así tenemos que, la duración en meses depende de la experiencia del Analista. Con experiencia Media en un proyecto pequeño se tardará un máximo de 3 meses utilizando herramientas de cuarta generación; lo mismo sucede con las demás categorías de experiencia, llegando a tener la siguiente tabla:

	PEQUEÑO (MESES MAXIMOS)	MEDIANO (MESES MAXIMOS)	GRANDE (MESES MAXIMOS)	MUY GRANDE (MESES MAXIMOS)
ALTA	1.5	3	6	12
MEDIA	3	6	12	24
BAJA	6	12	24	48
SIN EXPERIENCIA	12	24	48	96

Tabla 4.2. Duración de los proyectos en Base a la Experiencia de los Analistas y al tipo de proyecto.

CAPITULO V.

METODOLOGIA PARA LA DETERMINACION DE COSTOS EN LA REALIZACION DE SISTEMAS DE INFORMACION.

5.1.- INTRODUCCION.

En este capítulo, se presenta la metodología para estimar el costo de un Sistema de Información, para ello se evalúan cada una de las fases del Ciclo de Vida de Sistemas , encontrando los costos para cada una de ellas y llegando al final a una sumatoria de cada fase para encontrar el costo final del proyecto.

Cuando una empresa realiza un proyecto, es por dos razones:

- Aplicar Re - Ingeniería total.
- Cambiar de Plataforma (Software, Hardware o ambas).

En ambos casos tiene que realizarse una inversión en Hardware y/o Software. En la metodología que se presenta, se consideran aspectos subjetivos en el desarrollo de un sistema (personal involucrado, Hardware, Software, etc.) cuantificándolos para realizar un proyecto en el cual el riesgo sea medible y controlable.

CONSIDERACIONES .

- El salario promedio de Analistas que utilizan 4GLs y sus herramientas correspondientes esta dado en la tabla siguiente:

EXPERIENCIA	SALARIO
ALTA	¢9,000
MEDIA	¢7,000
BAJA	¢5,000
SIN EXPERIENCIA	¢3,000

Tabla 5.1

- El salario de Analistas Programadores que utilizan 3GLs y sus herramientas correspondientes esta dado en la siguiente tabla:

EXPERIENCIA	SALARIO
ALTA	¢5,000
MEDIA	¢4,000
BAJA	¢3,500
SIN EXPERIENCIA	¢2,500

Tabla 5.2

- El precio aproximado de las licencias de los diferentes productos 4GL es:

ADMINISTRADO DE BASE DE DATOS	Por PC
SYBASE	¢100,000
ORACLE	¢100,000
INFORMIX	¢ 96,000

Tabla 5.3 (Hasta la fecha).

- Las inversiones en Hardware no son cuantificadas, ya que varían en base al mercado existente.
- Después de que el personal involucrado (al implementar un Nuevo sistema) , es capacitado existe un ajuste salarial para evitar de esta forma que los empleados no se desmotiven y tiendan a la fuga por falta de incentivos económicos.
- El salario del Personal del Departamento de Organización y Métodos, Usuario Final y Usuario de alto nivel es ¢ 2,500, ¢ 2,000 y ¢ 3,000 respectivamente, de este salario se parte para encontrar los salarios por hora laboral (Estos valores corresponden a la fecha respectiva de la elaboración de este documento).
- El reajuste salarial será determinado en base a la política de la empresa.
- El incremento de costo por no finalizar el proyecto en el tiempo estipulado se estima en base al contrato establecido, o en base a la política de la empresa.
- Los meses de duración del proyecto (MDP) se encuentran en base a la tabla 4.2.

- Para cada fase se considerará un 10% de incremento de imprevistos. ($IMP, IMP = 10\% \times \text{Costo de cada FASE}$).

5.2.- PASOS DE LA METODOLOGIA.

I.DETERMINACION DE LA CANTIDAD DE MODULOS, ENTIDADES Y RELACIONES DEL PROYECTO.

Después del Estudio de Requerimientos se tiene los parámetros necesarios para establecer la cantidad de módulos existentes en el sistema. De tal forma que se pueda calcular las Entidades y Relaciones existentes en la misma, considerando las fórmulas 4.1 y 4.2.

Por ejemplo:

- a) Para un proyecto con tres módulos existirán un máximo de $12.5 \times 3 = 38$ Entidades y un máximo de $38 \times 4 = 152$ Relaciones, para el caso, este proyecto, es considerado un "Proyecto Pequeño", ya que esta en el rango de 1-4 Módulos, 1 - 50 Entidades y de 3 - 200 Relaciones, este criterio se determina a partir de la tabla 4.1.
- b) Para un proyecto con Cinco módulos existirán un máximo de $12.5 \times 5 = 63$ Entidades y un máximo de $63 \times 4 = 252$ Relaciones, para el caso, este proyecto, es considerado un "Proyecto Mediano", ya que esta en el rango de 4-10 Módulos, 1 - 125 Entidades y de 3 - 500

Relaciones, este criterio se determina a partir de la tabla 4.1.

II.DETERMINACION DE LA CANTIDAD DE MESES QUE SE TARDARA EN DESARROLLAR EL SISTEMA.

Para determinar la cantidad de meses que se tardará una persona en desarrollar un sistema debe considerarse la experiencia del personal involucrado, de tal forma que para un Analista experimentado se tardará en desarrollar un proyecto pequeño en 1 ½ meses, (para considerar este valor refiérase a la tabla 4.2).

Por ejemplo:

- a) Para un proyecto con tres módulos existirán un máximo de 38 Entidades, de tal forma que un Analista con Alta experiencia se tardará 1 ½ meses.

- b) Para un proyecto con tres módulos existirán un máximo de 38 Entidades, de tal forma que un Analista sin experiencia se tardará 8 meses.

III. ANALISIS COMPARATIVO DE LOS TIEMPOS REALES CON LOS TIEMPOS ESTIMADOS.

En base a la cantidad de meses obtenido en el paso (4), se estiman los tiempos que tuvieron que haberse asignado al estudio de factibilidad y análisis de requerimientos, considerando la tabla siguiente:

Estudio de Factibilidad : 10 % Meses de duración del proyecto (tabla 4.2).	Análisis de Requerimientos : 20 % Meses de duración del proyecto (tabla 4.2).
--	---

Si los tiempos reales con los estimados son inferiores, existirán holguras, las cuales pueden dedicarse a posibles retrasos en las siguientes fases. Si los tiempos reales son mayores que las estimadas, el proyecto esta atrasado, razón por la cual, debe seguirse la ruta crítica del proyecto, considerando la siguiente tabla:

Diseño y Codificación: 50% de los meses de duración del proyecto (tabla 4.2).	Prueba e Implementación: 15% de los Meses de duración del Proyecto.
---	---

IV. IDENTIFICACION DE LAS VARIABLES EN EL ESTUDIO DE FACTIBILIDAD.

Objetivo de esta fase : Investigar la información que el usuario necesita.

En esta fase , se debe establecer, en base a los criterios de los grupos multidisciplinarios, si el proyecto será viable o no; considerando los siguientes aspectos:

- Inversión en Software (IS).
- Inversión en Hardware (IH).
- Reajuste Salarial (RS).
- Cambios en los procesos o formularios de los sistemas manuales o automatizados (CP).
- Inversión en Capacitación (IC).
- Inversión de Tiempo Productivo por el Usuario (ITPU).
- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en los Salarios del Personal de Organización y métodos (ISOM).
- Duración de la Fase de Factibilidad (DFF).
- Costo global del Proyecto (CGP).
- Duración de la fase de estudio de factibilidad (DFF).

$$DFF = MDP \times 10\%.$$

Fórmula 5.1.

El costo de la fase de Estudio de Factibilidad (CEF), se determina en base a la siguiente fórmula:

$$CEF = IS + IH + RS + CP + IC + (ITPU + ISAS + ISOM) \times (DFF) + IMP. \quad \text{Fórmula 5.2.}$$

El costo total global del proyecto se determina en base a la siguiente fórmula:

$$\text{CGP} = \text{IS} + \text{IH} + \text{RS} + \text{CP} + \text{IC} + (\text{ITPU} + \text{ISAS} + \text{ISOM}) \times (\text{MDP}) + \text{IMP}. \quad \text{Fórmula 5.3.}$$

Con el valor de CGP, se construye un flujo de caja en el período de duración del proyecto, y se realiza con este flujo, un Análisis de riesgo para determinar si el proyecto se puede aceptar o rechazar. En este caso, se debe comparar la Tasa de rendimiento de la empresa con la tasa de rendimiento del proyecto (que se calcula en base a los Valores Actuales Netos, mostrados en el flujo de caja), considerando que:

- Si la Tasa del proyecto es menor que la tasa de rendimiento de la empresa, el proyecto es de alto grado de certeza.
- Si la Tasa del proyecto es mayor que la tasa de rendimiento de la empresa, el proyecto es considerado de alto riesgo.

V. ESTABLECIMIENTO DE LAS VARIABLES EN EL ANÁLISIS DE REQUERIMIENTOS.

Objetivo de esta fase : Identificar el problema del que surge la necesidad del desarrollo del sistema y

proponer soluciones generales, satisfaciendo los objetivos pretendidos.

En esta fase, se determinan, las necesidades del usuario, de tal forma que se comience a detallar cada una de las relaciones.

Se deben considerar en esta fase, conceptos de Reingeniería, de tal forma, que si es necesario cambiar un proceso, formulario, etc., se tiene que hacer en pro del correcto funcionamiento del sistema.

Se consideran los siguientes aspectos:

- Inversión de Tiempo Productivo por el Usuario (ITPU).
- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Incremento de costo por no finalizar el proyecto en el tiempo estipulado (ICPTE).
- Inversión en los Salarios del Personal de los diferentes departamentos para usuario (Gerentes Mercadotecnia, Compras, Producción o Contabilidad (ISDD).
- Duración de la Fase de Análisis de Requerimientos (DFR).

$$DFR = MDP \times 20\%$$

Fórmula 5.4.

El costo de la fase de Análisis de Requerimientos (CAR), se determina en base a la siguiente fórmula :

$$CAR = (ITPU + ISAS + ICPTE + ISDD) \times (DFR) + IMP$$

Fórmula 5.5.

Se establecerá por otra parte, las relaciones existentes del Sub-Sistema (módulo) con los demás sub-sistemas dentro del Supra-Sistema.

VI.DETERMINACION DE LAS VARIABLES EN LA FASE DE DISEÑO.

Objetivo de esta fase : aceptar los requisitos del análisis de sistemas como base principal del proceso de diseño, para realizar el diseño físico del nuevo sistema, con su respectiva documentación.

Usa los requerimientos especificados para desarrollar el diseño del software.

Se consideran los siguientes aspectos:

- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en los Salarios de los Diseñadores de Sistemas (ISD).
- Duración de la fase de diseño (DFD).

$$DFD = MDP \times 30\%.$$

Fórmula 5.6.

El costo de la fase de Diseño (CD), se determina en base a la siguiente fórmula :

$$CD = (ISAS + ISD) \times (DFD) + IMP$$

Fórmula 5.7.

VII. IDENTIFICACION DE LAS VARIABLES EN LA FASE DE CODIFICACION.

Objetivo de esta fase : Codificar y documentar los módulos individuales que constituirán el sistema.

En esta fase, se alcanza el proceso que transforma el diseño en un lenguaje de programación.

Se consideran los siguientes aspectos:

- Inversión en los Salarios de los Programadores (ISP).
- Duración de la fase de Codificación (DFC).

$$DFC = MDP \times 20\% \qquad \text{Fórmula 5.8.}$$

El costo de la fase de Codificación (CC), se determina en base a la siguiente fórmula :

$$CC = (ISAS + ISP) \times (DFC) + IMP \qquad \text{Fórmula 5.9.}$$

VIII. ESTABLECIMIENTO DE LAS VARIABLES EN LA FASE DE PRUEBA.

Objetivo de esta fase : Comprobar que el sistema cumple con eficiencia todos los objetivos propuestos, tratando de encontrar si existen fallas en este sentido.

Valida el diseño para verificar que el software satisface los requerimientos.

Se consideran los siguientes aspectos:

- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en los Salarios de Usuarios finales (ISUF).
- Duración de la fase de Prueba (DFP).

$$DFP = MDP \times 7.5\%$$

Fórmula 5.10.

El costo de la fase de Prueba (CP), se determina en base a la siguiente fórmula :

$$CP = (ISAS + ISUF) \times (DFP) + IMP$$

Fórmula 5.11.

IX.DETERMINACION DE LAS VARIABLES EN LA FASE DE IMPLEMENTACION.

Objetivo de esta fase : La completa aceptación del usuario.

Durante esta fase, el software y toda la documentación requerida son entregados al usuario.

Se consideran los siguientes aspectos:

- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en los salarios de Usuarios finales (ISUF).
- Inversión en capacitación (IC).
- Duración de la fase de Implementación (DFI).

$$DFI = MDP \times 7.5\% \qquad \qquad \qquad \text{Fórmula 5.12.}$$

El costo de la fase de Implementación (CI), se determina en base a la siguiente fórmula :

$$CI = (ISAS + ISUF + IC) \times (DFI) + IMP \qquad \qquad \qquad \text{Fórmula 5.13.}$$

X. IDENTIFICACION DE LAS VARIABLES EN LA FASE DE MANTENIMIENTO.

El objetivo de esta fase : Proveer mejores bases para realces futuros.

Incluye la reparación, modificación y ampliación del software para su vida restante.

Se consideran los siguientes aspectos:

- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en capacitación (IC).
- Duración de la fase de Mantenimiento (DFM).

$$DFM = MDP \times 5\% \qquad \qquad \qquad \text{Fórmula 5.13.}$$

El costo de la fase de Mantenimiento (CM), se determina en base a la siguiente fórmula:

$$CM = (ISAS + IC) \times (DFM) + IMP$$

Fórmula 5.14.

5.3. EJEMPLO DE COSTEO:

"Realización de un sistema de Administración, Financiero y Contable"

5.3.1. INTRODUCCION.

El Sistema a desarrollar, debe ser capaz de llevar un registro contable (contabilidad general), así como también controles de cuentas por cobrar, cuentas por pagar, etc. Debe realizar también asignación de presupuestos a las diferentes áreas de la empresa. Para ello se compraran cuatro Computadoras, las cuales se conectarán en red (una de las cuatro será el servidor), y se adquirirá la licencia de Sybase.

En base a estos requerimientos generales, la empresa desea que se realice un estudio por parte del departamento de Informática para determinar cual será el costo total del proyecto, su duración y si se debe invertir o no. Debe considerarse además que las personas que participarán en el desarrollo de cada módulo no tienen experiencia en el uso de Herramientas de Cuarta

Generación, no se posee Hardware, ni licencias de ninguna herramienta de Cuarta Generación.

5.3.2 PASOS DE LA METODOLOGIA.

I. DETERMINACION DE LA CANTIDAD DE MODULOS, ENTIDADES Y RELACIONES DEL PROYECTO.

El sistema esta constituido de tres módulos, los cuales son contabilidad, administración y financiero, en base a esto se tiene que:

Cantidad de Entidades posibles:

$$12.5 \times 3 = 38 \text{ Entidades Máximas.}$$

Cantidad de relaciones posible:

$$38 \times 4 = 152 \text{ Relaciones Máximas,}$$

de los cual concluimos que el sistema estará clasificado como un "Proyecto Pequeño".

II.DETERMINACION DE LA CANTIDAD DE MESES QUE SE TARDARA EN DESARROLLAR EL SISTEMA.

Para el caso, un analista sin experiencia en el uso de Herramientas de Cuarta Generación, se tardará en el proyecto 12 meses (Ver tabla 4.2).

Por tanto la variable MDP será igual a 12 (MDP = 12).

III. ANALISIS COMPARATIVO DE LOS TIEMPOS REALES CON LOS TIEMPOS ESTIMADOS.

Este paso debe realizarse cuando cada fase haya terminado, y se pueda estimar un juicio sobre el avance o retraso del proyecto.

IV. IDENTIFICACION DE LAS VARIABLES EN EL ESTUDIO DE FACTIBILIDAD.

En este paso, se determina un costo general de todo el proyecto, tratando de incrementar los costos para tener holguras, de esta forma se reducen riesgos al implementar el Proyecto. Para ello se consideran las siguientes variables con sus respectivos valores:

- Inversión en Software (IS = ₡ 300,000),
- Inversión en Hardware (IH = ₡ 45,000).
- Reajuste Salarial (RS = ₡ 1,080).
- Cambios en los procesos o formularios de los sistemas manuales o automatizados (CP = ₡ 2,000).
- Inversión en Capacitación (IC = ₡ 4,000).
- Inversión de Tiempo Productivo por el Usuario (ITPU = ₡ 2,000).
- Inversión en los Salarios de los Analistas de Sistemas (ISAS = ₡ 9,000).

- ☑ Inversión en los Salarios del Personal de Organización y métodos (ISOM = \$ 2,500).

De donde podemos calcular : (en base a las fórmulas 5.1, 5.2 y 5.3)

$$CGP=300,000+45,000+1,080+2,000+4,000+(2,000+9,000+2,500)+IMP.$$

$$CGP = 352,080 + 162,000 + IMP$$

$$CGP = 514,080 + 51408 \approx \$ 570,000.$$

El valor anterior, determina un costo incrementado, ya que el personal del Departamento de OYM, y el usuario final se consideran trabajando a tiempo completo, otorgando con esto holguras al proyecto. Al calcular la sumatoria de los costos de cada fase, estos no deben exceder el antes calculado. Lo restante sobre el valor calculado será el valor cuantificado de la holgura del proyecto.

Una vez calculado el CGP, se pueden realizar Análisis de Riesgos sobre ese valor encontrado, considerando los flujos de caja que el proyecto tendrá. Así tenemos que:

- La tasa interna de rendimiento, que se va a considerar, para el presente proyecto es del 8% ($i = 8\%$).
- Las inversiones iniciales serán:

- ⇒ Hardware: ₡ 45,000.
- ⇒ Software: ₡ 300,000.
- ⇒ Cambios en procesos y formularios: ₡ 2,000.
- Inversión en salarios al final del año: ₡ 162,000.
- Los ingresos que se pueden tener al final del proyecto son:
 - ⇒ Beneficios (Reducción de tiempos, incremento de la producción, etc.) ₡ 200,000.
 - ⇒ Valor de recuperación en Hardware ₡ 40,500.
 - ⇒ Valor de recuperación en Software ₡ 270,000.
 - ⇒ Valor de finalizar a tiempo el proyecto: ₡ 50,900.

Para encontrar el valor de i , del proyecto se utiliza una fórmula de Ingeniería económica, para el caso, quedaría de la siguiente manera:

$$-347,000 - \frac{162,000}{(1 + i_p)^n} + \frac{561,000}{(1 + i_p)^n} = 0$$

Despejando la ecuación anterior:

$$\begin{aligned} -347,000i_p &= -52,400 \cdot \\ i_p &= 15\% \end{aligned}$$

Por tanto, como $i_p > 8\%$, el proyecto es considerado de alto riesgo.

DURACION Y COSTO DEL ESTUDIO DE FACTIBILIDAD.

La distribución de horas en un mes que corresponde a reuniones será (en esta fase):

- Usuario final 8 horas.
- Personal de OYM 8 horas.

$$DFF = MDP \times 10\%.$$

$DFF = 1.2 \text{ MESES.}$

$$CEF = (124.8 + 100 + 9,000) \times 1.2 + IMP$$

$CEF = 11,070 + 1,107 = \text{¢ } 12,177.$
--

V. ESTABLECIMIENTO DE LAS VARIABLES EN EL ANALISIS DE REQUERIMIENTOS.

En este paso, se incrementan la cantidad de horas que invertirá el usuario final y el usuario de alto nivel, a 15 y 10 respectivamente. De tal forma que la duración y costos que se encuentran en la fórmula 5.4 y 5.5 son:

$$DFF = MDP \times 20\% = 2.4$$

$$CAR = (187.5 + 9,000 + 21.9) \times 2.4 + IMP$$

$CAR = 22,576 + 2,258 = \text{¢ } 24,834$

VI. DETERMINACION DE LAS VARIABLES EN LA FASE DE DISEÑO.

Los Analistas y Diseñadores de Sistemas se dedican a tiempo completo en el desarrollo del mismo. Considerándose que la duración y costos que se encuentran utilizando la fórmula 5.6 y 5.7 son:

$$\text{DFD} = 12 \times 30\% = 3.6 \text{ meses.}$$

$$\text{CD} = \text{ISAS} + \text{ISD} + \text{IMP} = (9,000 + 9,000) \times 3.6$$

$\text{CD} = 64,800 + 6,480 = \text{¢ } 71,280$

VII. IDENTIFICACION DE LAS VARIABLES EN LA FASE DE CODIFICACION.

Los programadores invierten su tiempo para transformar el diseño lógico a una Implementación real. Al utilizar las fórmula 5.8 y 5.9 tenemos que:

$$\text{DFC} = 12 \times 20\% = 2.4$$

$$\text{CC} = \text{ISP} \times 2.4 = 9,000 \times 2.4$$

$\text{CC} = 21,600 + 2,160 = \text{¢ } 23,760$

VIII. ESTABLECIMIENTO DE LAS VARIABLES EN LA FASE DE PRUEBA.

Se involucra al usuario durante cinco días para realizar pruebas piloto sobre el sistema, estos días los debe dedicar para reeditar datos o acomodarse a nuevas formas de captura de datos, etc. Para encontrar el costo de esta fase se deben utilizar las fórmulas 5.10 y 5.11. Así tenemos que:

$$DFP = 12 \times 7.5\% = 0.9 \text{ MESES.}$$

$$CP = (9,000 + 500) \times 0.9$$

$$CP = 8,550 + 855 = \text{¢ } 9,405$$

IX.DETERMINACION DE LAS VARIABLES EN LA FASE DE IMPLEMENTACION.

El valor de la capacitación dependerá de las horas que se impartirán a los usuarios finales, para el caso se han considerado 10 horas de capacitación a un costo por hora de 400. La fórmula para encontrar el costo de esta fase y la duración de la misma determina que (Fórmula 5.12 y 5.13):

$$DFI = 12 \times 5\% = 0.9 \text{ MESES.}$$

$$CI = (9,000 + 2,000 + 4,000) \times 0.9.$$

$$CI = 13,500 + 1,350 = \text{¢ } 14,850$$

X. IDENTIFICACION DE LAS VARIABLES EN LA FASE DE MANTENIMIENTO.

El costo de esta fase esta determinado por la fórmula 5.15 y la duración por la 5.14. De tal forma que tenemos:

$$DFM = 12 \times 5\% = 0.6 \text{ MESES.}$$

$$CM = (9,000) \times 0.6 = 5,400 + 540 = \text{¢ } 5,940$$

Finalmente, en base a la sumatoria de los costos que anteriormente se determinaron en cada una de las fases del desarrollo del sistema dan como resultado que el costo final del proyecto es:

$$CTP = 147,496 + 352,080.$$

$$CTP = \text{¢ } 162,246.$$

Valor que no excede al calculado anteriormente, que nos da como holgura : ¢ 15,452.

5.3.3.- VALORES COMPARATIVOS.

FASE	REAL	ESTIMADO SIN EXPERIENCIA	ESTIMADO CON EXPERIENCIA ALTA.
Estudio de Factibilidad.	¢ 3,600	¢12,177	¢ 4,493
Análisis de Requerimientos.	¢ 3,600	¢24,834	¢ 7,184
Diseño.	¢ 32,000	¢71,280	¢26,730
Codificación.	¢ 21,600	¢23,760	¢ 8,917
Prueba.	¢ 7,200	¢ 9,405	¢ 3,403
Implementación	¢ 3,600	¢14,850	¢ 2,723
Mantenimiento	-----	¢ 5,940	¢ 2,228
Costo Total:	¢ 71,000.	¢162,246	¢55,678

Tabla. 4.7. Cuadro comparativo de costos.

5.3.4.- CONCLUSIONES.

Para un proyecto en el cual se tenga que contratar a programadores/analistas sin experiencia en la utilización de herramientas 4Gls para el desarrollo de Sistemas de Información, se esta incurriendo a un gasto de ¢ 162,246 (Tabla 4.7) solamente en las diferentes fases del ciclo de vida del sistema, es decir, sin considerar la inversión de Hardware, software y gastos de procesos. Si

para el caso, se hubiera contratado a programadores de alta experiencia, el costo sería de \$ 55,678.

En el Estudio de factibilidad se evaluo la inversión anterior (\$ 162,246) por medio de un análisis de riesgo, utilizando los valores actuales netos (VAN) con diferentes tasas de rendimiento, determinando que era de alto riesgo. La experiencia de los programadores, conllevó a un incremento en el período de finalización del proyecto (12 meses), así como también se desarrollo un producto de baja calidad, por ejemplo:

Un proyecto que se inicio el mes de mayo de 1996, con las mismas condiciones antes mencionadas (Analistas/programadores sin experiencia en el uso de 4GL y sus respectivas herramientas). Debe de tener el progreso siguiente al mes de diciembre del mismo año:

sistema.	del Proyecto real.	tiempo comparado.
Desde el estudio de factibilidad hasta el diseño.	8 meses.	6 meses
Codificación.	No se ha llevado a cabo	2.4 meses.
Prueba.	No se ha llevado a cabo	0.9 meses.
Implementación	No se ha llevado a cabo	0.9 meses.
Mantenimiento	No se ha llevado a cabo	0.6 meses.
Total de meses que faltan para terminar el proyecto (Calculado):		5 meses.

Tabla 4.8.

Con esto se observa que el proyecto tendrá que incurrir a 5 meses mas de duración, siempre y cuando se realice un trabajo planificado (Ver anexos en la página 126). Esto trae consigo, que el proyecto terminará, en 13 meses, un mes mas de lo establecido, lo cual trae un mínimo apróximado de:

- Salarios: ₡ 9,000.
- Costo por no término el proyecto en el período estipulado: ₡ 51,500.

5.3.5.- RESUMEN DE FORMULAS.

Las variables que se consideran para el desarrollo de un sistema, que involucran costos son:

- Inversión en Software (IS).
- Inversión en Hardware (IH).
- Reajuste Salarial (RS).
- Cambios en los procesos o formularios de los sistemas manuales o automatizados (CP).
- Inversión en Capacitación (IC).
- Inversión de Tiempo Productivo por el Usuario (ITPU).
- Inversión en los Salarios de los Analistas de Sistemas (ISAS).
- Inversión en los Salarios del Personal de Organización y métodos (ISOM).
- Duración de la Fase de Factibilidad (DFF).
- Costo global del Proyecto (CGP).
- Imprevistos(IMP, 10% del costo total de la fase).
- El costo de la fase de Estudio de Factibilidad (CEF).
- Incremento de costo por no finalizar el proyecto en el tiempo estipulado (ICPTE).
- Inversión en los Salarios del Personal de los diferentes departamentos para usuario (Gerentes Mercadotecnia, Compras, Producción o Contabilidad (ISDD).

- El costo de la fase de Análisis de Requerimientos (CAR).
- Duración de la fase de Análisis de Requerimientos (DFR).
- Inversión en los Salarios de los Diseñadores de Sistemas (ISD).
- Duración de la fase de diseño (DFD).
- El costo de la fase de Diseño (CD).
- Inversión en los Salarios de los Programadores (ISP).
- El costo de la fase de Codificación (CC).
- Duración de la fase de Codificación (DFC).
- Inversión en los Salarios de Usuarios finales (ISUF).
- El costo de la fase de Prueba (CP).
- Duración de la fase de Prueba (DFP).
- El costo de la fase de Implementación (CI).
- Duración de la fase de Implementación (DFI).
- Duración de la fase de Mantenimiento (DFM).
- El costo de la fase de Mantenimiento (CM).

Listado de Fórmulas que se utilizan para la determinación de costos en el desarrollo de sistemas de información:

VARIABLES	NO. FORMULA	DESCRIPCION
$CEF=IS+IH+RS+CP+IC+(ITPU+ISAS+ISOM) \times (DFE) + IMP$	FORMULA 5.1	COSTO DE FACTIBILIDAD
$DFE=MDP \times 10\%$	FORMULA 5.2	DURACION DE FACTIBILIDAD
$CGP=IS+IH+RS+CP+IC+(ITPU+ISAS+ISOM) \times (MDP) + IMP$	FORMULA 5.3	COSTO GLOBAL DEL PROY.
$DFR=MDP \times 20\%$	FORMULA 5.4	DURACION REQUERIMIENTO
$CAR=(ITPU+ISAS+ICPTE+ISDD) \times (DFR) + IMP$	FORMULA 5.5	COSTO REQUERIMIENTOS
$DFD=MDP \times 30\%$	FORMULA 5.6	DURACION DISEÑO
$CD=(ISAS+ISD) \times (DFD) + IMP$	FORMULA 5.7	COSTO DISEÑO
$DFC=MDP \times 20\%$	FORMULA 5.8	DURACION CODIFICACION
$CC=(ISAS + ISP) \times (DFC) + IMP$	FORMULA 5.9	COSTO CODIFICACION
$DFP=MDP \times 7.5\%$	FORMULA 5.10	DURACION DE PRUEBA
$CP=(ISAS + ISUF) \times (DFP) + IMP$	FORMULA 5.11	COSTO DE PRUEBA
$DFI=MDP \times 7.5\%$	FORMULA 5.12	DURACION IMPLEMENTACION
$CI=(ISAS + ISUF + IC) \times (DFI) + IMP$	FORMULA 5.13	COSTO DE IMPLEMENTACION
$DFM=MDP \times 5\%$	FORMULA 5.14	DURACION MANTENIMIENTO
$CM=(ISAS + IC) \times (DFM) + IMP$	FORMULA 5.15	COSTO MANTENIMIENTO

Tabla 4.9.

ANEXOS

encadenamiento.

FASE	Plan y Requerimientos					Diseño de Producto					Programación					Integración y Prueba					Desarrollo					Mantenimiento				
	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL	S	I	M	L	VL
Tamaño del producto																														
Porcentaje de las Fases	8	8	8	8	8	18	18	18	18	18	60	57	54	51	48	22	25	28	31	34										
Porcentaje de Actividades																														
Análisis de requerimientos	50	48	46	44	42	10	10	10	10	10	3	3	3	3	3	2	2	2	2	2	4	4	4	4	4	6	6	6	5	5
Diseño de productos	12	13	14	15	16	42	42	42	42	42	6	6	6	6	6	4	4	4	4	4	12	12	12	12	12	11	11	11	11	11
Programación	2	4	6	8	10	10	11	12	13	14	55	55	55	55	55	32	36	40	44	48	42	43	43	44	45	38	39	39	40	41
Plan de prueba	2	3	4	5	6	4	5	6	7	8	4	5	6	7	8	3	3	4	4	5	4	4	5	6	7	3	3	4	5	6
Verificación y Validación	6	7	8	9	10	6	7	8	9	10	8	9	10	11	12	30	20	25	23	20	12	13	14	14	14	12	13	14	14	14
Oficina de proyectos	16	14	12	10	8	15	13	11	9	7	9	8	7	6	5	10	9	8	7	6	10	9	8	7	6	10	9	8	7	6
CM / QA	5	4	4	4	3	4	3	3	3	2	8	7	7	7	6	10	9	9	9	8	8	7	7	7	6	8	7	7	7	6
Manuales	7	7	6	5	5	9	9	8	7	7	7	7	6	5	5	9	9	6	7	7	8	8	7	6	6	12	12	11	11	11

Tabla 2.4 Distribución de Actividades del proyecto por fases: Modo Semi-Capturado.

ACTIVIDAD	Plan y Requerimientos		Diseño de Producto		Programación		Integración y Prueba		Mantenimiento	
	%	FSP	%	FSP	%	FSP	%	FSP	%	FSP
Análisis de requerimientos	45	3.6	10	1.8	3	1.4	2	0.8	5.5	0.5
Diseño de productos	14.5	1.2	42	7.5	6	2.9	4	1.5	11	1
Programación	7	0.6	12.5	2.2	55	26.3	42	15.9	39.5	3.4
Plan de prueba	4.5	0.4	6.5	1.2	6.5	3.1	4	1.5	4.5	0.4
Verificación y Validación	8.5	0.7	8.5	1.5	10.5	5	24	9.1	14	1.2
Oficina de proyectos	11	0.9	10	1.8	6.5	3.1	7.5	2.8	7.5	0.6
CM / QA	4	0.3	3	0.5	7	3.4	9	3.4	7	0.6
Manuales	5.5	0.4	7.5	1.4	5.5	2.6	7.5	2.8	11	1
TOTAL	100	8.1	100	17.9	100	47.8	100	37.8	100	6.7

FASE	Plan y Requerimientos	Diseño de Producto	Programación	Integración y Prueba	Desarrollo	Mantenimiento
Tamaño del producto	S I M L	S I M L	S I M L	S I M L	S I M L	S I M L
Porcentaje de las Fases	6	16	68 65 62 59	16 19 22 25		
Porcentaje de Actividades						
Análisis de requerimientos	46	15	5	3	6	7
Diseño de productos	20	40	10	6	14	13
Programación	3	14	58	34	48 47 46 45	45 44 43 42
Plan de prueba	3	5	4	2	4	3
Verificación y Validación	6	6	6	34	10 11 12 13	10 11 12 13
Oficina de proyectos	15	11	6	7	7	7
CM/ QA	2	2	6	7	5	5
Manuales	5	7	5	7	6	10

BIBLIOGRAFIA

- BOEHM Barry N., Software Engineering Economics. Editorial Prentice Hall (United States of American, New Jersey 1981).
- BOVILLE D., Introducción a la Evaluación de Proyectos. Instituto de Economía Energética (S.F).
- BRAITHWAITE Kenmore S., Database Management & Control. Editorial McGrawHill (United States of American, NEW YORK 1990).
- CHARETTE R. N., Applications Strategies for Risk Analysis. Editorial McGraw Hill (United States of American, New York, 1990).
- CHORAFAS Dimitris, System Architecture & System Design. Editorial MacGrawHill (United States of American, New York, 1989).
- CHRISTIE Linda Gail - CHRISTIE John, Enciclopedia de Términos de Microcomputación, Editorial Prentice Hall (Mexico s.f.).

Universidad Don Bosco

KROEKEN David, Managament Information System. Editorial MacGrawHill (United States of American, California, 1989).

MATTHEWS Lawrence, Estimación de Costos de Producción. Editorial MacGrawHill.

SAMUELSON Paul A - NORDHAUS William. Microeconomics. Edotirial MacGrawHill (United States of American, 1989).

TAYLOR George A., Ingeniería Económica, Editorial Limusa (Mexico 1991).

WYATT A.L., Computer Professional Dictionary. Editorial McGrawHill (United States of American, California, 1990).