

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POSTGRADO



EL IMPACTO DE LA OMISION DE METRICAS DE  
CALIDAD EN EL DESARROLLO DE SOFTWARE

POR

ING. IRMA GUADALUPE UICAB LUNA

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS  
DE LA ADMINISTRACION CON ESPECIALIDAD  
EN SISTEMAS

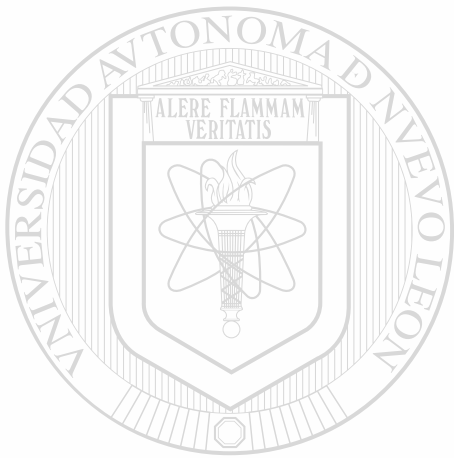
SAN NICOLAS DE LOS GARZA, N. L., JUNIO DEL 2001

2001

TM  
Z5853  
.M2  
FIME  
2001  
U4

EL IMPACTO DE LA OMISION DE METRICAS DE  
CALIDAD EN EL DESARROLLO DE SOFTWARE

IG  
U  
F



UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POSTGRADO



EL IMPACTO DE LA OMISION DE METRICAS DE  
CALIDAD EN EL DESARROLLO DE SOFTWARE

UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

POR

ING. IRMA GUADALUPE UICAB LUNA

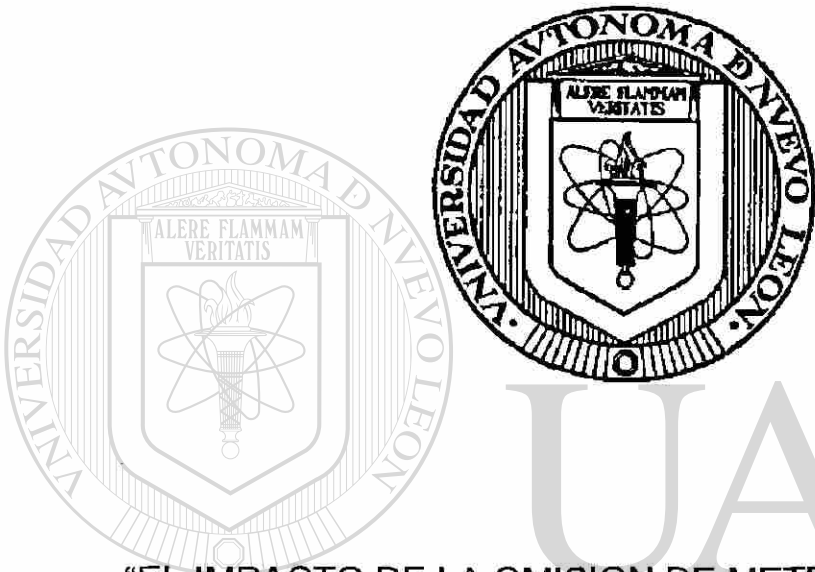
TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS  
DE LA ADMINISTRACION CON ESPECIALIDAD  
EN SISTEMAS

SAN NICOLAS DE LOS GARZA, N. L., JUNIO DEL 2001



UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POST-GRADO



“EL IMPACTO DE LA OMISION DE METRICAS DE CALIDAD EN  
EL DESARROLLO DE SOFTWARE”

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

POR  
DIRECCIÓN GENERAL DE BIBLIOTECAS

ING. IRMA GUADALUPE UICAB LUNA

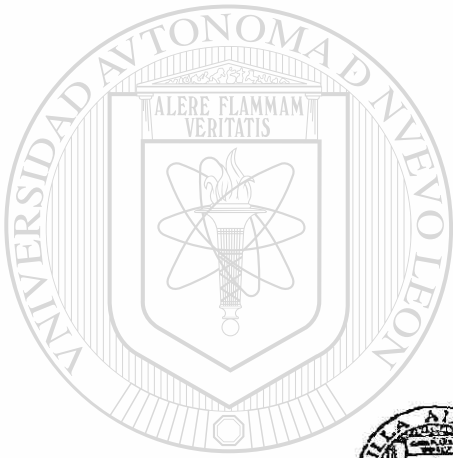
TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS DE LA  
ADMINISTRACION CON ESPECIALIDAD EN SISTEMAS

SAN NICOLAS DE LOS GARZA, N.L, JUNIO DEL 2001.

305990 U. Leon

TH  
Z5853  
•Ma  
FILE  
2001  
.U4



FONDO  
TESIS

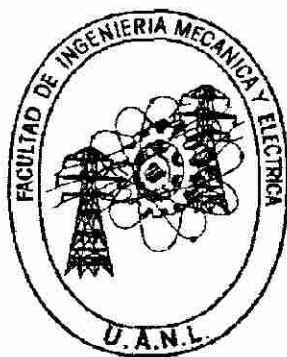
# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA  
DIVISION DE ESTUDIOS DE POST-GRADO



**“EL IMPACTO DE LA OMISION DE METRICAS DE CALIDAD EN  
EL DESARROLLO DE SOFTWARE”**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

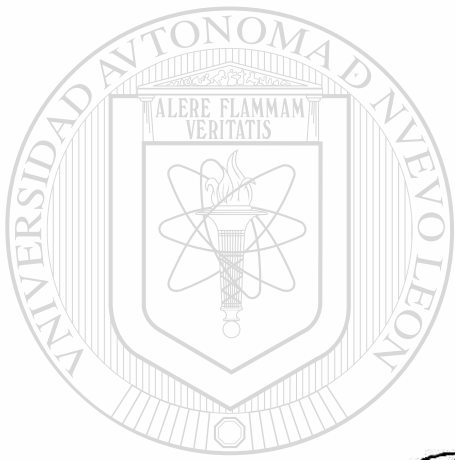
POR

DIRECCIÓN GENERAL DE BIBLIOTECAS  
ING. IRMA GUADALUPE UICAB LUNA

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS DE LA  
ADMINISTRACION CON ESPECIALIDAD EN SISTEMAS

SAN NICOLAS DE LOS GARZA, N.L, JUNIO DEL 2001.



# UANL



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

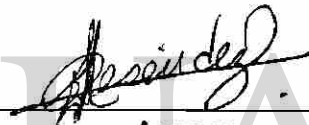
FONDO DE TESIS  
DIRECCIÓN GENERAL DE BIBLIOTECAS



**UNIVERSIDAD AUTONOMA DE NUEVO LEON**  
**FACULTAD DE INGENIERIA MECANICA Y ELECTRICA**  
**DIVISION DE ESTUDIOS DE POST-GRADO**

Los miembros del comité de tesis recomendamos que la tesis "El Impacto de la Omisión de Métricas de Calidad en el Desarrollo de Software", realizada por la alumna Ing. Irma Guadalupe Uicab Luna, matrícula 701340 sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Administración con especialidad en Sistemas.

El Comité de Tesis



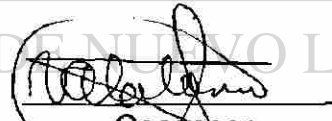
Asesor

M.C. Rosa María Reséndez Hinojosa



Coasesor

M.C. David Garza Garza



Coasesor

Dr. Victoriano F. Alatorre González



Vó.Bo.

M.C. Roberto Villarreal Garza  
División de Estudios de Post-grado

San Nicolás de los Garza, N.L., a Junio del 2001.

## DEDICATORIAS

**A DIOS NUESTRO SEÑOR.** Por ser la luz que guía e ilumina cada instante de mi vida; que me acompaña tanto en los momentos más difíciles como en los más placenteros; por que su infinito amor me dio la fortaleza espiritual, salud e inteligencia para poder llegar a este momento tan importante para mi.

**A MIS AMADISIMOS PADRES.** Profr. Gustavo Uicab Chan y Profra. Irma Luna Rubio, esos seres tan maravillosos que me han dado todo su amor incondicional ; cultivando en mi los valores que desde niña llevo en mi corazón y que han sido el principal motivo de mi superación personal. A ustedes con mi devoción de siempre.

**A MIS HERMANOS.** Gustavo Angel, Héctor Javier, Karina, María Guadalupe y Víctor Hugo; en especial a mi querida hermana Karina, por no dejarme caer nunca. Con todo mi amor y cariño.

**A MIS SOBRINAS.** Isamar y Paola, por la alegría y la ternura que me inspiran.

**AI ING. MYRNA A. MARTINEZ OBREGON.** Por el inmenso ejemplo de valor y de superación que me ha dado.



## **AGRADECIMIENTOS**

**AL INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY.** Por brindarme la oportunidad de realizar una investigación de alto nivel .

**A MI ASESOR Y COASESORES.** Por el tiempo que dedicaron a la elaboración y revisión del trabajo de tesis.

**A MIS ENTRAÑABLES AMIGOS.** Isaías, Rosa Elva, Mónica y Emilio , por que con su infinita paciencia, ánimos y consejos, han sido testigos y han contribuido en la terminación de la presente tesis; por sus constantes porras.

**A MIS MAESTROS Y COMPAÑEROS DE LA DIVISION DE POSTGRADO DE LA :**

**FACULTAD DE INGENIERIA MECANICA Y ELECTRICA DE LA U. A .N. L.**

## PROLOGO.

Desde que se inició la era de la computadora y el desarrollo de sistemas de información computacionales en las organizaciones, se ha buscado desarrollarlos en una forma rápida, sencilla y efectiva para que satisfagan de la manera mas adecuada los requerimientos establecidos.

Desde entonces se han desarrollado técnicas y metodologías que pretenden ayudar a producir sistemas de información más eficientes que sean fáciles de construir y de mantener. Pero desafortunadamente aún se tienen problemas con el diseño, desarrollo e implementación de software debido a la complejidad que conlleva este proceso.

La sociedad actual demanda sistemas que les proporcionen un alto nivel competitivo al menor costo posible esto es, calidad en los productos que adquieren que les permitan diseñar estrategias de negocio.

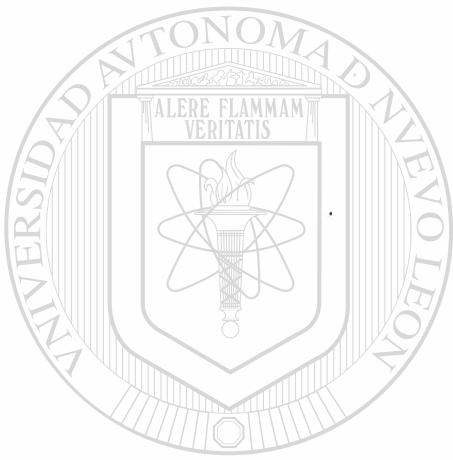
Es por esto que la presente tesis se enfoca en el impacto que tienen las actividades del SQA ("Software Quality Assurance") en el éxito o fracaso de un sistema de información y presenta mediante un caso práctico los costos que implica el no llevar a cabo el plan de actividades que SQA propone con el objetivo de proporcionar al cliente productos de calidad que cumplan con las metas antes señaladas.

# INDICE

Capitulo	Página
Sintesis.	1
1. INTRODUCCION.	
1.1. Planteamiento del Problema.	3
1.2. Objetivo de la tesis.	4
1.3. Definición de Hipótesis.	4
1.4. Limites de la Investigación.	4
1.5 Justificación del Trabajo.	4
1.6. Metodología .	5
1.7. Revisión Bibliográfica.	5
2. EL ASEGURAMIENTO EN LA CALIDAD DEL SOFTWARE . (ANTECEDENTES)	
2.1. Descripción General de la Calidad.	7
2.2. Aseguramiento de la Calidad.	8
2.3. Calidad en los Sistemas de Información.	9
2.4. El Objetivo del SQA.	11
2.5. El Software, el SQA y el Éxito de los Sistemas.	13
2.5.1. SQA en Software de Misión Crítica.	14
2.5.2. SQA en Sistemas de Tiempo Real.	14
2.5.3. SQA en Sistemas Interactivos.	15
2.5.4. SQA en Software de Negocios.	15
2.6. Contenido del Plan de Calidad.	16
2.6.1. Establecimiento de Requerimientos.	17
2.6.2. Establecer la Metodología.	17
2.6.3. Evaluar la Calidad del Producto.	17
2.7. Factores y Métricas de Software.	18
2.7.1. Factores de Calidad del Software.	19
2.7.2. Métricas de Calidad del Software.	20
2.7.3. Factores de Calidad según Roger Pressman.	21
2.7.4. Métricas de Calidad según Roger Pressman.	23
2.7.5. Factores de Calidad de McCall.	28
2.7.6. Métricas de Calidad de McCall.	30
2.7.7. Factores de Calidad de Lowell Jay Arthur.	33
2.7.8. Factores de Calidad según Peter Freeman.	34
2.7.9. Factores que afectan la Calidad y Productividad de los Sistemas.	36
2.8. El Proceso de Pruebas.	39
2.8.1. Objetivo del Proceso de Pruebas.	40
2.8.2. Diseño de Casos de Pruebas .	41
2.8.3. Prueba de Funcionalidad del Sistema.	42
2.8.4. Prueba de Confiabilidad y Seguridad en la	42

Información.	43
2.8.5. Prueba de Validación.	43
<b>3. EL COSTO DE LA CALIDAD Y SU IMPACTO EN LA SATISFACCION DEL USUARIO.</b>	
3.1. Costo de la Calidad en Sistemas de Información.	44
3.2. Costos de la Etapa de Desarrollo del Software.	45
3.3. Costos de Mantenimiento del Software que no utiliza Métricas SQA.	47
3.4. Costos de la Calidad del Software.	52
3.5. Costos de los Recursos y Tecnología Empleados.	53
3.6. Nivel de Satisfacción del Usuario.	54
<b>4. METODOLOGIA.</b>	
4.1. Introducción.	56
4.2. Metas de Medición.	56
4.3. Modelo Propuesto.	57
4.4. Tamaño de la Muestra.	59
4.5. Variables.	59
<b>5. ANALISIS DE LOS DATOS, CONCLUSIONES Y RECOMENDACIONES.</b>	
5.1. Historia de la Empresa.	60
5.2. Organización del Departamento de Informática.	61
5.3. Tecnología de Información.	62
5.4. ABAPRO.	65
5.5. Análisis de las Etapas de Desarrollo del Sistema ABAPRO.	67
5.5.1. Costos del Análisis, Diseño y Desarrollo del ABAPRO.	67
5.5.2. Empleo de los Factores de Calidad de Freeman para La Evaluación del ABAPRO.	69
5.5.3. Empleo de los Factores de Calidad de Lowell Jay Arthur para la Evaluación del ABAPRO.	73
5.5.4. Costos de la Etapa de Implementación del ABAPRO.	77
5.5.5. Costos del Mantenimiento del Sistema ABAPRO.	79
5.5.6. Cuadro Comparativo de Costos.	80
5.6. Estadística Descriptiva.	82
5.6.1. Distribución del Negocio.	82
5.6.2. Distribución por el Tipo de Servicio.	83
5.6.3. Distribución por Enfoque Informático.	86
5.6.4. Análisis de los Mantenimientos a Programas.	88
5.6.5. Distribución de la Criticalidad de las Solicitudes.	89
5.6.6. Calificación de las Solicitudes de Servicio.	89
5.7. Conclusiones Generales.	90
5.8. Trabajos Futuros.	91
 Bibliografía.	 92

Lista de Figuras.	94
Lista de Tablas.	95
Lista de Abreviaturas.	96
Anexos.	97
Resumen Autobiográfico.	116



# UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

## SINTESIS.

El Aseguramiento de la Calidad del Software es un área que suele ser desatendida dentro del desarrollo de software; la presión en los tiempos de entrega o la falta de conocimiento son las principales razones por las que las casas constructoras de software le restan importancia a este punto tan por demás crítico.

Es por eso que en la presente investigación se presenta el plan de actividades SQA para garantizar la calidad de los sistemas que se producen, que básicamente consta del establecimiento de requerimientos ,desarrollo de procedimientos para mantener el software y la implementación de revisiones y auditorías para evaluar la calidad del mismo.

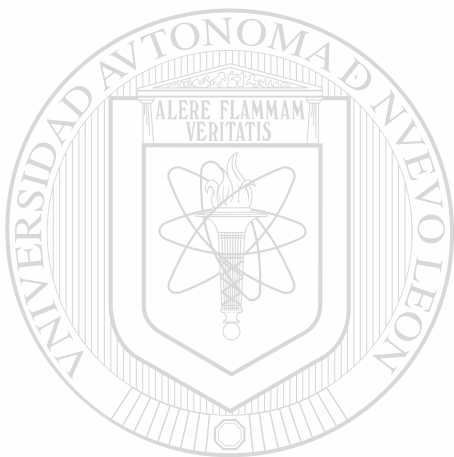
Además se ilustran los Factores de Calidad propuestos por Roger Pressman, McCall, Lowell Jay Arthur y Peter Freeman para determinar el nivel de calidad de un sistema; mismos que sirven para evaluar el grado de calidad del sistema ABAPRO (Abastecimientos y Proveedores) .

Mediante una comparación de los costos en cada una de las etapas de construcción del sistema antes mencionado se confirma el impacto monetario que se tiene al darle mantenimiento a programas que no cumplen con las funciones para los cuales fueron creados.

También se presenta el análisis del impacto de esta omisión de métricas de calidad en el nivel de satisfacción del usuario en la etapa de liberación del sistema ABAPRO .



El objetivo principal de esta investigación es presentar los problemas que suelen presentarse en las etapas de implementación y de operación de los sistemas en cuyo análisis y diseño no se aplicaron técnicas que garantizarán el nivel de calidad del sistema.



# UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

# CAPITULO 1

## INTRODUCCION

### 1.1. Planteamiento del Problema.

El desarrollo de software es un proceso complejo. Este proceso involucra diversas variables y se ve afectado por muchas situaciones, tales como una mala administración del proyecto, rotación del personal involucrado, obsesión por recortar recursos humanos, económicos y tiempos establecidos, etc.

---

En la actualidad existen empresas mexicanas que en su afán de “minimizar costos” se olvidan de un punto crítico, el Aseguramiento de la Calidad del Software; y esto repercute tarde o temprano una vez que se ha implementado el Sistema ya que los costos de mantenimiento a programas que no se probaron o que no cumplen en su totalidad con los procesos del negocio inflan el presupuesto inicial e incluso lo duplican a tal grado que en ocasiones hubiese sido más viable adquirir alguno de los productos del mercado que desarrollar un sistema propio.

## **1.2 Objetivo de la Tesis.**

El objetivo primordial de la tesis es presentar mediante un caso práctico (a realizar en la empresa Grupo Cydsa, S.A. de C.V.); los altos costos económicos que resultan de la implementación de sistemas sin calidad (en los cuales no se tuvo el cuidado de obtener el visto bueno del usuario y para los cuales no se hicieron las pruebas pertinentes) para posteriormente reflejar su impacto en la satisfacción de los usuarios.

## **1.3 Definición de Hipótesis.**

Se pretende comprobar que existe una relación directa entre la omisión de métricas de calidad en sistemas al momento de su elaboración y los costos originados así como el pésimo nivel de funcionalidad de los mismos.

## **1.4 Límites de la Investigación.**

La presente investigación se enfocará básicamente en realizar un estudio exploratorio de los costos ocasionados por la falta de calidad en los sistemas propios en una empresa de giro productivo en la Ciudad de Monterrey (Grupo Cydsa, S.A. de C.V.) así como la aplicación de métodos estadísticos para medir el impacto de estos costos contra el grado de satisfacción de los usuarios de dichos sistemas.

## **1.5 Justificación del Trabajo.**

La principal razón por la que pretendo realizar este tema de tesis es con el fin de aportar la experiencia personal que obtuve al integrarme a la etapa de implementación del sistema ABAPRO(Abastecimientos y Proveedores), en la cual descubrí que no se aplicaron las técnicas SQA(Software Quality

Assurance) en el análisis y diseño del mismo, por lo que a un año y medio de su implementación en los negocios de la compañía se siguen generando costos muy elevados en su mantenimiento debido a que las aplicaciones no cumplen con los requerimientos de información de los usuarios e incluso se está evaluando la posibilidad de adquirir otro software ya que el actual le resta competitividad a la empresa por la falta de datos confiables en sus estados financieros. Además de que investigando con algunos colegas me he percatado de que este problema se ha presentado en otras grandes empresas del Estado.

### **1.6 Metodología .**

- Recopilar información de libros y tesis relacionadas con el tema.
  - Recopilar documentos que reflejen los costos originados tanto en la implementación como en la etapa post-implementación de los sistemas institucionales de la empresa Grupo Cydsa , S.A. de C.V.
  - Investigar y presentar los métodos actuales con los que se mide el nivel de funcionalidad del sistema ABAPRO.
- 
- Emplear métodos estadísticos para medir el impacto de los costos y establecer una relación entre estos y el nivel de insatisfacción del usuario por falta de calidad en los sistemas institucionales.
  - Obtener conclusiones .
  - Proponer trabajos futuros sobre la base de la experiencia obtenida.

### **1.7 Revisión Bibliográfica.**

Nancy Uvalle [Uvalle,1991] en su trabajo de tesis titulado “Control de Calidad en Sistemas de Información” , desarrolla una metodología que ayuda a diseñar y desarrollar sistemas de información con calidad que no requieren mantenimientos costosos y que cumplen con los requerimientos de usuario.

En el trabajo de tesis presentado por Brenda Villarreal [Villarreal,1995] titulado "Aseguramiento de la Calidad de Sistemas de Información" se propone un documento que se utiliza como guía a los usuarios para evaluar la calidad en los sistemas de información y se llega a la conclusión de que la calidad de un sistema se determina principalmente por la perspectiva del usuario.

Además sugiere como trabajo futuro un estudio sobre el efecto del costo de calidad en productos que son modificados por no satisfacer las necesidades para las cuales fueron creados; tema base de la presente tesis.

Por su parte Eduardo Gómez [Gómez,1995] en su tesis cuyo tema principal "Métricas de Calidad y Productividad en la Codificación de Software" ; concluye que las métricas de software son herramientas importantes para las empresas nacionales para no dejar en entre dicho la falta de control en los departamentos de sistemas de información de dichas empresas.

Un último trabajo de tesis, presentado por Asley Cristales [Cristales,1996] titulado "Estudio Exploratorio de la Aplicación de Actividades SQA en el Proceso de desarrollo de Software" se enfoca a establecer la relación que existe entre la aplicación de actividades y técnicas SQA en el proceso de desarrollo de sistemas como un posible factor determinante del éxito de los sistemas y propone como futuro tema a desarrollar un estudio que proponga métricas de aseguramiento de calidad en software aplicables de manera sencilla en la práctica real .

# CAPITULO 2

## EL ASEGURAMIENTO EN LA CALIDAD DEL SOFTWARE (ANTECEDENTES)

### 2.1 Descripción General de la Calidad.

A continuación se presentan algunas definiciones de calidad propuestas por los grandes pioneros de este campo :

---

Todas las instituciones humanas (compañías industriales, escuelas, hospitales, iglesias, etc.) existen para proporcionar productos o servicios a la existencia humana. Un aspecto esencial de estos productos o servicios es que están listos para usarse. Esta frase "listos para usarse" es el principio básico de la palabra "Calidad" [Juran, 1970].

La Calidad es la totalidad de las peculiaridades y características de un producto o servicio que determinan su capacidad de satisfacer necesidades declaradas ó implícitas [Stebbing, 1991].



Calidad es cualquier característica que hace de un objeto mejor o peor, recomendable o reprobable; es el grado de excelencia que posee alguna cosa, superioridad, excelencia [Diccionario Webster's Unabridged].

La Calidad nunca es accidental, siempre es el resultado de un esfuerzo inteligente [John Ruskin].

La Calidad es una entidad alcanzable, medible y rentable [Crosby,1991].

Habría que agregar a esta última definición que aunque la calidad es una entidad alcanzable y medible el que también es relativa. Ya que el grado de calidad de una persona puede parecer falta de calidad para otra.

## 2.2 Aseguramiento de la Calidad.

¿Qué significa Asegurar la Calidad? "hacer que la gente haga mejor todas las cosas importantes que de cualquier forma tiene que hacer".

El gran compositor alemán Richard Wagner (1813-1883) al preparar sus dramas musicales(óperas) era muy cuidadoso para reunir todas las relaciones entre las diferentes formas del arte utilizadas en la presentación de un drama musical. Al hacerlo y antes de escribirlo; le dedicaba tanta atención a las palabras, al escenario, al vestuario y la presentación general como a la música, con la intención de crear una imagen sonora completa.

Reunir todas las actividades y funciones en forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático, si lo llevamos al escenario industrial, en la actualidad a esta filosofía podría llamársele Aseguramiento de la Calidad.

Para el Aseguramiento de la Calidad, se necesita la integración y el completo control de todos los elementos dentro de un área específica de operación para que ninguno este subordinado a otro. Otro punto crítico es en primer lugar, asegurar que se conozcan todas las necesidades para la presentación total. En otras palabras, las necesidades del cliente deben presentarse bien detalladas para que el proveedor las comprenda por completo y así no existan áreas dudosas en cuanto a las necesidades de servicios.

### **2.3 Calidad en los Sistemas de Información.**

La presión del mundo de los negocios por lograr mayor eficiencia, costos más bajos y aumentos de productividad, suele recaer sobre la tecnología de información. Los enfoques tradicionales de tecnología de información han fracasado y siguen fracasando y sólo las compañías que han comprendido el reto de la calidad han prosperado, han sobrevivido a recesiones y han aniquilado a la competencia.

La sociedad actual exige calidad en todos los productos y servicios que adquiere. El tema de calidad en sistemas de información(SI) se ha desarrollado en el ambiente de tecnología, transformándose en un factor clave para la selección de los sistemas de información.

De manera general, se puede considerar que un SI es de alta calidad si :

1. Hace lo que el usuario desea que haga.
2. Utiliza los recursos computacionales correcta y efectivamente.
3. Es fácil de aprender y de usar para el usuario.
4. El desarrollador puede diseñar, codificar, probar y mantener el sistema con facilidad relativa.

Pero existe cierta relatividad en el concepto de Calidad en los Sistemas de Información, ya que como se comentó en el punto 2.1, lo que para una persona representa un alto nivel de calidad, para otra no lo es tanto, es decir : para cada persona el mismo producto generalmente tendrá diferente "calidad" . Pero si la calidad es diferente para cada persona, ¿quién o quiénes determinan lo que realmente significa este término?. Considérense las siguientes ideas, escasas pero conflictivas acerca de lo que constituye calidad en el software:

- Cero defectos representa calidad
  - a. Para los usuarios a quienes su trabajo se ve afectado por dichos defectos.
  - b. Para los gerentes que son criticados por esos defectos.
- Tener muchas características representa mucha calidad.
  - a. Para los usuarios quienes su trabajo saca provecho de estas características.
  - b. Para el personal de mercadotecnia, que cree que entre más características , más venden productos.
- Código elegante es índice de calidad.
  - a. Para los desarrolladores que dan preferencia a este punto.
  - b. Para los profesores de computación que disfrutan de lo elegante.
- Un alto grado de ejecución es calidad.
  - a. Para los usuarios que le dan mucho valor a la capacidad de sus máquinas.
  - b. Para los vendedores que tienen que someter sus productos a pruebas de ejecución de otros productos.
- Bajos costos de desarrollo representan un alto nivel de calidad.
  - a. Para clientes que desean comprar miles de copias del software.

- b. Para los administradores de proyectos que tienen presupuestos recortados.
- Rápido desarrollo es alta calidad.
  - a. Para los usuarios cuyo trabajo es esperar el tiempo de respuesta del software.
  - b. Para mercadotecnia que desea ganar mercado antes que los competidores lo hagan.
- Una Interfase amigable es alta calidad.
  - a. Para los usuarios que pasan ocho horas al día enfrente de una computadora usando software.
  - b. Para usuarios que no pueden recordar detalles de la pantalla entre un campo y otro.

Por lo que se puede concluir que la calidad en el software la determina el usuario final, quien de acuerdo a sus necesidades evalúa si un sistema cumple con sus expectativas o no.

## 2.4 El Objetivo del SQA.

La calidad en el software es el grado en el que un producto de software satisface los requerimientos y presenta buenas condiciones de entrega, la probabilidad de que el software se ejecute durante un periodo particular sin fallas .

“Software Quality Assurance” (SQA) es una actividad que tiene como objetivo asegurar la calidad del software o de los sistemas de información que se producen, siendo esa calidad una adecuación a las especificaciones de los usuarios.

Donald Reifer proporciona su propia definición : "El Aseguramiento en la Calidad del Software es el sistema de métodos y procedimientos usados para asegurar que el producto del software satisfaga los requerimientos. Este sistema involucra planeación, medición y monitoreo de actividades de desarrollo ejecutados por otros" [Schulmeyer,19 87].

Pressman por su parte, opina: "la calidad del software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo detalladamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente" [Pressman ,1992].

Para la definición de Pressman se deben enfatizar tres puntos importantes:

- Los requerimientos del software son los fundamentos desde los que se mide la calidad. La falta de concordancia con los requerimientos es una falta de calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre se dará una falta de calidad.
- Existe un conjunto de requerimientos implícitos que a menudo no se mencionan (por ejemplo: el deseo de un buen mantenimiento).

En diversas organizaciones se han adoptado ya las ideas de SQA y se han establecido grupos o equipos de trabajo dedicados a la administración e implementación de programas de SQA. Y es que en el desarrollo de software la responsabilidad de la garantía de la calidad corresponde a muchas partes: ingenieros de software, analistas, clientes, y personas del grupo SQA.

## 2.5 El Software, el SQA y el Éxito de los Sistemas.

El software se puede definir como una colección de programas escritos para coordinar la operación de todo el circuito computacional, ayudando a la computadora a ejecutarse eficientemente. Un sistema de software es el sistema operativo que esta compuesto de programas de control y programas de proceso.

Los programas de control controlan la eficiencia y efectividad de la computadora en la cual residen.

Los programas de proceso controlan el proceso eficiente de preparación para el programador.

El desarrollo de este tipo de software es una actividad común dentro de la industria manufacturera de computadoras tales como IBM, Digital Equipment Corp, Burroughs, etc. Existe una similitud entre estas compañías y los desarrolladores de software que no pertenecen a este tipo de industrias; en donde surge la siguiente pregunta ¿cuál es la principal área para concentrar la calidad del software?

Tomando como base la experiencia de cientos ( o miles) de clientes que han recibido software, el mayor énfasis debe radicar en mantener todo bajo control. Dado que los clientes deben entender como trabaja el software para ganar valor en sus inversiones, el siguiente punto crucial es la documentación .

Asimismo, el uso de estándares, revisiones ,auditorías y pruebas son áreas importantes para garantizar la seguridad y la precisión del software.



Es importante entender como interactúa Software Quality Assurance (SQA) con los diferentes tipos de software. A continuación se verá como el SQA se relaciona con las aplicaciones en sistemas de misión crítica, sistemas en tiempo real, software interactivo y aplicaciones de negocios.

### **2.5.1 SQA en Software de Misión Crítica.**

El Software de Misión Crítica es aquel que se ejecuta en “recursos de misión crítica”, estos incluyen computadoras y servicios para misiones militares del Departamento de la Defensa relacionados a actividades de inteligencia, actividades de criptología relacionadas con la seguridad nacional, comandos y control de fuerzas militares y equipos integrales en sistemas de armas .

Este software debe ser confiable, por lo que la seguridad debe ser la principal característica de estos sistemas. El tipo de aseguramiento que se necesita es que los niveles de requerimientos del sistema, deberes y seguridad se cumplan.

---

### **2.5.2 SQA en Sistemas de Tiempo Real.**

Un Sistema en Tiempo Real es aquel que proporciona servicios o controla un proceso físico que va y viene. Este software está sujeto a obligaciones y a un nivel de desarrollo muy estricto, porque debe interactuar con otras entidades externas (otros programas de computadora, terminales, subsistemas, etc.), además es muy complejo y puede existir en diferentes formas tales como firmware, lenguaje ensamblador, microcódigo y en versiones recompiladas de desarrollos de alto nivel.

En este tipo de sistemas, el personal SQA debe asegurar la eficiencia.

Una opción sería checando la velocidad mediante tiempos de las computadoras , dado que los sistemas en tiempo real están conformados de muchas piezas, el siguiente punto es poner énfasis en la administración de la configuración . Las inspecciones, auditorías, y la documentación no deben ser subestimadas ya que sin ellas el producto final podría ser incompleto o incorrecto.

### **2.5.3 SQA en Sistemas Interactivos.**

Los Sistemas Interactivos son aquellos sistemas que requieren una interfase (interacción) con un operador humano en tiempo real o en tiempo no real. ¿Cuál es el rol del personal SQA en estos sistemas? asegurarse que el software a ser usado en la PC es compatible, hace lo que dice que hace, está lo suficientemente documentado como para que el usuario pueda aprender y manejar el sistema por si solo.

### **2.5.4 SQA en Software de Negocios.**

La aplicación mas ejemplar de este tipo de sistemas de negocios es el ambiente que usualmente se denomina procesamiento de datos electrónicos (EDP). El rol de la persona SQA en este tipo de software es desarrollar una auditoría del sistema EDP para determinar los procedimientos adecuados , la seguridad del sistema, la documentación adecuada y si el sistema está funcionando correctamente.

La importancia de hacer el trabajo bien hecho durante el desarrollo de estos tipos de sistemas se hace extremadamente obvia cuando las cosas van mal una vez que se han puesto en producción o en las fases de operación y mantenimiento. El éxito de un sistema (de cualquier tipo de los antes señalados) radica fundamentalmente en que cumpla con los requerimientos y

expectativas para los que fueron creados y que puedan proporcionar al negocio el nivel de competitividad que se necesita para sobrevivir en el ambiente de negocios que persiste en la actualidad. Y es aquí en donde el SQA se encarga de que este éxito sea el resultado de sus revisiones, auditorías y documentación de procesos del sistema.

## 2.6 Contenido del Plan de Calidad .

El Programa de Calidad de Software se implementa para determinar el nivel de calidad actual del producto de software. Puede ser definido como el conjunto de actividades que se llevan a cabo por el administrador del programa para :

- Establecer los requerimientos para la calidad de un producto de software.
- Establecer y ejecutar procedimientos para desarrollar y mantener el software.
- Establecer e implementar procedimientos para evaluar la calidad del producto de software y para evaluar la documentación asociada, procesos y actividades que impactan la calidad del producto .

## DIRECCIÓN GENERAL DE BIBLIOTECAS

El fundamento principal del programa SQA es el concepto de que un producto de calidad significa en efecto, que el producto cumpla con los propósitos para los que fué diseñado. Como resultado se tiene que los propios requerimientos del producto son en realidad los requerimientos de calidad: software funcional, mantenible , interoperable , que se pueda probar , etc.

### **2.6.1 Establecimiento de Requerimientos.**

Este punto se refiere a establecer los componentes o una base de requerimientos para que la calidad del software se lleve a cabo en el proyecto.

Este esfuerzo también incluye el control de los cambios a estos requerimientos después de que han sido establecidos. Cuando un cambio al producto de software es hecho, los mismos requerimientos de calidad deberán ser aplicados, a menos que estos requerimientos hayan sido cambiados también. El proceso de establecer y controlar estos cambios es conocido como "proceso de administración de configuración".

### **2.6.2 Establecer la Metodología.**

El segundo elemento del programa SQA pertenece al establecimiento e implementación de la metodología a seguir en las actividades del desarrollo y soporte del producto. El término "desarrollo" se usa para explicar la definición, diseño, codificación, pruebas del proyecto y su respectiva documentación.

---

Estas actividades de soporte son aquellas que se llevarán a cabo una vez que el software se ha probado y se ha puesto en producción. Si los procedimientos adecuados no han sido establecidos para manejar los cambios al software, la calidad inicial del producto puede ser degradada .

### **2.6.3 Evaluar la Calidad del Producto .**

Este elemento del programa se refiere a cómo la calidad del producto de software es medida y valorada(SQAM). SQAM es un conjunto de actividades de valoración y medición desarrolladas a través del ciclo de desarrollo para evaluar la calidad del producto y sus respectivos procesos, documentación y actividades que impactan la calidad.

Todas estas actividades tienen que ver con las acciones, revisiones, evaluaciones, pruebas, análisis, inspecciones y demás que fueron desarrolladas para determinar que los requerimientos técnicos establecidos han sido satisfechos en su totalidad. Las ventajas de aplicar el programa SQA se pueden resumir en los siguientes puntos:

- Es efectivo en cuanto a costos (elimina los costos producidos por mantenimientos a sistemas que no cumplen con sus objetivos).
- Permite llevar un control de las modificaciones y mantenimientos.
- Asegura la satisfacción del usuario del producto.
- Agrega un valor agregado a las aplicaciones del negocio.

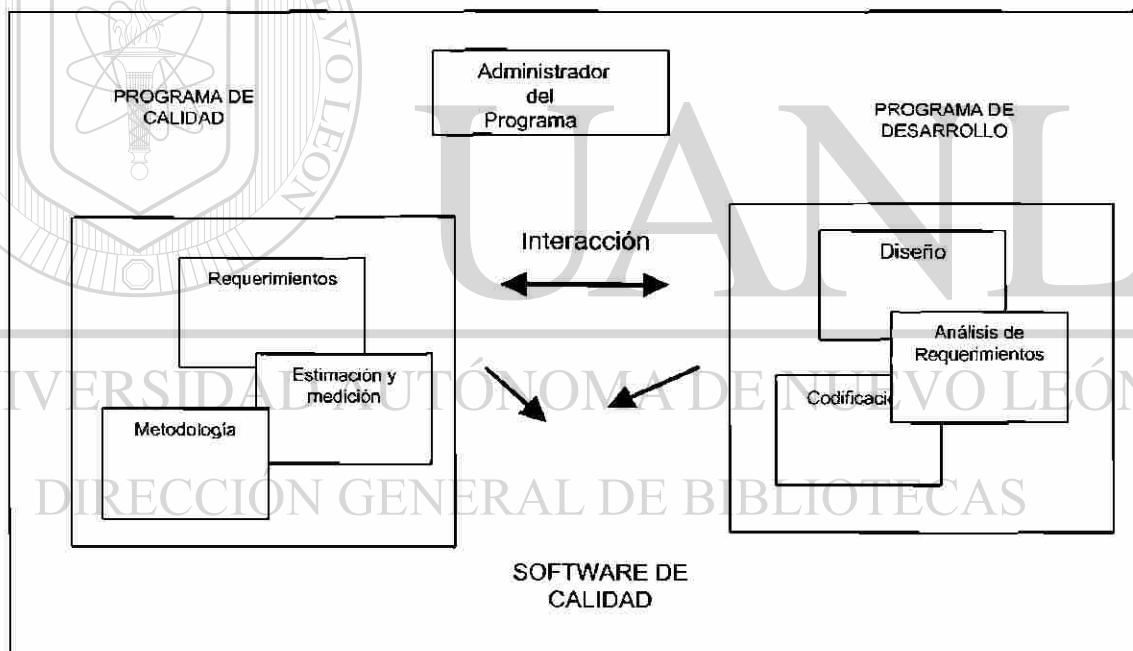


Figura 2-1 Generación de Software de Calidad

## 2.7. Factores y Métricas de Software.

Cuando se pretende expresar la calidad de un producto o servicio se debe contar con un método de medición exacto. Sin embargo, hay

características que no se pueden evaluar, sobre todo si se desea evaluar la satisfacción del usuario respecto a un producto de software.

A continuación se presentan los elementos o factores que contribuyen a obtener o determinar la calidad en el software de acuerdo a la perspectiva de diversos autores. Asimismo, con la lista de factores seleccionados, se presentan diferentes métricas o formas de evaluar estos elementos.

### **2.7.1 Factores de Calidad del Software.**

Entre los modelos de calidad de software existen aquellos que señalan que ésta se refiere a satisfacer los requerimientos del usuario para cumplir con atributos de los sistemas tales como funcionalidad, desempeño, facilidad de mantenimiento y falta de tolerancia [Ralya,1994]; por lo que se ha sugerido establecer un conjunto de factores que la caractericen.

Sin embargo, es difícil desarrollar una serie de factores de calidad del software. Existen diversas propuestas con respecto a la consideración y definición de atributos y factores de calidad.

Los modelos de calidad que se han desarrollado consideran diferentes puntos de vista de sus autores, algunos indican que el software manifiesta verdaderamente ciertos atributos y otros indican que es muy difícil llegar a establecer algunos que sean válidos, dada la naturaleza misma del software.

Lo que debe reconocerse en cada intento de construir un modelo de calidad, dice Dromey, es que el software no manifieste directamente atributos de calidad. En su lugar, debe presentar características de los productos que implican o contribuyen a los atributos de calidad y a otras características (como

los defectos de los productos), que disminuyen a partir de los atributos de calidad de un producto [Dromey,1995].

Por su parte Terno ven afirma que los factores de calidad pueden ser eficiencia, integridad, confiabilidad, facilidad de uso, grado de corrección, facilidad de mantenimiento, verificabilidad, expandibilidad, flexibilidad, interoperabilidad, portabilidad y reusabilidad, por ejemplo , y los factores son refinados con ciertos criterios de calidad, y debido a las características multifuncionales de calidad se deben considerar las interrelaciones entre los factores de calidad [Terno ven,1992].

### **2.7.2 Métricas de Calidad del Software.**

“Cuando puedes medir lo que estas hablando y expresarlo en números, sabes algo de eso, pero cuando no puedes medirlo, cuando no puedes expresarlo mediante números, tu conocimiento es insatisfactorio; puede ser el principio del conocimiento, pero te encuentras apenas en tus pensamientos, en el escenario de la ciencia”.

Lord Kevin [Pressman,1992].

La medición tiene un lugar importante en la ingeniería. Se mide el consumo de energía, el peso, las dimensiones físicas, la temperatura, el voltaje ,etc. Desafortunadamente, la medición esta lejos del mundo de la ingeniería de software. Ya que se tienen problemas en estar de acuerdo en qué se va a medir y a evaluar.

Una métrica en la ingeniería de software es cualquier medida, proceso o documentación del sistema. Ejemplos: medir el tamaño de un sistema en líneas de código, el número de fallas reportadas en un sistema entregado o el número

de personas por día requeridos para desarrollar una parte del sistema [Bvillarreal,1995].

¿Por qué es tan importante medir el proceso de ingeniería de software y el producto (software) que se produce?. El software es medible por muchas razones: (1) para determinar la calidad del producto; (2) para evaluar la productividad de la gente que desarrolla el producto; (3) para asegurar los beneficios (en términos de productividad y calidad) derivados de los métodos y herramientas de la nueva ingeniería de software; (4) para formar una base de estimación; (5) para justificar los requerimientos de nuevas herramientas o capacitaciones; si no se realiza una medición, no existe un camino real para determinar que tanto se está mejorando.

### 2.7.3 Factores de Calidad según Roger Pressman.

Roger Pressman indica los siguientes factores que afectan la calidad de un producto de software:

- **Grado de corrección:** Un programa debe operar correctamente o proporcionar valor a sus usuarios. La corrección es el grado en el cual el software desarrolla sus funciones. La medida más común de corrección son los defectos por KLOC(miles de líneas de código) donde un defecto es definido como una falta de conformidad con los requerimientos. Estos defectos son reportados por el usuario del programa después de que el programa ha sido liberado para su uso general y son contados a partir de un periodo de tiempo, casi siempre después de un año.
- **Mantenibilidad :** Es la facilidad con que un programa puede ser corregido si se encuentra un error, su adaptabilidad a los cambios de entorno extendido según los cambios en los requerimientos de los usuarios. No



existe una manera directa de medir la mantenibilidad, sin embargo, podemos hacer uso de las mediciones indirectas.

- **Integridad:** La integridad del software se ha incrementado desde que hicieron su aparición los hackers y los virus. Este atributo mide la habilidad del sistema para evitar los ataques intencionales o accidentales a su seguridad. Estos ataques pueden ser a los tres componentes del software: programas, datos y documentos . Para medir la integridad es necesario definir otros dos atributos: amenaza y seguridad . La amenaza es la probabilidad (que puede ser estimada o derivada de evidencias empíricas) de que un ataque de un tipo específico ocurra en un tiempo de tiempo dado. La seguridad es la probabilidad (que también puede ser estimada o derivada) de que un ataque sea repelido. La integridad de un sistema puede ser definida como :

$$\text{Integridad} = \sum_{i=1}^n [1 - \text{amenaza} \times (1 - \text{seguridad})]$$

donde la sumatoria de la amenaza y la seguridad dependen del número de ataques recibidos (van de 1 a n...).

- **Facilidad de uso:** Es un intento para cuantificar una "interfase de usuario amigable" y puede ser medida en términos de 4 características: (1) la habilidad física o intelectual necesaria para aprender el sistema , (2) el tiempo necesario para llegar a ser moderadamente eficiente en el uso del sistema, (3) el incremento en productividad, producto de ser usado por un usuario moderadamente eficiente , (4) una valoración subjetiva ( a veces obtenida mediante un cuestionario) de las actitudes de los usuarios hacia el sistema.

#### 2.7.4 Métricas de Calidad según Roger Pressman.

Pressman [Pressman,1992] indica que las mediciones pueden ser clasificadas de dos maneras:

- Las mediciones directas (por ejemplo el ancho de un tornillo).
- Las mediciones indirectas (por ejemplo la “calidad” producida por los tornillos, medida por la cantidad de rechazos).

Las mediciones directas del proceso de ingeniería de software incluyen el costo y el esfuerzo empleado. Estas mediciones en el producto son las líneas de código producidas, la velocidad de ejecución, tamaño de la memoria y los defectos reportados en un período de tiempo. Las mediciones indirectas del producto tienen que ver con la funcionalidad, calidad, complejidad, eficiencia, confiabilidad y mantenibilidad. También propone que las métricas se pueden categorizar como se muestra en la figura 2-2,

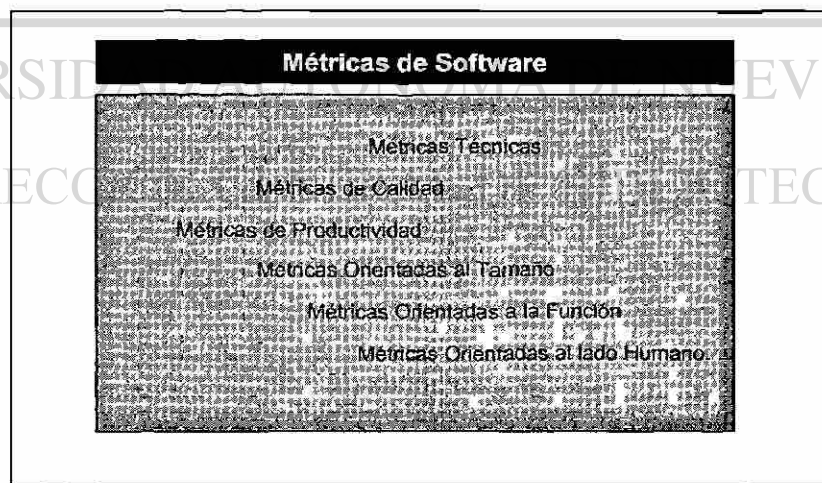


Figura 2-2 Clasificación de Métricas de Calidad según Roger Pressman

En la cual se puede observar que las métricas de productividad se concentran en las salidas del proceso de ingeniería de software, las métricas de calidad proveen un indicador de que tan cerca el software concuerda con los requerimientos de los clientes , y las métricas técnicas abarcan el carácter del software( complejidad lógica, grado de modulación ). En esta figura también se puede apreciar una segunda categorización , las métricas orientadas al tamaño, que son usadas para coleccionar métricas de la calidad y salidas de la ingeniería del software.Las métricas orientadas al funcionamiento, que proveen mediciones indirectas y las métricas orientadas al factor humano que recopilan información acerca de la manera en la cual la gente desarrolla software computacional y las percepciones humanas de la efectividad de las herramientas y métodos.

#### **Métricas Orientadas al Tamaño:**

Las Métricas Orientadas al Tamaño, son mediciones directas de software y del proceso mediante el cual es desarrollado. Si la organización del software mantiene registros; una tabla como la 2-1, puede ser creada. La tabla enlista cada proyecto de desarrollo de software. Así por ejemplo, para el proyecto aaa-01, el dato 12.1 KLOC(líneas por código) indica estas líneas de código fueron desarrolladas en 24 meses de trabajo a un costo de \$168,000.00 . Cabe señalar que el esfuerzo y el costo registrados en la tabla representan todas las actividades (análisis, diseño, codificación y pruebas), no solamente la codificación. En la misma tabla se puede apreciar que fueron desarrolladas 365 páginas de documentación y que 29 errores fueron encontrados en el siguiente año después de que fue liberado el proyecto , además, 3 personas trabajaron en el desarrollo del proyecto.

Proyecto	meses	\$	KLOC	pgs.doc	errores	personas
aaa-01	24	168	12.1	365	29	3
ccc-04	62	440	27.2	1,224	86	5
fff-03	43	314	20.2	1,050	64	6
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Tabla 2-1 Métricas Orientadas al Tamaño

Roger Pressman propone las siguientes métricas para medir en base al tamaño:

**Productividad = KLOC / meses de trabajo**

**Calidad = Defectos / KLOC**

**Costo = \$ / LOC**

**Documentación = páginas de documentación / KLOC**

Este tipo de métricas son controversiales y no son universalmente aceptadas como la mejor manera de medir el proceso de desarrollo del software. Por una parte, los defensores de estas métricas afirman que las líneas de código (LOC) son un artefacto de los proyectos de desarrollo de software que pueden ser fácilmente encontrado.

Por otra parte, los oponentes proclaman que las mediciones de líneas de código (LOC) son idóneas para analizar programas pequeños, pero que no se acomodan fácilmente a lenguajes no procedimentales y que su uso requiere un nivel de detalle que puede ser difícil de archivar.

### Métricas Orientadas a la Función:

Las Métricas Orientadas a la Función son mediciones indirectas del software y de su proceso de desarrollo. Están enfocadas a la “funcionalidad” o “utilidad” del programa, fueron propuestas por Albrecht [ALB,1979] quien propone una medida de productividad llamada “método de punto de función”. Los puntos de función (FPs) son derivados del uso de una relación empírica basada en mediciones contables del dominio de información y de las tasa de complejidad del software. Son cinco los dominios de información que determinan la función del software:

- Número de entradas de usuario.
- Número de salidas de usuario .
- Número de consultas de usuario.
- Número de archivos.
- Número de interfaces externas.

Los cuales se muestran de manera gráfica en la tabla 2-2.

Parámetro de Medición	Contador	Ponderación			
		Simples	Promedio	Complejo	
Número de Entradas de Usuario	<input type="text"/>	x 3	4	6	<input type="text"/>
Número de Salidas de Usuario	<input type="text"/>	x 4	5	7	<input type="text"/>
Número de consultas de Usuario	<input type="text"/>	x 3	4	6	<input type="text"/>
Número de Archivos	<input type="text"/>	x 7	10	15	<input type="text"/>
Número de Interfaces Externas	<input type="text"/>				<input type="text"/>
<b>CONTEO TOTAL</b>					<input type="text"/>

Tabla 2-2 Métricas de Puntos de Funcionalidad

Una vez que los datos han sido recopilados , un valor de complejidad es asociado a cada conteo. Para determinar la funcionalidad de cada punto, se usa la siguiente relación:

$$FP = \text{conteo-total} \times [0.65 + 0.01 \times \text{SUM}(Fi)]$$

Donde el conteo total es la suma de todos los FP's obtenidos en la tabla de la tabla 2-3 .  $F_i$  ( $i = 1$  a  $14$ ) son "valores de ajuste de complejidad" , basados en las respuestas de las preguntas de la tabla 2-3.

Tabla 2-3						
<b>Puntos de Funcionalidad Computacional</b>						
Tasa de cada factor en una escala del 0 al 5:						
0	1	2	3	4	5	6
No	Incidental	Moderado	Promedio	Significativo	Esencial	Influye
F:						
1. ¿Requiere el sistema un respaldo confiable y de recuperación?						
2. ¿Se requieren comunicación de datos?						
3. ¿Están distribuidas las funciones de procesamiento?						
4. ¿Es el performance crítico?						
5. ¿Se ejecutará el sistema en un ambiente operacional pesado?						
6. ¿El sistema requiere entrada de datos en línea?						
7. ¿Se Requiere construir múltiples pantallas para las entradas de datos en línea ?						
8. ¿Son actualizados en línea los archivos maestros?						
9. ¿Son complejas las entradas, salidas y consultas?						
10. ¿Es el sistema interno de procesamiento complejo?						
11. ¿El código es re usable?						
12. ¿Están incluidas las conversiones e instalaciones en el diseño?						
13. ¿Esta diseñado el sistema para múltiples instalaciones en diferentes organizaciones?						
14. ¿Esta diseñada la aplicación de manera que los cambios de los requerimientos del usuario puedan ser realizados fácilmente?						

Tabla 2-3 Puntos de Función Computacionales

Una vez calculados los puntos de fusión, se usan de manera analógica a las métricas de Líneas de Código para determinar atributos de productividad, calidad, etc:

**Productividad = FP / meses de trabajo**

**Calidad = Defectos / FP**

**Costo = \$ / FP**

**Documentación = páginas de documentación / FP**

Las métricas de funcionalidad fueron originalmente diseñadas para ser aplicadas en sistemas de aplicaciones de información sin embargo, los trabajos extendidos propuestos por Jones [JON,1986] llamadas puntos de características, adaptaron estas métricas a sistemas y aplicaciones de ingeniería de software. Estas métricas se utilizan en aplicaciones con alta complejidad algorítmica y en el control de procesos en tiempo real .

#### **Métricas Orientadas a la Calidad:**

La calidad puede ser medida a través de procesos de ingeniería y después de que el software ha sido liberado. Son las métricas derivadas antes de que el software sea liberado las que proporcionan las bases para diseñar y probar decisiones.

#### **2.7.5 Factores de Calidad de McCall.**

McCall fue uno de los primeros pioneros en realizar trabajos acerca de los atributos de calidad y muchos de los trabajos actuales se basan en sus investigaciones. El y sus colegas han propuesto una útil clasificación de los factores que afectan la calidad del software [Pressman,1992]. Estos factores de calidad de software, que aparecen en la figura 2-3 se centran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad de soportar los cambios y su adaptabilidad a nuevos entornos.

Para los factores indicados en la figura 2-3 , McCall proporciona las siguientes descripciones:

- **Corrección:** El grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el cliente.
- **Fiabilidad:** El grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida.
- **Eficiencia:** El conjunto de recursos computacionales y de código necesarios para que un programa desarrolle su función .
- **Integridad:** El grado en el que puede controlarse el acceso al software o a los datos por personal no autorizado.
- **Facilidad de uso:** El esfuerzo necesario para aprender un programa, trabajar con el, preparar su entrada e interpretar su salida.
- **Facilidad de mantenimiento:** El esfuerzo requerido para localizar y arreglar un error en un programa.
- **Flexibilidad:** El esfuerzo necesario para modificar un programa operativo.
- **Facilidad de prueba:** El esfuerzo requerido para probar un programa de forma que se asegure que realiza su función requerida.
- **Portabilidad :** El esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistemas de software a otro.
- **Reusabilidad:** El grado en que un programa ( o partes de un programa) se puede reusar en otras aplicaciones. Esto va relacionado con el empaquetamiento y el alcance de las funciones que realiza el programa.
- **Facilidad de interoperación:** El esfuerzo requerido para acoplar un sistema a otro.



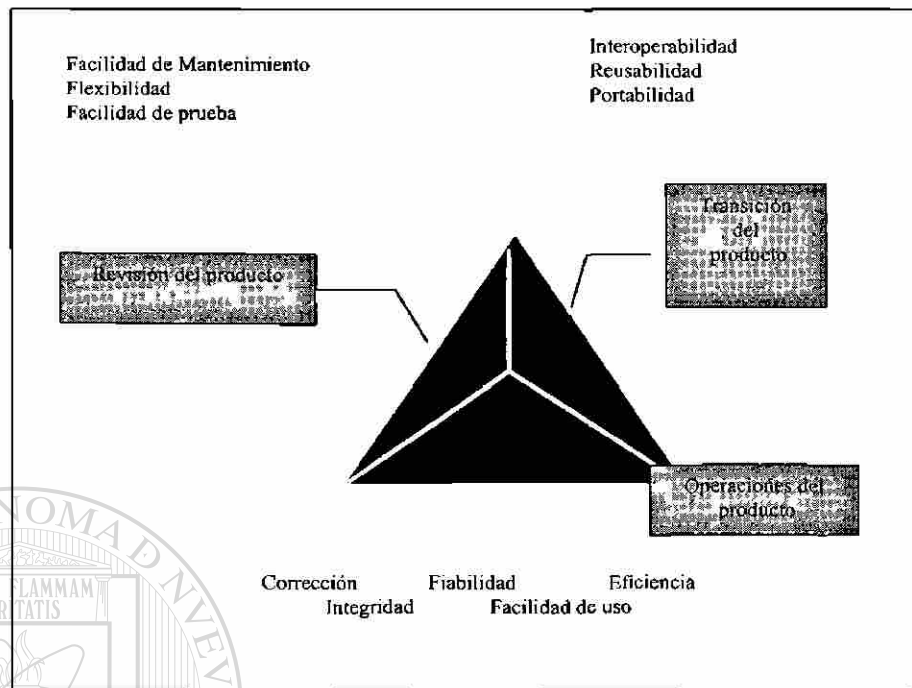


Figura 2-3 Factores de Calidad de Software de McCall

### 2.7.6 Métricas de Calidad de McCall.

Se menciona en [Pressman,1992] que es muy difícil y en algunos casos hasta imposible, desarrollar medidas directas para los factores anteriormente señalados. Por lo tanto, el autor define un conjunto de métricas para desarrollar expresiones de los factores de McCall de acuerdo a la siguiente relación:

$$F_c = C_1 \times M_1 + C_2 \times M_2 + \dots + C_n \times M_n.$$

Donde  $F_c$  es un factor de calidad de software,  $C_n$  son los coeficientes de regresión y  $M_n$  son las métricas que afectan al factor de calidad. Desafortunadamente, muchas de las métricas definidas por McCall solo pueden medirse de manera subjetiva. Estas métricas pueden aparecer en listas de

comprobaciones utilizadas para obtener un grado de los atributos específicos del software. El esquema de graduación propuesto por McCall va en escala de 0 a 10. En este esquema se utilizan las siguientes métricas :

- **Facilidad de auditoría:** La facilidad con que se puede comprobar el cumplimiento de los estándares.
- **Exactitud:** La precisión de los cálculos y el control (confiabilidad).
- **Normalización de las comunicaciones:** El grado en el que se utiliza la interfaz estándar, el protocolo y el ancho de banda.
- **Consistencia:** El uso de un diseño uniforme y de técnicas de documentación a lo largo del desarrollo del software.
- **Normalización de datos:** El uso de estructuras de datos y tipos de estándar en el desarrollo del software.
- **Tolerancia a errores:** El daño que se ocasiona cuando se genera un error en un programa.
- **Eficiencia en la ejecución :** El rendimiento en tiempo de ejecución de un programa.
- **Facilidad de expansión:** El grado en que se puede ampliar el diseño arquitectónico de datos o procedimientos.
- **Generalidad:** La capacidad de aplicación potencial de los componentes del programa.
- **Independencia del hardware:** El grado en el que el software es independiente del hardware en el que opera.
- **Instrumentación:** El grado en el que el programa monitorea su propia operación e identifica cuando ocurren los errores.
- **Grado de modulación:** La funcionalidad independiente de los componentes del programa.
- **Operabilidad:** La facilidad de operación de un programa.
- **Seguridad:** La disponibilidad de mecanismos que controlan o protegen los datos o el software.

- Documentación propia: El grado en el que se proporciona documentación del código fuente.
- Simplicidad: El grado en el que un programa puede ser interpretado sin dificultad.
- Independencia del software: El grado en el que un programa es independiente de lenguajes de programación con interfaces no estandarizadas.
- Facilidad de capacitación: El grado en el que el software asiste a los nuevos usuarios del sistema.

A continuación se presenta una tabla comparativa de la relación entre los factores y las métricas de calidad de McCall.

**Tabla 2-4**

Factor de Calidad \ Métrica de Calidad	Corrección	Confiabilidad	Eficiencia	Integridad	Mantenibilidad	Flexibilidad	Facilidad de Prueba	Portabilidad	Reuso	Inter Operabilidad	Facilidad de Uso
Facilidad de Auditoría				x			x				
Exactitud		x									
Normalización de la Comunicación										x	
Consistencia	x										
Complejidad		x				x	x				
Normalización de los datos										x	
Tolerancia de errores		x									
Eficiencia de ejecución			x								
Facilidad de expansión						x					
Generalidad						x		x	x	x	
Independencia de hardware								x	x		
Instrumentación			x	x			x				
Grado de modulación		x			x	x	x	x	x	x	
Operabilidad			x								x
Seguridad				x							
Documentación propia					x	x	x	x	x		
Simplicidad		x			x	x	x				
Independencia de sistema								x	x		
Facilidad de Capacitación											x

**Tabla 2-4 La Relación entre los Factores y Métricas de calidad de McCall**

### 2.7.7 Factores de Calidad de Lowell Jay Arthur.

Lowell Jay Arthur en [Jay,1988] propone los siguientes factores de calidad para la evaluación del software:

- Facilidad de mantenimiento : ¿Puede arreglarse?, ¿Requiere poco esfuerzo arreglar o modificar el defecto?, ¿Tiene indentación en forma estándar? , ¿Esta bien comentado con prólogo y puntos de decisión?, ¿Son los módulos simples?.
- Flexibilidad: ¿Se le pueden añadir al programa nuevos módulos sin problemas?, ¿Contiene código "spaguetti" o "goto's"?, ¿Existe una copia de bibliotecas para definiciones de datos?.
- Confiabilidad: ¿Correrá y producirá los resultados correctos todo el tiempo?.
- Re-Uso: Puede el sistema, programa o módulo ser usado en otras aplicaciones para reducir costos de desarrollo?.
- Facilidad de uso: ¿Puede el usuario/cliente aprender y usar el sistema fácilmente?, ¿Pueden las operaciones ejecutarse?.
- Eficiencia: ¿Corren las aplicaciones en el hardware tan rápido como es posible?.
- Facilidad de prueba: ¿ Puede probarlo?.
- Integridad: ¿Son las aplicaciones y los datos seguros de cualquier intromisión?.
- Portabilidad: ¿Puede moverse fácilmente de un hardware/ambiente de sistema a otro?.
- Interoperabilidad: ¿Puede interactuar fácilmente con otros sistemas?.
- Corrección : ¿Son los datos y las aplicaciones completos, exactos y consistentes?.

### 2.7.8 Factores de Calidad según Peter Freeman.

Freeman por su parte, afirma que la calidad debe basarse en los requerimientos, que provienen principalmente de los clientes, usuarios, estándares, la buena práctica, etc; [Freeman,1987].

Freeman clasifica los factores de calidad en dos grupos. El primer grupo se refiere a la Calidad Básica y el segundo a la Calidad Extra (tabla 2-5). Esta distinción está basada en el concepto de qué se considera un factor esencial y qué un factor extra.

Calidad Básica	Calidad Extra
Funcionalidad Confiabilidad Facilidad de uso Economía Seguridad	Flexibilidad Reparabilidad Adaptabilidad Facilidad de entender Documentación Facilidad de conexión

Tabla 2-5 Clasificación de los Factores de Calidad según Freeman

A continuación se definen cada uno de los factores propuestos por este autor:

- **Funcionalidad:** Si el sistema puede ejecutar las funciones que son requeridas para la tarea destinada. Un ejemplo sería un reporte financiero que no pueda comparar totales de períodos anteriores.

- **Confiabilidad:** El sistema debe ser capaz de ejecutar sus funciones consistentemente aún en condiciones anormales debe continuar ejecutándose tanto como sea posible, ya que el software que no proporciona resultados correctos o que es propenso a proporcionar estos resultados de manera aleatoria, es considerado pésimo por la mayoría de los usuarios.
  - **Facilidad de uso:** El sistema debe permitir a los usuarios trabajar con conceptos y términos familiares a ellos, aprender a usarlo con un mínimo entrenamiento y proporcionarles un fácil acceso a la información que sea necesaria.
  - **Economía:** El software debe cumplir con sus tareas con los más mínimos recursos. Significa que se tienen que contemplar no solo los recursos de procesador y de memoria, sino también el espacio en disco, los periféricos y el software de soporte.
  - **Seguridad:** La operación del software no debe permitir dañar u ocasionar una destrucción de información. Junto con la evolución de los sistemas , el criterio de la calidad también se ha incrementado y ahora se basa en la habilidad de un sistema para soportar catástrofes.
- 
- **Flexibilidad:** Un sistema es considerado flexible si puede manejar un rango de diferentes entradas. Por ejemplo, cualquier sistema que solo acepte el número de empleado en lugar de aceptar también otro tipo de información (tal como su nombre o el departamento al que pertenece) no se puede considerar flexible.
  - **Reparabilidad :** Es la habilidad de que un sistema corrija sus errores. Un sistema que esta bien diseñado y documentado puede ser muy fácil de reparar.
  - **Adaptabilidad:** La habilidad de que un sistema opere bajo un rango de condiciones. Una situación frecuente es cuando se desea operar un software con diferentes configuraciones de hardware(tal vez con menos memoria o con diferentes tipos de terminales), con un nuevo sistema

operativo, o tal vez en un nuevo ambiente de ejecución. Un sistema adaptable no debe poseer mucha dificultad para resolver estos casos.

- **Facilidad de entender:** Un sistema es entendible si no se requiere de mucho esfuerzo para comprender que hace y como se usa.
- **Documentación:** Una buena documentación ayudará al buen entendimiento del sistema, pero si el sistema no está bien entendido por la persona que documenta, difícilmente se llegará a tener un sistema bien documentado.
- **Facilidad para conectarse a otros sistemas:** Es la habilidad para poder agregar nuevos módulos o interfases a una pieza de software.

### **2.7.9 Factores que afectan a la Calidad y a la Productividad de los Sistemas.**

La palabra productividad es muy ambigua, considerando la frecuencia con la que la usamos. Capers Jones [Jones, 1986] en [Gómez, 1995] después de haber realizado una encuesta entre administradores de sistemas de información, programadores y ejecutivos de empresas, concluyó que la palabra productividad implica dos ideas:

1. Reducir el tiempo para desarrollar un nuevo sistema computacional.
2. Disminuir la cantidad de dinero que la empresa gasta por el nuevo sistema computacional.

Este autor identifica 20 factores que afectan a los proyectos de software, los cuales se listan a continuación:

- 1) El lenguaje de programación utilizado.
- 2) El tamaño del programa.
- 3) La experiencia de los programadores y diseñadores.
- 4) La novedad de los requerimientos.

- 5) La complejidad del programa y los datos.
- 6) El uso de métodos de programación estructurados.
- 7) La clase del programa o el método de distribución.
- 8) El tipo de programa.
- 9) Herramientas y condiciones del medio ambiente.
- 10) Aumentar programas y sistemas existentes.
- 11) Dar mantenimiento a programas y sistemas existentes.
- 12) Reutilizar módulos y estándares de diseño existentes.
- 13) Generadores de programas.
- 14) Lenguajes de cuarta generación.
- 15) Separación geográfica de los programadores.
- 16) Defectos potenciales y métodos para prevenirlos.
- 17) Documentación.
- 18) Desarrollar prototipo antes de comenzar con el desarrollo principal.
- 19) Equipos y la estructura organizacional.
- 20) Moral y compensaciones del staff.

Además de los factores anteriores, Capers Jones identifica 25

factores no cuantificables:

- 1) Restricciones de horario y recursos.
- 2) Horas extras pagadas.
- 3) Tamaño del staff.
- 4) Tamaño total de la organización.
- 5) Agotamiento durante el desarrollo.
- 6) Contrataciones y reubicaciones durante el desarrollo.
- 7) Sistemas de negocios y / o planeación estratégica.
- 8) Participación del usuario en requerimientos y diseño.
- 9) Desarrollo de usuarios finales.
- 10) Centros de Información.



- 11) Centros de desarrollo.
- 12) Entrenamiento y educación del staff.
- 13) Estándares y métodos de desarrollo formales.
- 14) Proyectos cancelados y desastres de software.
- 15) Rediseño y reinicio de proyectos.
- 16) Transferencia de proyectos de una ciudad a otra.
- 17) Tiempo de respuesta y facilidades de la computadora.
- 18) Facilidades físicas y espacios de oficina.
- 19) Adquisiciones y compras de software.
- 20) Políticas internas y poder de lucha.
- 21) Restricciones legales y estatutarias.
- 22) Requerimientos de licencia de exportación.
- 23) Políticas de la empresa.
- 24) Métricas de calidad y productividad.
- 25) Etapas de mejoramiento de calidad y productividad.

Otros factores que el autor no menciona , pero que valen la pena indicar por ser puntos críticos en la productividad en el desarrollo de un producto de software son los siguientes:

- 1) La rotación de personal.
- 2) El grado en que se involucra el personal.
- 3) La tecnología de información manejada.
- 4) La disponibilidad de herramientas de trabajo.
- 5) El ambiente de trabajo.
- 6) El mal uso del presupuesto destinado al proyecto.

Aunque que son muchos los factores que se tienen que controlar por parte de los administradores del área desarrollo de software , pero tal vez los más críticos son los que tienen que ver con el factor humano ya que este es

muy difícil de controlar y muchas veces el éxito de un buen desarrollo consiste en saber distribuir las tareas de acuerdo al carácter y a las aptitudes de cada persona.

## 2.8 El Proceso de Pruebas .

Otra de las actividades de SQA que generalmente se asocia con el personal de aseguramiento de calidad son precisamente las pruebas; y su importancia e implicaciones directas sobre la calidad del software no pueden ser pasadas por alto. La fase de pruebas es el periodo del ciclo de vida del software en el cual los componentes de software son examinados e integrados, y los productos de software son evaluados para determinar si los requerimientos han sido satisfechos o no [Glick,1990].

Paul Jorgensen [Jorgensen,1995] indica que existen dos principales razones para llevar a cabo pruebas en el software: 1) para hacer un juicio acerca de la aceptabilidad de la calidad y 2) para descubrir problemas. "Probamos porque sabemos que el software esta fallando".

El incremento constante en los costos asociados con las fallas de los sistemas están forzando a los desarrolladores a realizar un "plan bien hecho" de pruebas al producto. No es raro que una compañía desarrolladora de software gaste mas del 40 por ciento del esfuerzo del proyecto en probar , incluso en las pruebas del software de alto peligro( como el de monitoreo de reactores nucleares) se gastan de 3 a 5 veces más en pruebas.

Solamente cerciorándose mediante la etapa de pruebas que las aplicaciones cumplen con los requerimientos de los clientes en cuanto a funcionalidad, desempeño, tiempo de ejecución, seguridad y confiabilidad se podrá ofrecer un software de calidad.

### 2.8.1. Objetivo del Proceso de Pruebas.

Glen Myers establece una serie de reglas que sirven acertadamente para describir los objetivos de prueba:

1. La prueba es un proceso de ejecución de un programa con la intención de encontrar un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba exitosa es aquella que descubre un error no detectado hasta entonces.

En la figura 2-4 se ilustra el flujo de pruebas que se debe seguir; donde se pueden apreciar dos tipos de entrada: (1) la configuración de software que incluye las especificaciones de los requerimientos de software, especificación del diseño y el código fuente; (2) la configuración de la prueba que incluye un plan y un procedimiento a seguir, cualquier herramienta de pruebas que vaya a ser usada y los casos de prueba con los resultados esperados. Los resultados de las pruebas son evaluados es decir, los resultados obtenidos son comparados con los resultados esperados. Cuando aparecen datos equivocados, significa que se tiene que empezar a eliminar estos errores.

El proceso de "eliminar errores" es la parte mas impredecible del proceso de pruebas ya que una discrepancia del .01% entre los datos obtenidos y los datos esperados puede tomar una hora, 1 día o hasta un mes en diagnosticar y corregir el error. Una vez que los resultados son recopilados y evaluados, se puede empezar a medir la calidad y la confiabilidad del sistema. Si con regularidad se encuentran numerosos errores en el diseño del software su baja calidad será dudosa en un futuro. Si por otra parte, las funciones del software

parecen que trabajan adecuadamente y los errores encontrados son fáciles de corregir, se puede deducir una de las siguientes dos conclusiones : (1) que la calidad del software es aceptable o (2) que las pruebas no son las adecuadas para cubrir tantos errores.

Finalmente, si no aparecen errores cabe la duda de que la configuración de las pruebas no fue la correcta y de que los errores andarán al acecho. Estos defectos aparecerán eventualmente y serán corregidos en la etapa de mantenimiento (donde el costo por corregir un error puede ser de 60 a 100 veces el costo de arreglarlo en la etapa del desarrollo).

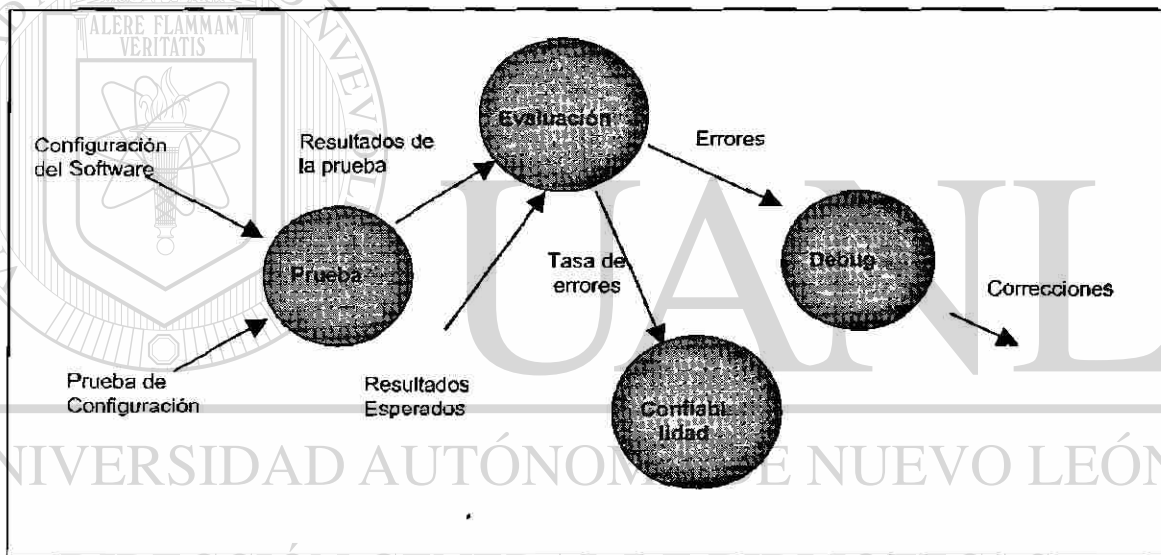


Figura 2-4 Flujo de Información de Pruebas

### 2.8.2. Diseño de Casos de Pruebas.

El diseño de casos de prueba se basa en elaborar un esquema de prueba con alta probabilidad de encontrar errores con la mínima cantidad de tiempo y de esfuerzo, sobre todo desde un enfoque sistemático.

Cualquier producto de ingeniería puede ser probado de dos formas: conociendo la función específica para la que fue diseñado el producto o conociendo el funcionamiento del producto; de la primera manera se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa, mientras que de la otra manera se pueden hacer pruebas para verificar que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. El primer enfoque se denomina prueba de la Caja Blanca , mientras que el segundo prueba de la Caja Negra. [Cristales, 1996]

### **2.8.3. Prueba de Funcionalidad del Sistema.**

Para sistemas en tiempo real, el software que proporciona las funciones necesarias pero que no cumple con los requerimientos de funcionalidad, es inaceptable. La prueba de funcionalidad esta diseñada para probar el tiempo de ejecución del software. Esta prueba se lleva a cabo en todos los pasos del proceso de pruebas. Aun en el nivel de unidad, el tiempo de ejecución de cada módulo es importante. Sin embargo, solo hasta que todos los elementos son integrados puede medirse la funcionalidad completa del software [Pressman, 1992].

### **2.8.4. Prueba de Confiabilidad y Seguridad en la Información.**

Cualquier sistema de computadora que maneje información sensible que cause acciones que puedan dañar (o beneficiar) los datos es blanco de penetraciones ilegales. Estas "penetraciones" abarcan un amplio rango de actividades: hackers que amenazan con penetrar a los sistemas como deporte, empleados disgustados que amenazan con introducirse a los sistemas a manera de venganza, gente deshonesto que amenaza con accesar a la información para obtener ganancias ilícitas, etc.

Las pruebas de seguridad se utilizan para verificar que los mecanismos de protección construidos dentro del sistema lo protejan de cualquier amenaza de penetración.

Durante las pruebas de seguridad, el personal que realiza las pruebas juega el papel del intruso que desea penetrar al sistema. En este proceso cualquier cosa puede suceder, el intruso trata de adquirir claves de acceso usando decodificadores, puede atacar el sistema utilizando software diseñado para dar de baja cualquier defensa que haya sido construida, puede aplastar el sistema negando el servicio a otros, causar intencionalmente errores de sistema, esperar penetrar durante una recuperación del sistema, curiosear en datos inseguros o esperar a encontrar la llave de entrada al sistema.

#### **2.8.5. Prueba de Validación.**

El resultado de una prueba de integración es un paquete de software completamente ensamblado, con errores de interfase que han sido descubiertos y corregidos y es aquí donde una serie de pruebas finales de validación comienzan. La palabra "validación" puede ser definida de muchas maneras, pero una definición simple es aquella que dice que la validación es el conjunto de actividades que aseguran que el software implementa correctamente una función específica y que el software desarrollado concuerda con los requerimientos del cliente.

Es importante notar que tanto la verificación como la validación encajan con las actividades del SQA que incluyen revisiones técnicas, auditorías de calidad y configuración, monitoreo del desempeño, simulación, revisiones de la documentación, análisis de los algoritmos, desarrollo de pruebas, cuantificación e instalación de las pruebas .

## CAPITULO 3

### EL COSTO DE LA CALIDAD Y SU IMPACTO EN LA SATISFACCION DEL USUARIO

#### 3.1. Costo de la Calidad en Sistemas de Información.

Al hablar de calidad en los sistemas de información es inevitable omitir la palabra "costo" puesto que cada vez resulta más caro darle mantenimiento a un sistema de baja calidad .

---

Cuando un sistema es de baja calidad requiere de un mantenimiento continuo debido a que no cumple con las funciones para las que fué creado. Esto trae como consecuencias que se tengan que invertir grandes sumas de dinero en los recursos empleados para estabilizar dicho sistema .

Por esta razón el presente capítulo se enfocará a presentar los costos derivados de no emplear un Plan de Aseguramiento en la Calidad del Software, los que se hubieran podido evitar si se hubiese hecho un buen plan del proyecto de software.

### 3.2. Costos de la Etapa de Desarrollo del Software.

Uno de los puntos principales a considerar en la elaboración de un proyecto de desarrollo de software, es llevar a cabo un análisis costo- beneficio que permita determinar si es más conveniente desarrollar el sistema internamente o comprar un producto del mercado que se amolde a las necesidades del negocio.

Roger Pressman [Pressman, 1992] propone un análisis costo-beneficio donde describe los costos del desarrollo del proyecto y los compara contra los beneficios tangibles (medidos en pesos) e intangibles del sistema. Este análisis varía de acuerdo a las características del sistema, al tamaño del proyecto y a las expectativas del negocio de los resultados obtenidos al optimizar procesos con el uso del sistema. Como ya se mencionó, muchos de los beneficios obtenidos pueden ser intangibles (por ejemplo una mejor calidad en el diseño, un incremento en la satisfacción del usuario, una mejor toma de decisiones como resultado de la calidad los datos obtenidos, etc. ) .

---

El administrador del proyecto debe crearse una visión de los posibles beneficios que se pueden obtener del proyecto, en donde el objetivo primordial esté enfocado al cumplimiento de los objetivos que se haya trazado los cuales pueden concentrarse en un rápido acceso a la información y su impacto en las funciones del usuario , en una mejor ingeniería de software, o en un incremento en las actividades del procesador .

Así por ejemplo, en un sistema de diseño de ayuda por computadora, mediante el cual se dibujan diseños de ingeniería, los posibles beneficios que se esperarían serían un incremento en el número de diseños diarios y una reducción en el tiempo necesario para llevar a cabo cada dibujo. Suponiendo que se obtuvieron las siguientes mejoras del sistema propuesto :



(t) Tiempo promedio de elaboración de cada diseño = 4 horas.

(\$\$) Costo por hora de cada dibujante = 20.00.

(n) Número de dibujos por año = 8,000.

Se podrían estimar la reducción de costos anuales de los dibujos:

**Costos ahorrados en el dibujo del diseño = (t) X (\$) X (n) =**

$$(4) X (20) X (8,000) = 640,000$$

Lo que indica que existe un ahorro anual de \$ 640,000 empleando el nuevo sistema.

Pressman [Pressman, 1992] añade que la contraparte consiste en realizar una tabla con los costos empleados en el desarrollo del software. El analista debe estimar cada costo para empezar a construir una gráfica en la que se muestren el punto de recuperación de la inversión, punto de quiebra y el período de reembolso del sistema.

**Tabla 3-1**

***Possible información de costos del sistema***

Costos de obtención

- Consultorías
- Nueva tecnología requerida
- Instalación de equipo
- Modificaciones al SITE (aire acondicionado, seguridad, etc.)
- Capital
- Costos administrativos

Costos de arranque

- Del sistema operativo
- Del equipo de comunicaciones necesarios (líneas telefónicas, líneas de datos, etc.)
- Contrataciones

Costos del proyecto

- Costos de la compra de aplicaciones del software
- Costos del personal, gastos generales, etc. del desarrollo del software
- Capacitación
- Documentación
- Costos de la administración de proyecto

Otros Costos

- Costos del Mantenimiento del sistema (hardware, software, facilidades)
- Rentas (electricidad, teléfonos, etc.)
- Depreciaciones de hardware

**Tabla 3 -1 Costos Estimados en el desarrollo del software**

### 3.3. Costos de Mantenimiento del Software que no Utiliza Métricas SQA.

El término “mantenimiento” ha sido aplicado al proceso de modificar un programa después de que ha sido liberado y está en uso. Estas modificaciones pueden abarcar cambios muy sencillos desde corregir errores en el código, cambios mas complejos para corregir errores en el diseño o cambios drásticos para corregir errores en las especificaciones o para acomodar nuevos requerimientos.

Ian Sommerville [Sommerville,1989] menciona que es imposible producir sistemas de cualquier tipo que no necesiten mantenimiento. A lo largo del ciclo de vida del sistema, los requerimientos originales pueden ser modificados para reflejar cambios en las necesidades, en el entorno y pueden surgir errores de validaciones.

Existen tres tipos de mantenimiento:

- Mantenimiento de perfección.
- Mantenimiento de adaptación.
- Mantenimiento correctivo.

El mantenimiento de perfección son los cambios demandados por el usuario o por el programador del sistema que mejoran el sistema de tal manera que no afectan la funcionalidad del mismo. El mantenimiento de adaptación son las modificaciones que se realizan al sistema debido a cambios en el entorno del programa y el mantenimiento correctivo son las correcciones de errores descubiertos en el sistema.

Un examen realizado por Lientz y Swanson(1980) arrojó como resultados finales el que aproximadamente un 65% del mantenimiento es perfectivo, un 18% es mantenimiento adaptativo y un 17% es correctivo, como se muestra en la figura 3-1. Además, encontraron que una larga lista de empresas dedican alrededor de un 50% de sus esfuerzos de programación en mantener sus sistemas actuales.

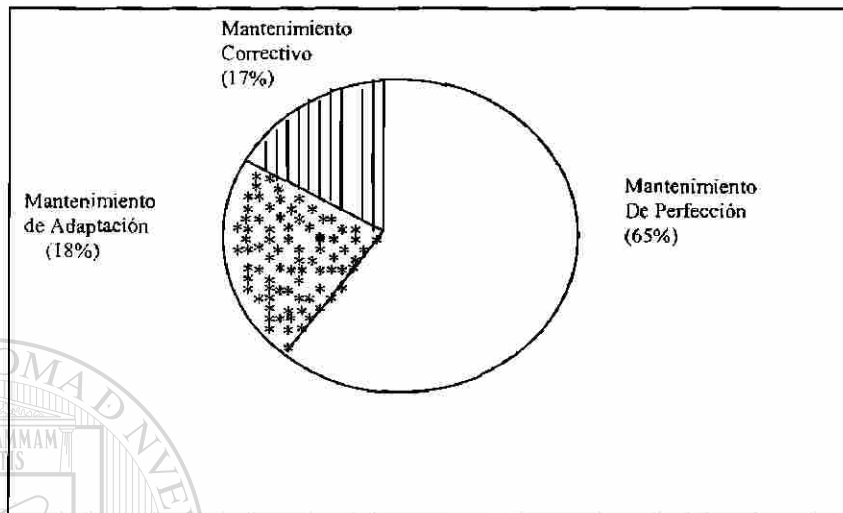
Las técnicas utilizadas en el mantenimiento del software deberán ser aquellas que se emplearon en el desarrollo del mismo. Los nuevos requerimientos deben ser formulados y validados, los elementos del sistema deben ser rediseñados e implementados y la mayor parte de ellos deben ser probados.

Es particularmente importante que el personal que da mantenimiento al software ponga atención en la manera en que realiza las modificaciones al software, ya que debe respetar al máximo la estructura del programa.

Uno de los problemas de la administración del mantenimiento al software es que el mantenimiento tiene una imagen muy pobre entre los ingenieros de software, ya que es visto de manera que resta habilidades al programador y en algunas organizaciones el mantenimiento es delegado a gente inexperta. El resultado de esta imagen negativa es que los costos de mantenimiento se ven incrementados porque el personal no tiene la suficiente experiencia en el desarrollo de las aplicaciones y esto provoca retrasos en los tiempos de entrega de las modificaciones requeridas.

Los costos de mantenimiento son difíciles de estimar. La experiencia dice que los costos de mantener el software son mucho mas grandes que en la etapa de desarrollo y uso del sistema. Generalmente, estos costos son subestimados cuando el sistema fue diseñado e implementado. Los costos de

mantenimiento varían de aplicación en aplicación pero en promedio parecen ser de 2 a 4 veces los costos de la etapa del desarrollo.



**Figura 3 -1 Distribución de los Tipos de Mantenimiento**

A medida que el software se va utilizando, se requiere de un esfuerzo mayor para mantenerlo. La razón de esto es que estos sistemas pudieron haber sido escritos en lenguajes de programación obsoletos que no son utilizados por mucho tiempo en la programación de aplicaciones.

Sommerville propone una teoría muy interesante que dice que vale la pena invertir tiempo y esfuerzo en el diseño e implementación del sistema con el objetivo de reducir los costos en el mantenimiento del mismo. Indica que un esfuerzo extra en el desarrollo representa un multiplicador negativo en los costos de mantenimiento (como se aprecia en la figura 3-2). Un porcentaje de incremento de costos en el desarrollo disparará un decremento en los costos de la etapa de mantenimiento.

Además, indica que los costos se relacionan con dos tipos de factores: los factores técnicos y los factores no técnicos.

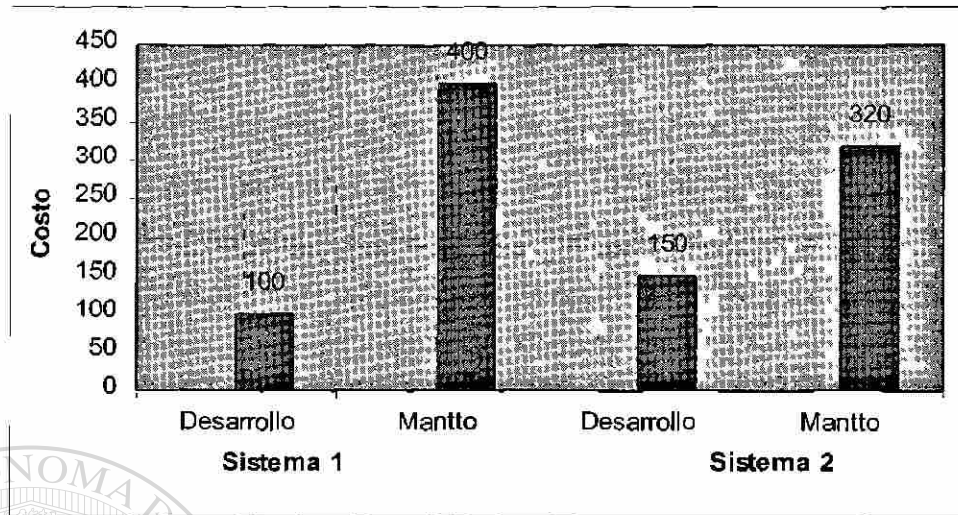


Figura 3-2 Costos de Desarrollo y Mantenimiento

Algunos de los factores no técnicos son los siguientes:

1. La aplicación debe ser soportada. Si la aplicación del programa está claramente definida y es bien entendida, los requerimientos del sistema pueden ser definitivos y el mantenimiento debido a cambios puede ser minimizado. Si la aplicación es completamente nueva, los requerimientos originales se cambiarán tan frecuentemente como los usuarios ganen experiencia en el manejo del sistema.
2. Estabilidad del personal. Es más fácil para el programador original entender y cambiar un programa que si se contrata a un programador nuevo que tenga que entender el programa teniendo que estudiar la documentación del código. Además, si el programador del sistema es quien realiza los mantenimientos al sistema, los costos de mantenimiento se verán reducidos. En la práctica, la naturaleza de un programador consiste en cambiar regularmente de trabajo. Es muy raro que el mismo personal desarrolle y de mantenimiento al sistema a lo largo de su ciclo de vida.

3. La vida de un programa. La vida útil de un programa depende de su aplicación . Los programas se vuelven obsoletos cuando la aplicación se vuelve obsoleta o el hardware original es remplazado y los costos de la conversión exceden los costos de re-codificación. Los costos de mantenimiento tienden a incrementarse conforme un programa se va haciendo viejo.
4. La dependencia de un programa a su entorno. Si un programa depende de su entorno externo, debe ser modificado conforme sufra cambios. Por ejemplo, si un sistema de impuestos es cambiado, los programas referentes a la nómina, contabilidad e inventarios también sufrirán cambios.
5. Estabilidad en hardware. Si se diseña un programa para ser operado con una configuración particular de hardware y esta configuración no cambia durante la vida útil del programa, no habrá costos por mantenimiento. Sin embargo esto es muy raro debido a que la tecnología cambia muy rápidamente. El programa debe ser modificado de manera que pueda utilizar el hardware que remplace al equipo obsoleto.

---

Además de los factores no técnicos antes mencionados, los costos por mantenimiento son controlados por factores técnicos. Algunos de ellos son:

1. Independencia de módulos. En medida de lo posible, debe modificarse un programa de un módulo sin afectar otro módulo.
2. Lenguajes de programación. Los programas escritos en un lenguaje de programación de alto nivel son fáciles de entender comparados con programas escritos con un lenguaje de bajo nivel.
3. Estilo de programación. La manera en que un programa es codificado contribuye a su correcto entendimiento a la hora de ser modificado.

4. Validaciones y pruebas de programa. Generalmente la mayor parte del tiempo y el esfuerzo son empleados en las validaciones y pruebas de los programas; los pocos errores en el programa y por consecuencia los costos de mantenimiento, son menores. Los costos por correcciones dependen del tipo de error a ser reparado. Los errores de código son relativamente baratos de corregir, los errores en el diseño son mas caros porque requieren la recodificación de uno o más módulos. Los errores en los requerimientos son mucho mas caros porque involucran cambios muy drásticos en su diseño.
5. Calidad y cantidad de la documentación. Si un programa cuenta con una documentación completa y concisa, cualquier programador lo entenderá enseguida. Por consecuencia, los costos de mantenimiento tenderán a ser menores .

### 3.4 Costos de la Calidad del Software.

Quando se habla de calidad en el software, el primer pensamiento que viene a la mente suele ser ¿cuánto me va a costar? . Esta respuesta solo puede ser contestada en relación a lo que ya se hecho o a lo que aún no se ha hecho. Las cosas que ya están hechas pueden medirse mediante las horas-hombre gastadas, mientras que las que aún no se han realizado suelen medirse por el nivel de impacto que causan en otras actividades por no llevarse a cabo.

Schulmeyer y Mcmannus en su "Manual de Aseguramiento de Calidad en el Software" [Schulmeyer,1987] indican que para determinar las tareas que se deben realizar para asegurar la calidad en los sistemas de información, es necesario realizar una evaluación de las horas que se necesitarán, el impacto en el plan de trabajo y los beneficios que se obtendrán en términos de calidad del producto, los costos implicados , recursos humanos necesarios, la ventaja

competitiva que se obtendrá y los costos de mantenimiento posteriores a su liberación.

### 3.5 Costos de los Recursos y Tecnología Empleados.

Schulmayer y Mcmannus mencionan los siguientes puntos críticos que no se deben omitir a la hora de costear la calidad en el software:

- **Recolección de datos:** los datos pueden ser obtenidos del análisis del sistema, proporcionados por el usuario o pueden ser salidas de otro proyecto. Una vez que los datos han sido determinados, el costo obtenerlos puede ser determinado.
- **Material:** Se deben determinar los costos de compra del material que se necesitará, por ejemplo la adquisición de computadoras personales para el análisis de los datos obtenidos.
- **Requerimientos del usuario:** los requerimientos que no son bien definidos o que son cambiados constantemente tendrán un impacto significativo en la calidad del producto. Se debe hacer una diferencia entre las necesidades y los deseos de los usuarios, ya que el cliente desea todo, pero regularmente solo pueden pagar lo que realmente necesitan. La principal dificultad de determinar los costos de los requerimientos no radica en escribirlos, sino en las diferentes maneras en que serán desarrollados una vez que han sido re-definidos, interpretados e implementados.
- **Capacitación:** Casi todas las actividades de calidad requieren capacitación, la cual involucra costos del personal que proporciona el entrenamiento (salario, gastos de transportación, etc. ); y el equipo necesario para llevarla a cabo .



- Recursos: Los recursos necesarios para la recolección de datos , la obtención del material o la capacitación deben ser costeados. La facilidad de recursos para estos elementos puede ser costeada en términos de espacio, recursos humanos, tecnología requerida, etc.
- Procedimientos: Cualquier actividad que involucre cambios en los requerimientos de los usuarios, deber ser acorde con el manual de procedimientos de la empresa , el cual si no existe, debe ser creado, ocasionando de esta manera costos de creación.
- Estándares: Cualquier actividad asociada con el desarrollo del producto debe ser hecha bajo ciertos estándares. Estos estándares pueden ser ya establecidos por el gobierno o la industria, o pueden ser internos (de la misma empresa). Aún cuando una actividad es llevada a cabo solo por cumplir con un estándar y no porque se necesite para mejorar la calidad del software, el costo de esta actividad debe ser considerado como costo de calidad.

### 3.6 Nivel de Satisfacción del Usuario .

El principal objetivo que persigue SQA es que el usuario quede satisfecho con la calidad del software que está adquiriendo. Esta calidad va ligada a la concordancia entre los requerimientos y los resultados esperados por el usuario.

En un estudio realizado por Alberto Cristales [Cristales,1996] para determinar el impacto de las actividades de SQA en el desarrollo de software en empresas de la localidad , se llegó a la conclusión de que a los usuarios no les interesa tanto el grado de calidad que tengan los sistemas al ser desarrollados, en realidad lo que les interesa es que sean exitosos al cumplir con sus requerimientos.

Recordando que los factores basados en la información que señalaron los usuarios como características de éxito fueron:

- La información generada debe ser confiable.
- La información generada debe ser relevante para su función.
- La información generada debe ser completa.

Aunque esto último viene a contradecir el objetivo principal de las actividades SQA, esto se puede explicar si se toma en cuenta que la percepción de los usuarios acerca del éxito de los sistemas generalmente es alta a pesar de que los desarrolladores consideren que sus sistemas no tienen un buen grado de calidad en realidad.

Para que los usuarios queden completamente satisfechos con el producto de software que adquieren, es necesario involucrarlos de manera que se sientan parte del proyecto y mientras más información se proporciona al usuario, más comprometido se sentirá.

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



# CAPITULO 4

## METODOLOGIA

### 4.1 Introducción.

Después de haber revisado algunos de los conceptos más importantes acerca del Aseguramiento de la Calidad del Software (SQA) como lo son las métricas de calidad a seguir en el desarrollo, diseño e implementación de los sistemas de información; y de analizar la información acerca de las factores por los cuales no se siguieron dichas métricas en la producción del sistema ABAPRO (Abastecimientos y Proveedores); cuya repercusión se reflejó en costos muy elevados tanto en su implementación como en su mantenimiento sobrepasando incluso los costos originados en el análisis y desarrollo del mismo; se presenta en este capítulo la metodología seleccionada para la obtención, análisis y evaluación de dichos factores.

### 4.2 Metas de la Medición.

Con la intención de alcanzar el objetivo principal de la investigación se consideró establecer como metas de medición las siguientes:

1. Conocer y evaluar las métricas que se utilizaron para la obtención de calidad en el sistema ABAPRO durante su construcción.
2. Conocer el grado de influencia que tuvo el empleo / omisión de estas técnicas en el nivel de calidad de sistema en el momento de su implementación.
3. Evaluar y comparar los gastos generados en las etapas de análisis, diseño, implementación y mantenimiento con el objetivo de demostrar que vale la pena invertir tiempo y esfuerzo en el diseño e implementación del sistema con el fin de reducir los costos en el mantenimiento del mismo.
4. Medir el grado de satisfacción del usuario en la etapa post-implementación para conocer el nivel con que el sistema cumplió con sus expectativas de operación.
5. Demostrar de manera gráfica la relación que existe entre la falta de métricas de calidad y la satisfacción del usuario.

#### **4.3 Modelo Propuesto.**

El modelo propuesto se ilustra en la figura 4-1 y se desarrollará como sigue: el instrumento de obtención de la información será mediante la recopilación de archivos históricos que demuestren que tanto los administradores como los desarrolladores que intervinieron en el proyecto ABAPRO no emplearon eficientemente las métricas básicas de calidad de un sistema: fiabilidad, eficiencia, integridad, facilidad de uso, facilidad de mantenimiento, flexibilidad, portabilidad, reusabilidad, interoperabilidad, etc.

Para la evaluación de estos factores se emplearán las guías propuestas por Peter Freeman y Lowell Jay Arthur, descritas en el capítulo 2 de la presente tesis.

Se realizará un estudio comparativo de los costos incurridos en el desarrollo del ABAPRO (viáticos, honorarios del personal, gastos por la adquisición de nueva infraestructura, etc.) contra los costos ocasionados durante su mantenimiento, empleando como herramienta principal las hojas de cálculo de Excel (Microsoft), donde se graficarán estos datos para poder ser analizados mediante estadística descriptiva.

Asimismo se explotará de la base de datos del negocio (la cual se encuentra en Lotus Notes) un reporte con las solicitudes de servicio más significativas de cada usuario con el fin de analizar el grado de satisfacción y de operabilidad del sistema.

Estas solicitudes contienen datos que proporcionan el nivel de calidad del sistema ABAPRO en el período posterior a su implementación tales como:

- El número de solicitud de servicio.
- Fecha de elaboración.
- Nombre del usuario que coloca la solicitud.
- El tema ( sistema al que se le solicita el servicio).
- El tipo de servicio.
- La prioridad de la solicitud (grado en que afecta el problema en la operación del sistema).
- Calificación de satisfacción del usuario .

Al final se analizará toda la información obtenida mediante el uso de estadística descriptiva para probar la hipótesis de que la omisión de métricas de calidad impacta tanto en los costos como en el nivel de calidad de los sistemas de información.

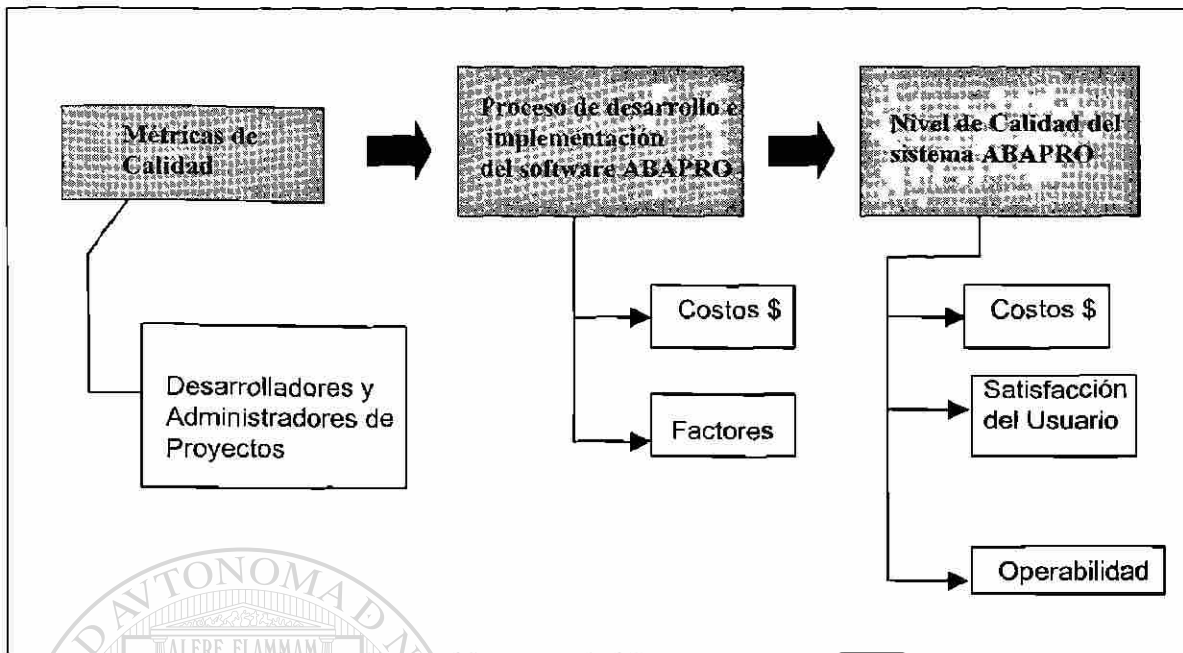


Figura 4-1 Modelo Propuesto de la relación de las Métricas de Calidad con el Éxito de los Sistemas

#### 4.4 Tamaño de la Muestra.

El tamaño de la muestra será de 240 solicitudes de servicio dentro de una población infinita, ya que desde el último trimestre de 1999 que fue cuando se adaptó este control a la fecha, se siguen generando solicitudes diariamente de los 15 negocios de la compañía a los 8 sistemas institucionales.

#### 4.5 Variables.

Las variables implícitas son:

- La fiabilidad de los documentos a analizar.
- El tamaño de la muestra.
- La calificación de calidad de cada usuario.
- La criticidad de las solicitudes de servicio.
- La rotación del personal.

# CAPITULO 5

## ANALISIS DE LOS DATOS, CONCLUSIONES Y RECOMENDACIONES

### 5.1 Historia de la Empresa.

La historia de Grupo Cydsa se inicia en 1945 en Los Lirios, una antigua granja ubicada en las afueras de Monterrey. En ese año, Cydsa produce por primera vez en México fibra de rayón para la creciente industria textil. Dos años después de iniciar operaciones, se embarca en la producción de seda artificial, por lo que empieza a fabricar cuerda de rayón para la manufactura de llantas.

En 1955 se empieza a utilizar la viscosa, materia prima de rayón, para producir película transparente de celulosa. Como respuesta a una severa escasez de agua a principios de los años 50's, Cydsa crea Copropiedad, una planta de servicios comunes que provee agua, vapor y energía eléctrica a las plantas de Cydsa en Monterrey. En el año de 1958 entra en operación la planta de cloro y sosa cáustica de Plantas Químicas.

En el año de 1961, Cydsa visualiza la asociación con empresas internacionales y se asocia con Allied Chemical para crear Quimobásicos, empresa proveedora de gases refrigerantes .

En 1965 se integra al grupo Reyprint, una nueva planta productora de empaque y envolturas flexibles. Ese mismo año arranca en Jalisco la planta Crysel, la cual produce fibras acrílicas. Dos años más tarde, Cydsa adquiere Industria Química del Istmo, productora en México de cloro y sosa cáustica.

En 1971, Cydsa se asocia con la empresa extranjera BF Goodrich y adquiere la proporción mayoritaria de Policyd, empresa que utiliza la más avanzada tecnología en la fabricación de resinas de PVC, materia prima empleada en una amplia gama de productos de plástico. Para 1975 se asocia con Bayer A.G. y establece en Veracruz la compañía Industrial Cydsa-Bayer, dedicada a la elaboración de espumas flexibles y semirígidas de poliuretano.

En 1979 Cydsa crea dos empresas nuevas: Propirey en Monterrey, pionera en la fabricación de polipropileno para la industria de alimentos empaquetados, y Policyd Altamira. En 1981 el grupo adquiere Plásticos Rex, empresa que fabrica tubería, conexiones de PVC y sistemas de riego pasteurizado por goteo. En 1990 forma una nueva división, Mejoramiento Ambiental y crea la empresa Atlatec. Un año más tarde nace la empresa Masterpak y adquiere la empresa Litoenvases, fabricante de cartón plegadizo de calidad.

En 1992, Cydsa adquiere el Grupo Textil San Marcos en Aguascalientes y se consolida como el grupo textil más integrado en el Hemisferio Occidental.

## **5.2 Organización del Departamento de Informática.**

La organización del departamento de Informática se muestra en la figura 5-1.



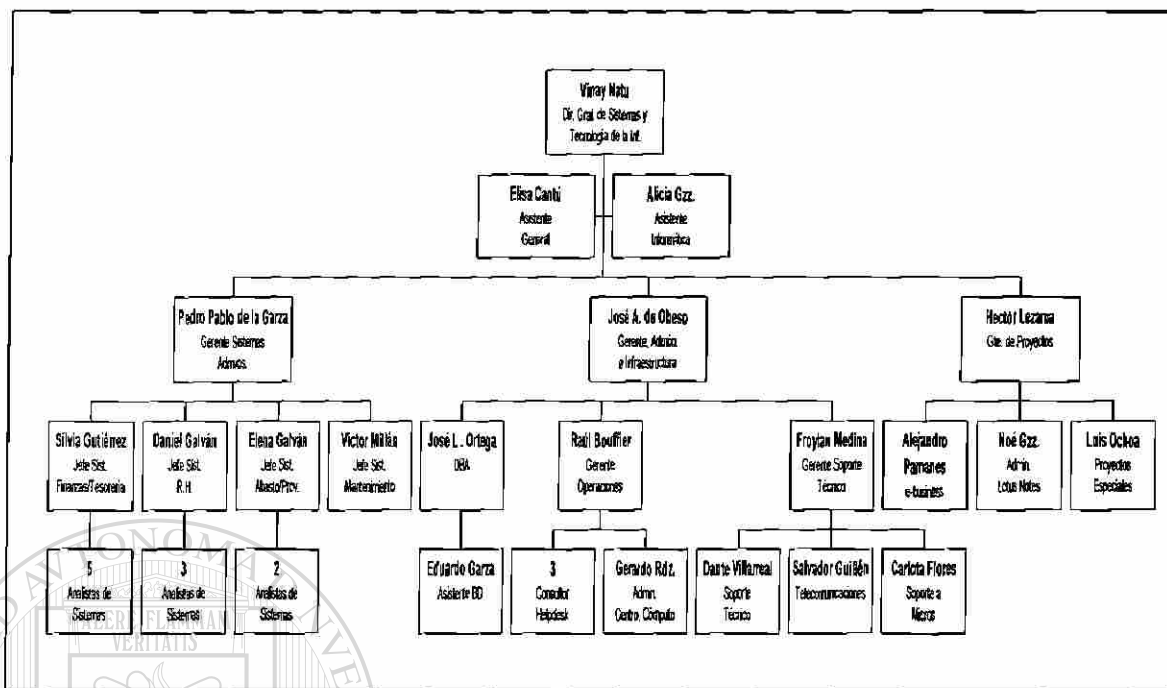


Figura 5-1 Organigrama del Departamento de Informática

### 5.3 Tecnología de Información.

El área de Informática se centraliza en el Corporativo de la empresa, por lo que la administración de los sistemas institucionales (soporte y el mantenimiento) de los 14 negocios del grupo, está asignado a Informática central.

El software que maneja el área administrativa es el siguiente:

- Windows 95 , Windows NT (Ver 4.0).
- Utilerías del Norton.
- Microsoft Office 97 (Word , PowerPoint, Excel, Microsoft Project).
- Lotus Notes (Ver 4.5).
- KEA! (Ver 5.0).

El software que utiliza el área de desarrollo es el siguiente:

- Progress 8 , corriendo bajo plataforma UNIX.
- Progress 9 , corriendo bajo plataforma UNIX.
- Visual Basic 6.0 .
- Web Speed 3.0.

Los sistemas institucionales que se manejan en el grupo son el de Nómina, Caja de Ahorro, Gastos de Viaje, Tesorería, Abastecimientos y Proveedores, Contabilidad, Ventas, Mantenimiento y PRO5%, los cuales son desarrollos propios y están hechos en el lenguaje de programación Progress 8, excepto el sistema de Mantenimiento, el cual esta hecho en Progress Gráfico y el de PRO5%, que está construido en Visual Basic 6.0.

Actualmente la empresa cuenta con su propio HOME PAGE y se está desarrollando un prototipo de una INTRANET bajo WebSpeed 3.0 para el sistema de pago a Proveedores.

El estándar de PC que se tiene para el personal es el siguiente:

- PC Genérica.
- Pentium PRO III (450-550 MHZ).
- 64 MB de memoria RAM.
- mínimo 6 GB en disco duro.
- Tarjeta Ethernet.

-Impresoras:


- HP LaserJet 4000, 2100.
- HP LaserJet II P.

- HP LaserJet 4, 4P y 5P.
- Tektronix Phaser 560 (color).
- Epson LQ-870 , FX-1050, LQ-2550.

-FAX:

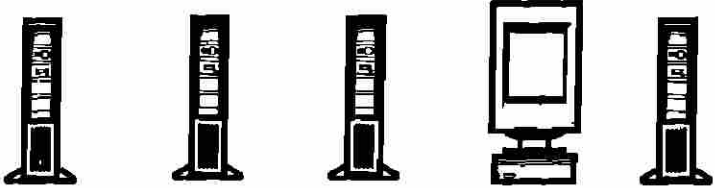
- Canon Multipass 800, Multipass C3500.
- Canon Fax-730, Fax-230 , Fax CFX-L4000.
- Panasonic KX-F580.

El Equipo que integra el Centro de Cómputo puede apreciarse en las figuras 5-2 y 5-3.



NOMBRE SERVER	CORCYD1	CORCYD2	CORCYD3	CYDNOTES1 CORP_MTY_1	CYDNOTES2 CORP_MTY_2
SERVICIOS	Backup DC : File & Print Server, Sistemas-Fox	Backup DC : Logon, File Server y Intranet	Primary DC : Logon, File & Print sevrer, Sala- Consejo y Proyectos	Lotus Notes: Usuarios México e Intermex y Remotos, Replicador de BD.	Lotus Notes: Usuarios Internos y BD
MARCA	COMPAQ	COMPAQ	COMPAQ	COMPAQ	COMPAQ
MODELO	PROLIANT	PROLIANT 5500	PROLIANT	PROLIANT 2500	PROLIANT
CPU	Pentium 133	Pentium III X	Pentium Pro 200	Pentium Pro 200 Mhz	Pentium III X
DISCOS	1500 Mhz 6 GB	45 GB	16 GB	12 GB	500 Mhz 45 GB
MEMORIA	RAM 256MB	RAM 256MB	RAM 128 MB	RAM 224 MB	RAM 256MB
SOFTWARE	RAM 64 MB  Windows NT 4.0 Compaq Utilities	Windows NT 4.0 Compaq Utilities SQL Server 7.0 SMS 2.0	Windows NT 4.0 Compaq Utilities 3.5 Norton AVirus NT	Windows NT 4.0 SP4 Compaq Utilities 3.5 Lotus Domino 4.6.1	Windows NT 4.0 Compaq Utilities Lotus Domino 4.6

Figura 5-2 Servidores de Uso Interno



	CYDNOTES3	CYDWWW	CYDEXNET	CYDFIREWALL	CYDDES
NOMBRE SERVER					Extranet de Desarrollo
SERVICIOS	SMTP_MTY_1 Gateway de Lotus-Notes a E-Mail de Internet	INTERCYD_MTY_1 Home-Page www.cydsa.com	INTERCYD_MTY_2 Extranet de Negocios Crysel / P.Rex	Gateway de Lotus-Notes a E-Mail de Internet	
MARCA					DIGITAL
MODELO	COMPAQ	COMPAQ	COMPAQ	COMPAQ	PRIORIS XL 590
CPU	PROLIANT 800	PROLIANT 2500	COMPAQ	DESKPRO 4000	Pentium 90 Mhz
DISCOS	Pentium Pro 200 Mhz	Pentium Pro 200 Mhz	PROLIANT 3000 Mhz	Pentium Pro 180 Mhz	6 GB
MEMORIA	4 GB	4GB	Pentium II 300 Mhz	2.4 GB	RAM 80 MB
SOFTWARE	RAM 224 MB	RAM 128 MB	RAM 128 MB	RAM 256 MB	Windows NT 4.0 SP4
	Windows NT 4.0 SP4	Windows NT 4.0 SP4	Windows NT 4.0 SP4	Windows NT 4.0 SP4	Compaq Utilities 3.5
	Compaq Utilities 3.5	Compaq Utilities 3.5	Compaq Utilities 3.5	FIREWALL-1	Lotus Domino 4.6
	Lotus Domino 4.6.3	IIS	Lotus Domino 4.6.3		IIS
		FTP			FTP
		Progress			Progress
		JWS			JWS

Figura 5-3 Servidores Con Conexión Externa

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



## 5.4 ABAPRO. DIRECCIÓN GENERAL DE BIBLIOTECAS

El sistema ABAPRO surge como una conjunción de los sistemas de Abastecimientos y Proveedores, los cuales estaban diseñados de manera independiente bajo el lenguaje de programación llamado COBOL. Debido a que estos sistemas presentaban las siguientes deficiencias:

- 1) Estaban desarrollados bajo un lenguaje de programación de bajo nivel.
- 2) Presentaban dificultad en la interacción con otros sistemas desarrollados en Progress.

- 3) Las interfases de usuario eran poco amigables.
- 4) No cubrían las necesidades administrativas.
- 5) Elevada inversión de tiempo y recursos operativos en la ejecución de los procesos.
- 6) No cumplía con los requerimientos necesarios para llegar al año 2000 (por el manejo en fechas principalmente).

Se diseñó un nuevo sistema que abarcaba las funciones de estos dos sistemas.

El sistema ABAPRO es un sistema que permite llevar un control de las compras hechas a los proveedores, así como el flujo que se sigue desde que se hace una requisición, se convierte en compra, se le da entrada al sistema, se factura y posteriormente se le coloca un cheque. Además, maneja contratos de servicio y permite llevar un control de las entradas y salidas del inventario, su costeo, ajustes y cálculo del punto de re-orden para cada artículo.

También proporciona reportes financieros que permiten diseñar estrategias de negociación de pagos a los proveedores.

Los módulos principales del sistema son: Compras, Requisiciones, Cotizaciones, Pedidos, Almacén (Entradas, Salidas, Inventario Físico, Costo Estándar), Contratos de Servicios y Cuentas por Pagar.

ABAPRO tiene interfase con los sistemas de Tesorería (autorización de pagos y actualización del saldo del proveedor), Contabilidad (validación de cuentas contables y descarga de la contabilidad de los movimientos) y Mantenimiento (validación de las ordenes de servicio).

En la siguiente figura se puede apreciar un esquema general del sistema.

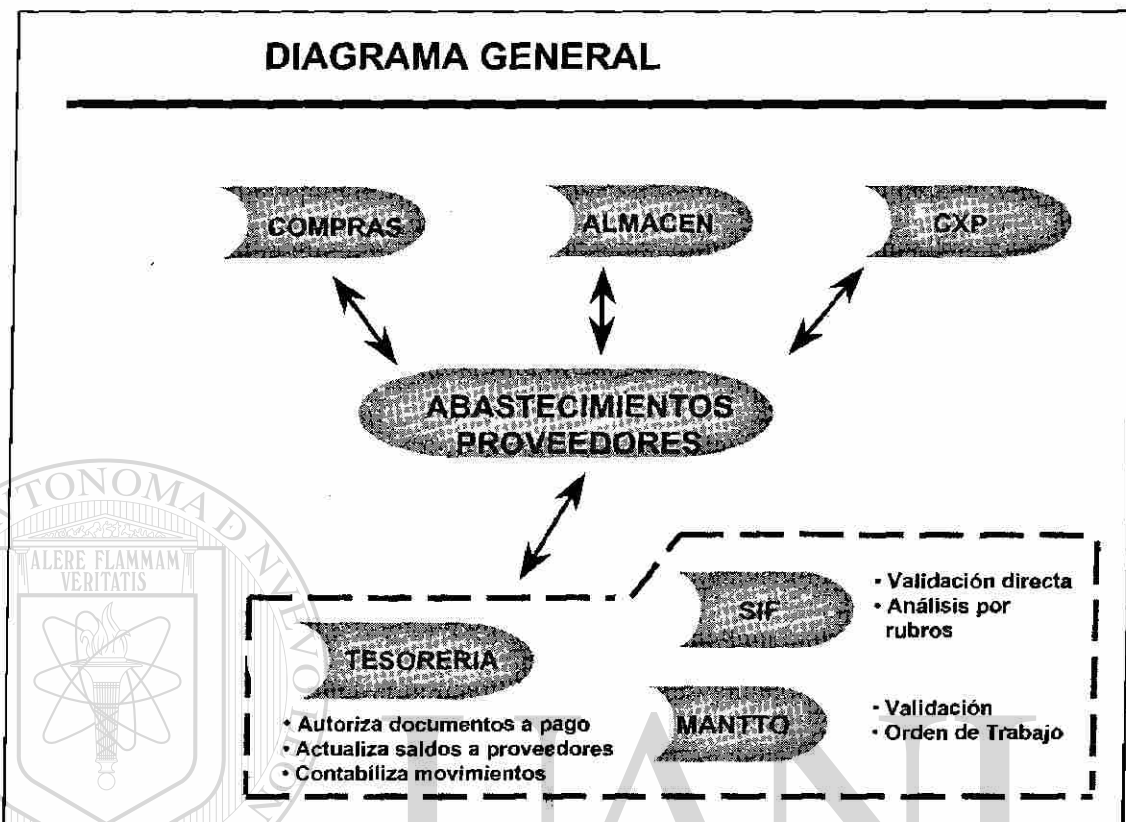


Figura 5-4 Esquema General del Sistema ABAPRO

## 5.5 Análisis de las Etapas de Desarrollo del Sistema ABAPRO

A continuación se presenta un estudio exploratorio de los costos y la evaluación de las métricas de calidad empleadas en cada una de las etapas de construcción del sistema ABAPRO.

### 5.5.1 Costos del Análisis, Diseño y Desarrollo del ABAPRO.

El análisis, diseño y desarrollo del sistema ABAPRO se llevó a cabo en el periodo de Septiembre de 1996 a Mayo de 1998. Durante este tiempo se realizaron viajes a los negocios con la finalidad de establecer los procesos de negocio y tratar de estandarizar un solo sistema ya que cada planta tenía

definido su propia metodología de trabajo. Uno de los principales objetivos era que el sistema fuera multi-compañía, ya que uno de los problemas era que para cada compañía-planta existía implementado un sistema de Abastecimientos y otro de Proveedores.

Otros gastos generados son los honorarios de los programadores y analistas ya que hubo mucha rotación de personal debido en gran parte a que continuamente se realizaban largas jornadas de trabajo para cumplir con las fechas de entrega y a que se contrataba a personal inexperto en la herramienta.

La relación de todos estos gastos (totales) se presenta en la tabla 5-1, y el detalle de los mismos se puede apreciar en los anexos A-D.

Concepto de Gasto	Monto \$	Divisa	Periodo Inicial	Periodo Final
Viáticos	245,947.90	MN	Ene-97	Dic-97
Honorarios	209,408.00	MN	Sep-96	Dic-96
Honorarios	508,774.00	MN	Ene-97	Jun-97
Honorarios	415,956.00	MN	Jul-97	Dic-97
Honorarios	425,113.20	MN	Ene-98	May-98
<b>TOTAL (MN)</b>	<b>1,805,199.10</b>			

**Tabla 5-1 Gastos generados durante el análisis y diseño del sistema ABAPRO**

### 5.5.2 Empleo de los Factores de Calidad de Freeman para la Evaluación del ABAPRO.

La fuente de información para esta evaluación fueron documentos históricos pertenecientes al en ese entonces administrador del proyecto ABAPRO. En ellos se detallan el plan de trabajo, especificaciones, administración de los recursos empleados, etc; contenidos en formatos de Word, Excel y PowerPoint ; además de la experiencia personal de una servidora en el momento en que se arrancó con la implementación de este sistema.

Recordando el principio fundamental en el que Peter Freeman basa la definición de calidad :

“La calidad debe basarse en los requerimientos que provienen principalmente de los clientes, usuarios, estándares, buena práctica, ect”. [Freeman,1987].

En este caso, nunca se tomó en cuenta el visto bueno del usuario en cada uno de los procesos que se iban terminando ya que los programadores se basaron únicamente en las especificaciones escritas; además, nunca se contó con un manual de procedimientos de negocio, o al menos, los desarrolladores y analistas nunca supieron de su existencia.

Recordando que Freeman clasifica la calidad en dos grupos, el de la calidad básica y el de la calidad extra (ver tabla 2-5 ), a continuación se procederá a evaluar los factores de calidad básica, los cuales han ido cambiando conforme se le ha dado mantenimiento al sistema, por lo que esta evaluación corresponde al momento en que se terminó el análisis y diseño.



- **Funcionalidad:** El sistema ABAPRO no cumplía con las funciones para las cuales fué diseñado, ya que varios módulos (principalmente el módulo contable) fueron rediseñados en la etapa de implementación, lo cual indica que en su inicio, el sistema tenía una mala funcionalidad.
- **Confiabilidad :** El grado de confiabilidad era medianamente aceptable, ya que aunque Progress posee un espejeo de transacciones, muchos programas no fueron estructurados para deshacer las transacciones si se generaba un error por parte del usuario.
- **Facilidad de uso :** Aunque el sistema se puede considerar amigable en su mayor parte, al implementar el sistema no se contaba con un manual de usuario formal y en la captura de datos no se presentaba ningún mensaje de ayuda, por lo que su facilidad puede determinarse como buena.
- **Economía:** Al implementarse el sistema se ahorraron recursos ya que por estar bajo un ambiente carácter, no hubo necesidad de adquirir equipo computacional, pues se siguió con el uso de terminales e impresoras existentes. Pero debido a que no se hizo un uso adecuado de índices, el tiempo de respuesta era demasiado lento y en las plantas se invirtió mucho dinero en la adquisición de procesadores y de memoria para la máquina central en la que corría ABAPRO, esto hace que este factor se considere como malo.
- **Seguridad:** El nivel de seguridad puede considerarse pésimo, ya que a la fecha, la base de datos no cuenta con ningún password de acceso y cualquiera se puede conectar conociendo la dirección lógica de la misma.

Para la evaluación de los factores de calidad extra se encontró lo siguiente:

- **Flexibilidad:** Al diseñar el sistema ABAPRO no se contempló la regresión de los procesos ya que por ejemplo no permitía la cancelación de

entradas de meses anteriores al período actual y lo que se tenía que hacer es solicitar al departamento de informática para que diera reversa a todos los procesos que implica la cancelación de la misma, por lo que la evaluación general a este punto sería de regular tendiendo a mala.

- **Reparabilidad:** El término nos dice que es la habilidad en que un sistema corrija sus errores. En este caso, es muy difícil corregir los programas ya que como no se contaba con manuales de los estándares de programación, cada desarrollador empleó su manera particular de diseñar y muchos de estos programas no cuentan con la documentación de los cambios hechos, lo que hace muy difícil el poder leerlos y darles mantenimiento. Por esto se puede considerar que este factor tiene un nivel pésimo de calidad.
  - **Adaptabilidad:** La adaptabilidad del sistema se califica como mala debido a que cuando se migró del sistema operativo VMS al sistema operativo UNIX (en la etapa de operación), se necesitó de muchos recursos y tiempo para realizar dicho cambio.
- **Facilidad de entender:** Va implícito con el factor de facilidad de uso
- **Documentación:** Se cuenta con muy pocos documentos de las especificaciones de los procesos, por lo que se deduce que se tomaron las mismas especificaciones del sistema anterior (desarrollado en Cobol) además, aunque se comenzó con el manual de usuario, nunca se ha terminado. Respecto a la documentación técnica del sistema, se vino desarrollando hasta Junio de 1999, es decir cuando el sistema ya empezaba su operación en los negocios, lo que hace que se determine este factor de calidad como malo.
- **Facilidad para conectarse a otros sistemas:** al realizarse el análisis de los sistemas con los cuales se tenía interfase (tesorería, mantenimiento y contabilidad), se tuvo mucho cuidado de que las conexiones fueran las indicadas, pero se planearon de tal manera que desde que el usuario inicia su sesión en ABAPRO se conecta a dichas interfases, aún cuando

no ejecute programas que requieran consultar o escribir en ellas. Por esto, la calidad se califica como Excelente.

Con esta información se calificaron los factores de calidad empleados en base a los criterios de "Excelente", si cumplía en su totalidad con su función principal; "Bueno", si cae en un rango de bueno o medianamente bueno; "Malo", si se cumplía de manera regular a malo y "Pésimo", si no se cumplía con la definición establecida para cada factor.

CALIDAD BASICA					CALIDAD EXTRA				
FACTOR	GRADO DE CALIDAD				FACTOR	GRADO DE CALIDAD			
	E	B	M	P		E	B	M	P
Funcionalidad			√		Flexibilidad			√	
Confiabilidad		√			Reparabilidad				√
Facilidad de Uso		√			Adaptabilidad			√	
Economía			√		Facilidad de Entender		√		
Seguridad				√	Documentación			√	
					Facilidad de Conexión	√			

Tabla 5-2 Evaluación de los Factores de Calidad de Freeman

Los resultados arrojados muestran que de los 5 los factores de Calidad Básica, 2 son malos, 1 pésimo y 2 solamente buenos, es decir, el 60% de los factores se encuentran en el rango de "malos a pésimos" y solamente el 40% se califican como "buenos"; por lo que podemos concluir que la Calidad Básica del ABAPRO se puede calificar como MALA.

En lo que respecta a los factores de Calidad Extra , solamente el factor de facilidad de conexión calificó como excelente, y el de facilidad de entender como bueno, ya que los cuatro restantes se evaluaron como malos o pésimos, es decir el 66% cae en el rango de MALA a PÉSIMA.

En la figura 5-5 se puede apreciar como en general, la calidad del sistema ABAPRO en el momento de su desarrollo fué MALA.

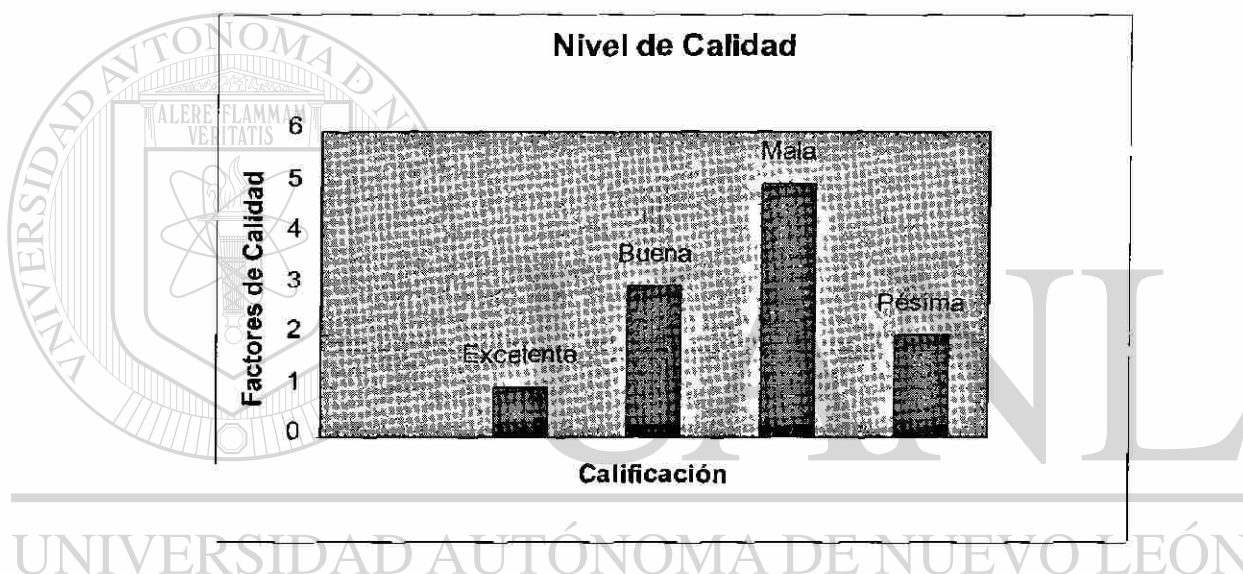


Figura 5-5 Nivel de Calidad del sistema ABAPRO en base a los Factores de Calidad de Freeman  
DIRECCIÓN GENERAL DE BIBLIOTECAS

### 5.5.3 Empleo de los Factores de Calidad de Lowell Jay Arthur para la Evaluación del ABAPRO .

Para efectos de corroborar el nivel de calidad del sistema ABAPRO en el momento de su implementación, se llevó a cabo una segunda evaluación , respondiendo a cada una de las preguntas propuestas por Lowell Jay Arthur ,

(mencionadas en el capítulo 2) donde la calificación final será un promedio de las respuestas dadas.

Para esta evaluación se siguieron los mismos criterios de calificación que en la evaluación de factores de Peter Freeman, mencionadas en el punto anterior.

Factor de Calidad	Pregunta	Resp.	Cal.
Facilidad de Mantenimiento	¿Puede arreglarse?	Si	Pésima
	¿Requiere poco esfuerzo arreglar o modificar el defecto?	Mucho	
	¿Tiene identificación en forma estándar?	No	
Flexibilidad	¿Se le pueden añadir al programa nuevos módulos sin problemas?	No	Mala
	¿Existe una copia de bibliotecas para definiciones de datos?	No	
Confiabilidad	¿Correrá y producirá los resultados correctos todo el tiempo?	No	Mala
Re-Uso	¿Puede el sistema, programa o módulo ser usado en otras aplicaciones para reducir costos de desarrollo?	Si	Buena
Facilidad de Uso	¿Puede el usuario aprender y usar el sistema fácilmente?	Si	Buena
	¿Pueden las operaciones ejecutarse?	Si	
Eficiencia	¿Corren las aplicaciones en el hardware tan rápido como es posible?	No	Mala
Facilidad de Prueba	¿Puede probarlo?	Si	Buena
Integridad	¿Son las aplicaciones y los datos seguros de cualquier intrusión?	No	Mala
Portabilidad	¿Puede moverse fácilmente de un hardware/ambiente de sistema a otro?	Mas o Menos	Buena
Interoperabilidad	¿Puede interactuar fácilmente con otros sistemas?	Si	Buena
Corrección	¿Son los datos y las aplicaciones completos, exactos y consistentes?	Mas o Menos	Mala

Tabla 5-3 Evaluación de los Factores de Calidad de Lowell Jay Arthur

De los resultados de esta evaluación se obtuvo el siguiente análisis: de los once factores propuestos, 5 calificaron como buenos, lo que representa el 45.5% de la calidad total; 5 fueron calificados como malos es decir, un 45.45% , y el 9 % restante calificó con un nivel de calidad pésimo.

Se sumaron los porcentajes de los factores que calificaron como "malos" y "pésimos", con lo que se concluyó que el 54.5% de la calidad del sistema ABAPRO en su etapa de implementación era de MALA a PESIMA .

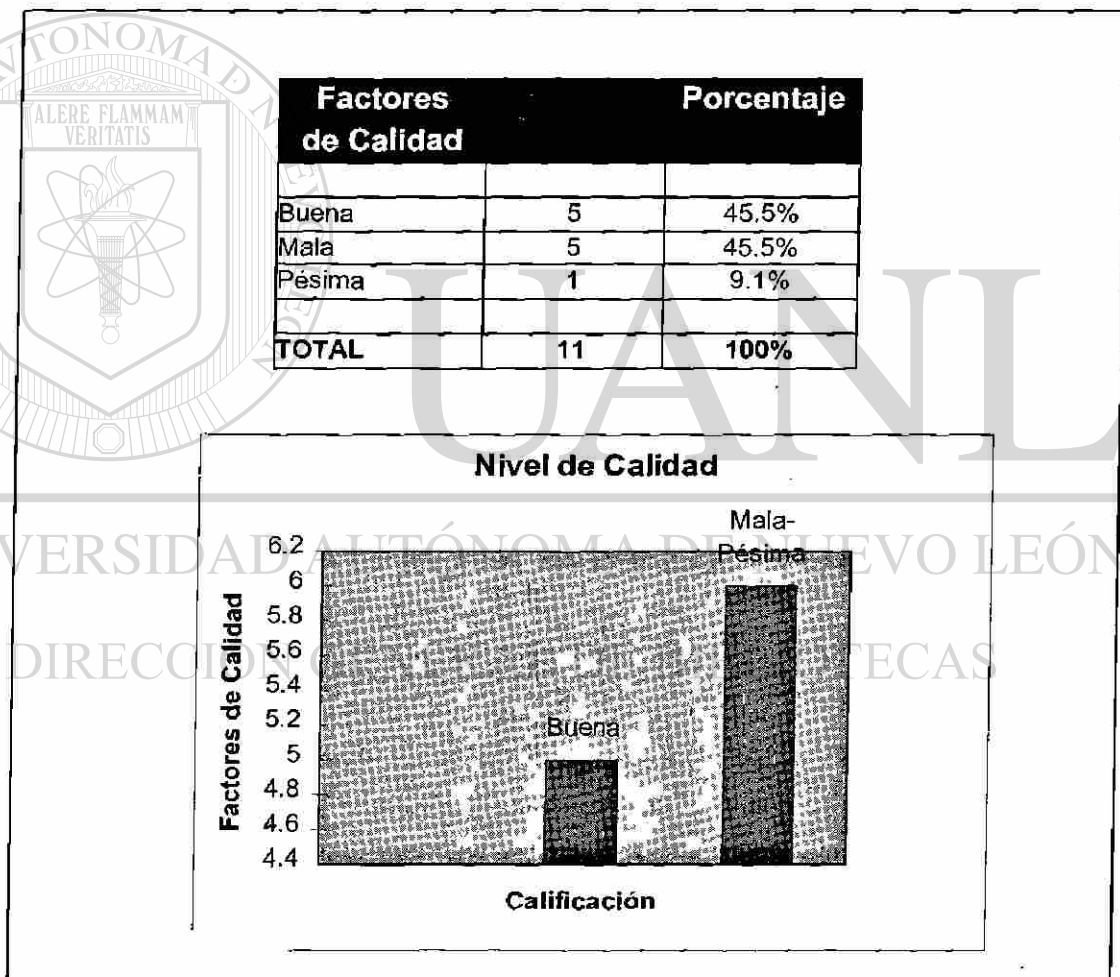


Figura 5-6 Nivel de Calidad del sistema ABAPRO en base a los Factores de Lowell Jay Arthur

Cabe mencionar que no se respetó el plan de trabajo original ya que según se muestra en la figura 5-7; la fase de diseño y construcción abarcaba el mes de Marzo de 1997 e incluía la elaboración de especificaciones, paso que se omitió ya que se tomaron las mismas bases del sistema anterior, además, la etapa de pruebas se programó para los meses de Julio y Agosto de 1997 y la realidad es que no se llevaron a cabo y se pasó directamente del desarrollo a la implementación del sistema.

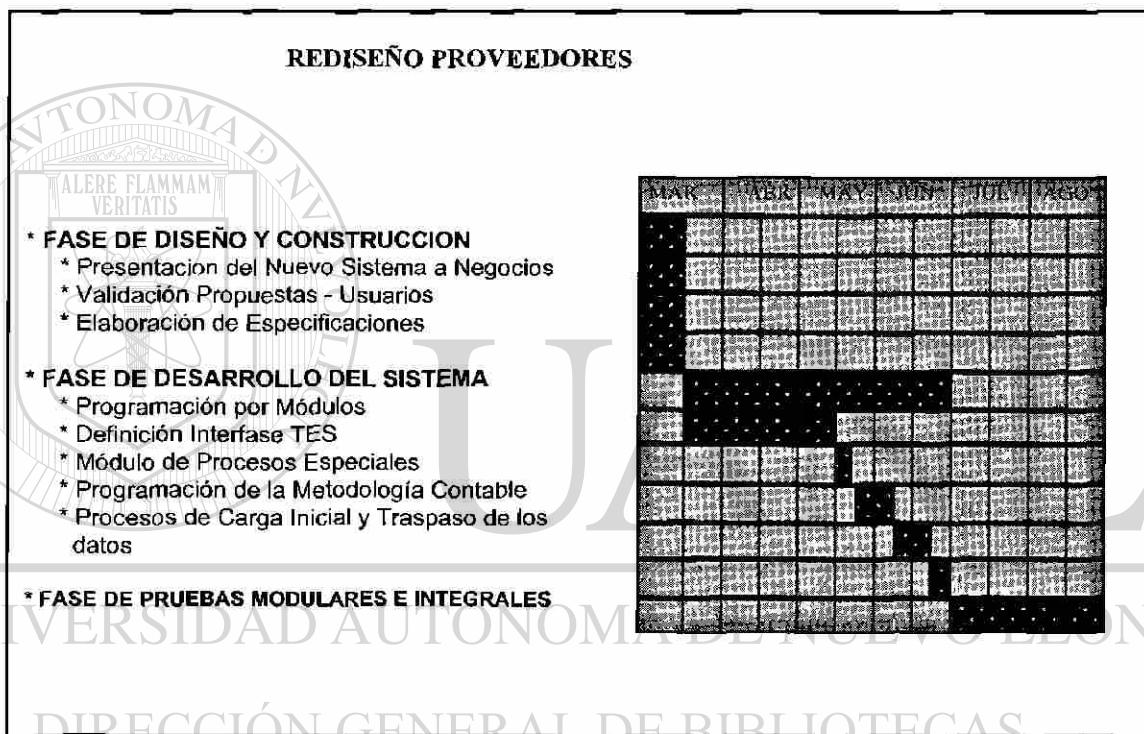


Figura 5-7 Plan de Trabajo Original del proyecto ABAPRO en el año de 1997

Recabando información en los documentos del análisis del sistema, se confirmó que se continuó con la programación secuencial, en lugar de aprovechar la programación indexada que Progress proporciona. Con lo cual aparte de que no se explota al 100% la herramienta, el tiempo de respuesta de los programas se ve deteriorada enormemente.

#### 5.5.4 Costos de la Etapa de Implementación del ABAPRO.

El período de implementación del ABAPRO en los 14 negocios de Grupo Cydsa, se llevó a cabo en el período de Junio de 1998 a Junio de 1999. Los costos originados fueron por concepto de viajes de implementación a cada planta (principalmente a Guadalajara, México, Aguascalientes y Veracruz), gastos originados por conceptos de consultorías y pagos de honorarios del personal externo (ver detalle en anexos E-F), y gastos generados por la adquisición de la infraestructura necesaria para el desarrollo y pruebas del software que se detallan a continuación (para más información ver anexo G):

- Cambios en los equipos de redes de voz y datos (troncales, líneas digitales).
- Adquisición de un nuevo equipo central que soportara el tamaño de la Base de Datos y pudiera correr las aplicaciones de manera óptima.
- Cambios en las Redes Locales (Infraestructura de la red, procesadores, licencias, PC's).

Concepto de Gasto	Monto \$	Divisa	Periodo Inicial	Periodo Final
Viáticos	946,680.00	MN	Jun-98	Jun-99
Honorarios	1,058,461.80	MN	Jun-98	Dic-98
Honorarios	1,495,933.70	MN	Ene-99	Jun-99
<b>**Cambio de Infraestructura</b>	<b>1,681,788.00</b>	<b>USD</b>	<b>Ene-98</b>	<b>*****</b>
<b>TOTAL(MN)</b>	<b>3,501,075.50</b>			

Tabla 5-4 Gastos generados durante la implementación del sistema ABAPRO



**Nota:** Los costos por el cambio de infraestructura son solamente informativos ya que fueron cubiertos por cada negocio.

La primera planta donde se implementó el ABAPRO fué en la planta del ISTMO, en Coatzacoalcos Veracruz ; una vez implementado ahí, se siguió con la plantas de Hilaturas y Ryltex, localizadas en Aguascalientes, después se implementó en Plásticos Rex (la matriz esta localizada en el Distrito Federal y una sucursal en Poncitlán, Jalisco) al mismo tiempo que otro equipo de trabajo implementaba en las plantas de Rayón y Quimobásicos en el complejo de Ruiz Cortines.

La siguiente etapa de implementación se dividió en otros cuatro grupos, uno que cubrió la planta de Crysel, otro la planta de Ultracril (ambas localizadas en la ciudad de Guadalajara) y los otros dos en las planta de Masterpak y Atlatec, ubicadas en el complejo de Ruiz Cortines.

La última etapa de implementación se llevó a cabo en el Corporativo de Grupo Cydsa y en el Club Cima ambos edificios localizados en el área metropolitana.

Algunos factores que contribuyeron a que se elevaran tanto los gastos en esta etapa son los siguientes:

- Debido a que cada planta tiene una forma distinta de llevar a cabo sus procesos de negocio, se tuvieron que personalizar muchas aplicaciones, lo que ocasionó que se contratara muchos recursos .
- No se tuvo el cuidado de llevar el proceso de pruebas, ya que de la etapa de diseño (Mayo de 1998) se pasó directamente a la etapa de implementación (Junio de 1998) , esto repercutió sobre todo en el módulo

contable que prácticamente se vino desarrollando en la implementación del ISTMO con personal externo de honorarios.

- Los viajes a los negocios se realizaban sin justificación aparente, anteponiendo motivos personales a los laborales.
- Debido a que el sistema presentaba un pésimo tiempo de respuesta, se determinó que era por el CPU de la máquina central, por lo que se adquirió nueva infraestructura, aún cuando la causa principal era lo mal estructurado del sistema (mal uso de índices y de programación).

### 5.5.5 Costos del Mantenimiento del Sistema ABAPRO.

Los costos de mantenimiento del sistema ABAPRO se incrementaron principalmente porque no se tuvo un buen análisis del mismo. Debido a que la mayoría de los módulos no cumplían con las necesidades de cada usuario, se tuvo que invertir mucho presupuesto en un nuevo análisis y diseño de cada una de las aplicaciones. La tabla 5-5 muestra estos costos.

Concepto de Gasto	Monto \$	Divisa	Periodo Inicial	Periodo Final
Viáticos	300,000.00	MN	Jul-99	Dic-00
Honorarios	1,711,492.60	MN	Jul-99	Dic-99
Honorarios	1,227,700.50	MN	Ene-00	Dic-00
<b>TOTAL(MN)</b>	<b>3,239,193.10</b>			

Tabla 5-5 Gastos generados por el mantenimiento al sistema ABAPRO

El detalle de estos datos se puede apreciar consultando los anexos F y H.

### 5.5.6 Cuadro Comparativo de Costos.

En el capítulo 3, Summerville menciona que los costos de mantenimiento varían de aplicación en aplicación y que en promedio parecen ser de 2 a 4 veces los costos de la etapa de desarrollo, y que a medida que el software se va utilizando se requiere un esfuerzo mayor para mantenerlo.

En la figura 5-8, se presenta un resumen de los costos generados en las etapas de desarrollo, implementación y mantenimiento del ABAPRO y la distribución en porcentajes de estos costos.

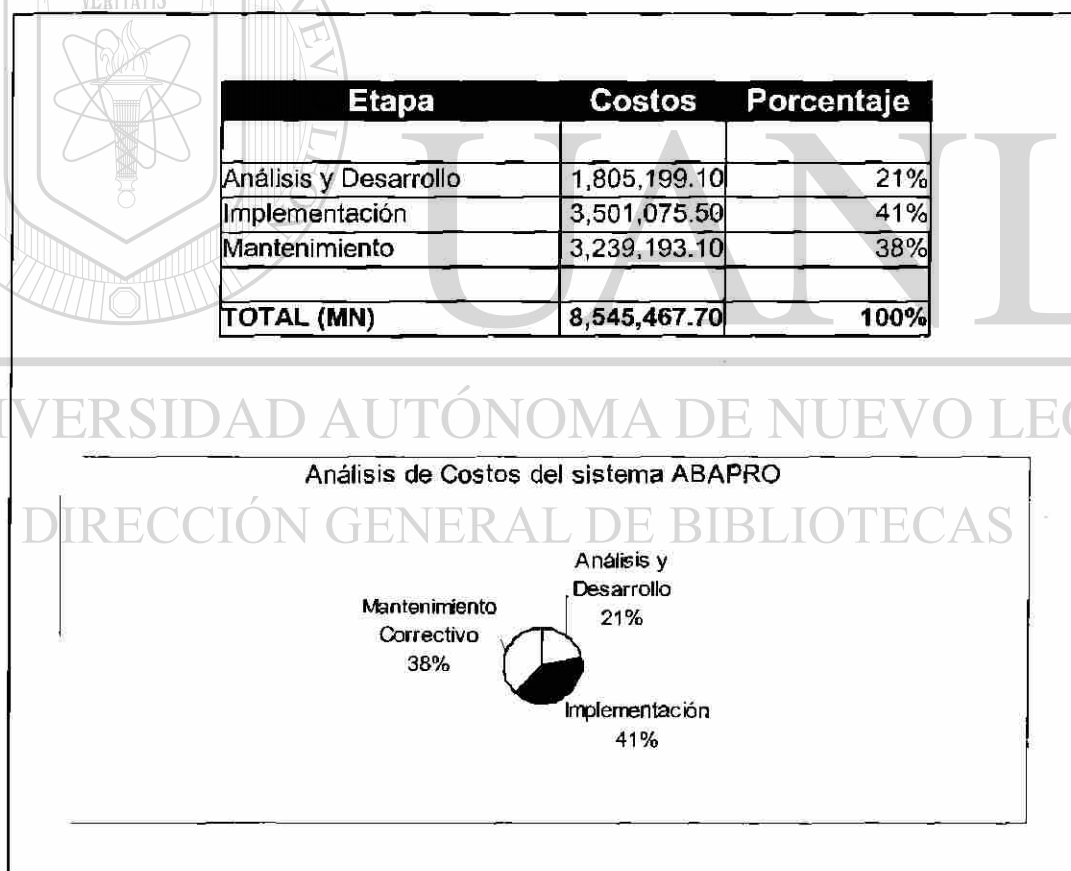


Figura 5-8 Distribución de los Costos del ABAPRO

Donde se puede apreciar que los costos de la implementación y mantenimiento del sistema conformaron un 79% de los costos totales.

Dividiendo el porcentaje de los costos del mantenimiento correctivo entre el porcentaje de los costos de desarrollo ( 38/21) se llegó a la conclusión de que los costos de mantenimiento representan 1.8 veces los costos de la etapa de desarrollo.

Estos costos se representan con el formato de la figura 3-2, para comprobar la teoría de Summerville que indica que un esfuerzo extra en el desarrollo representa un multiplicador negativo en los costos de mantenimiento, ya que en este caso se invirtió poco presupuesto en realizar un buen análisis y esto repercutió en los costos de implementación y mantenimiento como se observa en la figura 5-9.

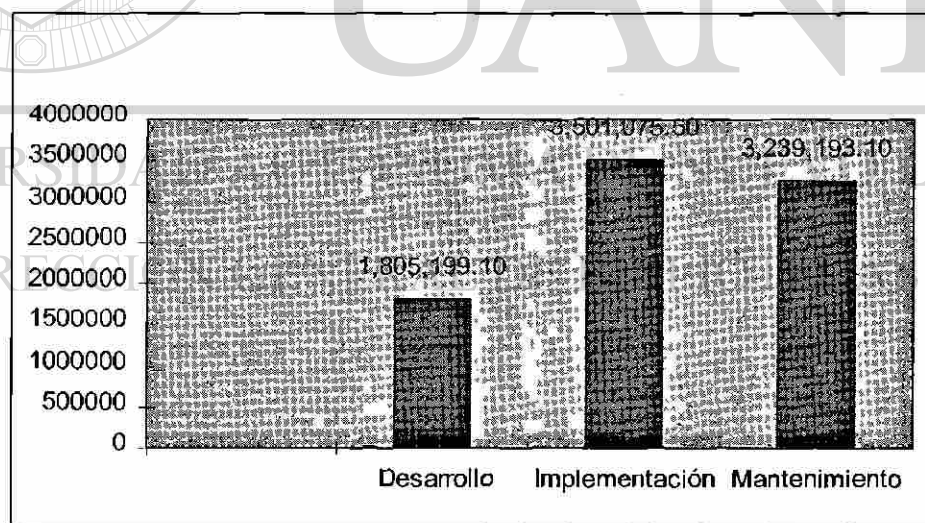


Figura 5-9 Comparación de los Costos de Desarrollo, Implementación y Mantenimiento

## 5.6 Estadística Descriptiva.

Mediante el uso de la estadística descriptiva se mostrarán algunos aspectos relevantes alrededor de la omisión de factores de calidad en el desarrollo del software como resultado de la recopilación de datos obtenidos en las solicitudes de servicio colocadas por los usuarios del sistema de ABAPRO en el período de Septiembre a Diciembre de 1999. Estos datos pueden ser apreciados en el anexo I .

### 5.6.1 Distribución del Negocio.

En las solicitudes de servicio por negocio se obtuvo la distribución que se observa en la figura 5-10, donde se puede apreciar que el negocio que más solicitudes colocó fue el Istmo , con el 36% de las solicitudes totales, siendo en este negocio donde se implementó el sistema ABAPRO por primera vez; el porcentaje de los demás negocios fue disminuyendo a medida que se iban corrigiendo las aplicaciones y se implementaba un software más pulido.

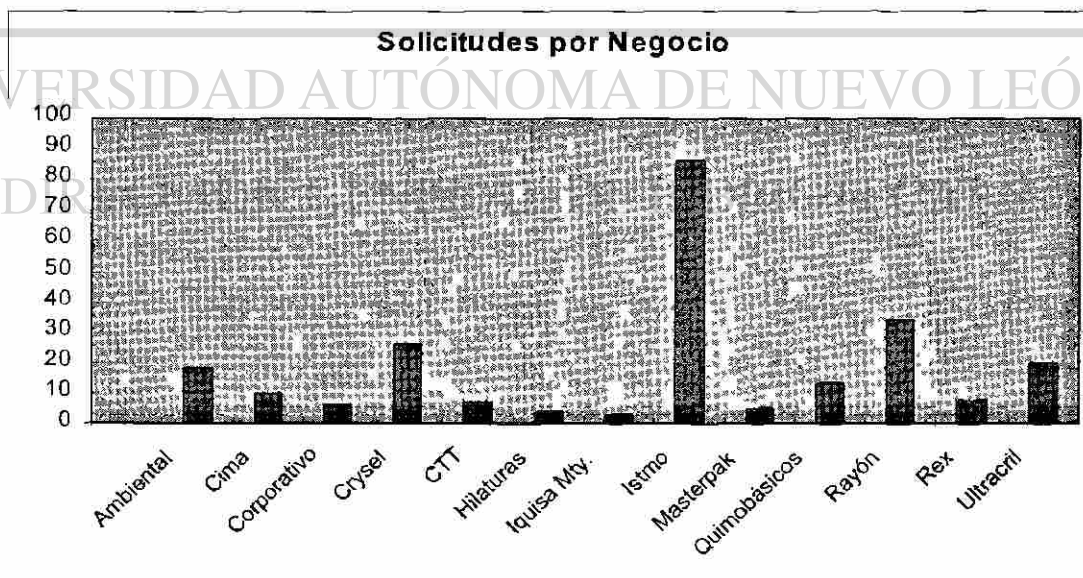


Figura 5-10 Distribución de las solicitudes en base a los Negocios de Grupo Cydsa S.A. de C.V.

### 5.6.2 Distribución por el Tipo de Servicio.

Se clasificaron los tipos de servicio de acuerdo a los siguientes criterios:

- Bloqueo. Se refiere a aquellas solicitudes en las que se reportan problemas de bloqueo entre archivos .
- Capacitación. Solicitudes en las cuales se pide un plan de capacitación en el manejo del sistema
- Config. Son las solicitudes en las que fué necesario configurar algún módulo, agregar ó eliminar alguna opción de menú, cambio de datos o clasificaciones, así como mantenimientos por parte del administrador del sistema.
- Depuración. Son las solicitudes en las que se pide la depuración de algunos directorios del sistema, archivos o información obsoleta.
- Duda . Son aquellas solicitudes que se colocaron para cuestionar sobre la manera de operar el sistema.
- Falla Acceso . Son los problemas para acceder la base de datos.
- Falla Cons . Son solicitudes que reportan fallas en las consultas .
- Falla Contab. Se refiere a solicitudes que señalan problemas en la contabilidad de los datos; ya sea en los procesos de descarga de pólizas o en problemas de conciliación de la información del ABAPRO contra los reportes del sistema de contabilidad (SIF).
- Falla Proc . Se refiere a solicitudes que reportan fallas en los procesos principales del sistema.
- Falla Rep . Son fallas en reportes como inconsistencias en la información, presentación de valores en ceros, errores en el formato de los datos, inestabilidad de la información, ect.
- Gen. Inf. Son aquellas en las que se solicitan reportes especiales.

- Interfase. Son solicitudes que reportan fallas con las interfaces de los sistemas con los cuales interactúa el sistema ABAPRO.
- NA . Son solicitudes que no aplican.
- Nvo Des. Se refiere a solicitudes de nuevos desarrollos.
- Personaliz.Pta . Son personalizaciones del negocio.
- Recompilación . Son solicitudes que reportan fallas en las aplicaciones por la falta de compilación de algún programa.
- Reproceso. Solicitudes en las que se pide el reproceso de información (principalmente reportes) que en su momento no se genero correctamente.
- Tiempo Resp. Son problemas relacionados con el tiempo de respuesta de las aplicaciones.
- Validaciones . Son solicitudes de validaciones a los procesos .

Es preciso mencionar que esta clasificación es el resultado del análisis detallado de la descripción de cada uno de los servicios solicitados en cada una de las solicitudes en cuestión.

Con la finalidad de mostrar estadísticamente la distribución de las solicitudes por el tipo servicio proporcionado se elaboró la tabla 5-6 ; en la cual se observa para cada tipo de servicio el porcentaje que representa cada servicio en el total de las solicitudes.

Como resultado se tiene que el 31.7% de los servicios proporcionados fueron por Fallas en los Procesos del sistema, siguiéndole un 16.3% por concepto de Configuraciones, las Personalizaciones del Negocio con el 12.1% , fallas en la contabilidad con el 8.3 % y un 5.8% en fallas de reportes como los pesos más significativos.

Esto indica que el 74.2% de las solicitudes se abocaron en fallas , configuraciones y personalizaciones .

En contraparte, los porcentajes más bajos lo mostraron los servicios tipo " Help Desk" , con un 2.9% en promedio; lo que permite deducir que el sistema de solicitudes esta más inclinado al área de desarrollo y no al de atención a usuarios como fué planeado en un principio. Otro dato importante es el de la Capacitación con un .4% ; lo que indica que el sistema en general es amigable .

Tipo de Servicio	Número Solicitudes	Porcentaje
Falla Proc	76	31.7%
Config.	39	16.3%
Personaliz.Pta	29	12.1%
Falla Contab.	20	8.3%
Falla Rep	14	5.8%
Depuración	7	2.9%
Duda	7	2.9%
Falla Acceso	7	2.9%
Nvo. Des.	7	2.9%
Reproceso	6	2.5%
Validación	6	2.5%
Tiempo Resp.	5	2.1%
Bloqueo	4	1.7%
Recompilación	4	1.7%
Interfase	3	1.3%
Falla Cons.	2	0.8%
Gen. Inf.	2	0.8%
Capacitación	1	0.4%
NA	1	0.4%
<b>TOTAL</b>	<b>240</b>	<b>100%</b>

**Tabla 5-6 Solicitudes recibidas por Tipo de Servicio**



### 5.6.3 Distribución por Enfoque Informático.

Para apreciar estos datos de manera más concisa, se realizó una nueva clasificación de los tipos de servicios: los de atención a usuarios y los de fallas. En la subdivisión de atención a usuarios se incluyeron los tipos de servicios que no representan fallas en el sistema, sino que se inclinan por lo que se denominan “obligaciones” o “funciones” de todo departamento de sistemas; como lo son configuraciones, depuraciones, fallas en accesos, generación de información, desarrollos nuevos, personalizaciones, recompilaciones y reprocesos.

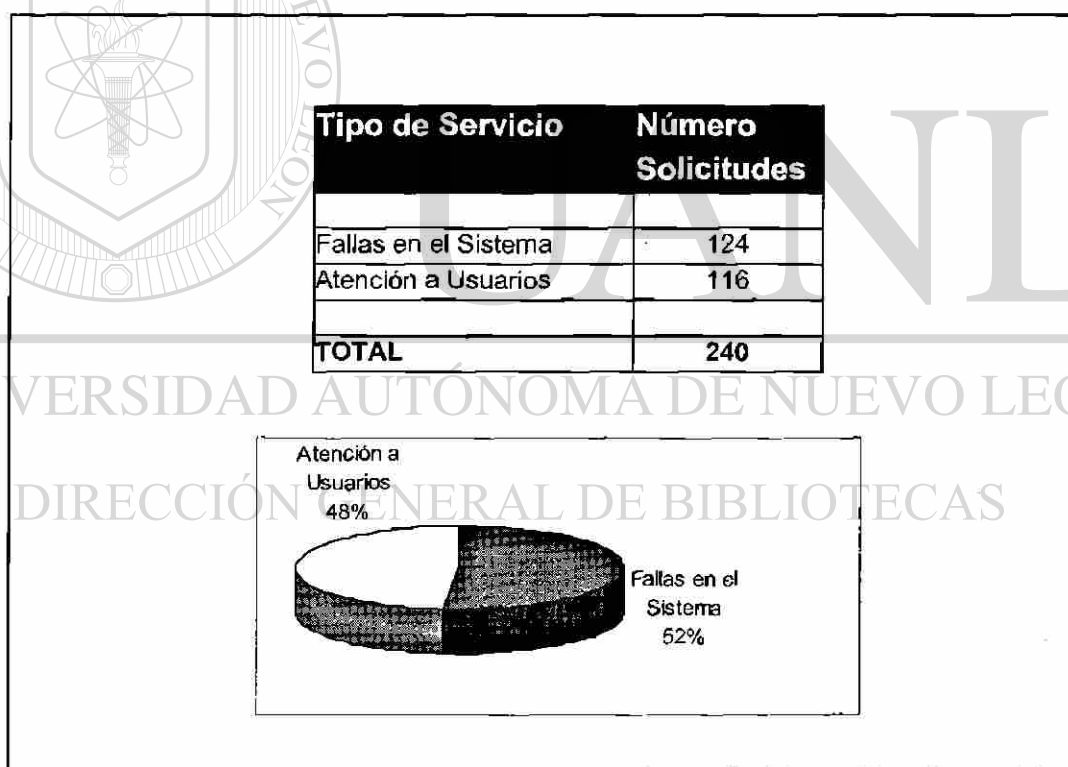
La otra subdivisión que se hizo corresponde a las solicitudes que reportan fallas en las aplicaciones del sistema, tales como las fallas en los procesos, consultas, reportes y en la contabilidad de la información; problemas con las interfases, degradación en el tiempo de respuesta de las aplicaciones y problemas de concurrencia de archivos.

Servicios de Atención a Usuarios	Servicios por Fallas del Sistema
Configuraciones Depuraciones Dudas Problemas de Acceso Generación de Inf. Nuevos desarrollos Personalizaciones Recompilaciones Reprocesos Capacitaciones	Bloqueos Fallas en Consultas Fallas en la Contabilidad Fallas en los Procesos Fallas en los Reportes Problemas con las Interfases Problemas de tiempo de respuesta

Tabla 5-7 Reclasificación de Tipos de Servicio

Analizando la información de esta nueva clasificación se obtuvo que el 52% de las solicitudes de servicio estuvieron enfocadas a la resolución de fallas, mientras que el 48% restante se inclinó hacia las solicitudes de atención a usuarios, como se ilustra en la Figura 5-11.

Un punto a destacar es que aunque las solicitudes de personalizaciones de la planta se clasificaron como "atención a usuarios", resultan ambiguas ya que también se pudieron haber clasificado como fallas debido a que debieron haberse realizado en la etapa de implementación, no en la operación del sistema, o al menos no en un porcentaje tan alto.



**Figura 5-11** Distribución de la Solicitudes en base al Enfoque del Departamento de Informática

#### 5.6.4 Análisis de los Mantenimientos a Programas.

Para efectos de determinar el peso que ejercieron los mantenimientos a programas en la distribución de las solicitudes, se hizo esta categoría, donde se incluirían todas aquellas modificaciones que se hubiesen llevado a cabo en los programas. Esto abarca los siguientes conceptos:

- Modificaciones por fallas en programas de procesos, consultas y reportes.
- Modificaciones por personalizaciones.
- Modificaciones hechas para arreglar problemas de concurrencia y de tiempo de respuesta.
- Validaciones .

Con lo que se obtuvo la Figura 5-12, donde se observa claramente que el 66% de las solicitudes propiciaron mantenimientos a programas.

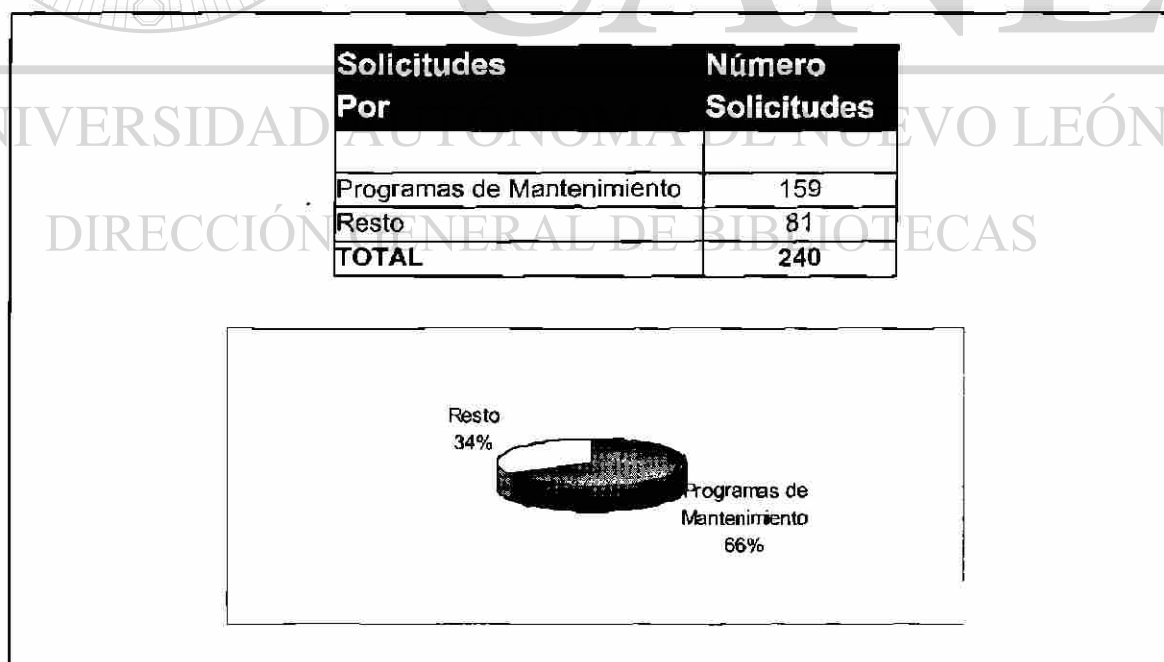


Figura 5-12 Distribución de Solicitudes por Mantenimiento

### 5.6.5 Distribución de la Criticalidad de las Solicitudes.

El sistema de solicitudes de servicio las clasifica de acuerdo a su prioridad, la cual puede ser Crítica, Indispensable o Deseable. Se obtuvo la distribución que se observa en la figura 5-13, donde los resultados mostraron que de las 240 Solicitudes, 110 fueron Críticas, lo que representa el 46% del global de solicitudes; 96 se colocaron como Indispensables, es decir el 40% del total; y el 14% restante lo conformaron las solicitudes marcadas como Deseables.

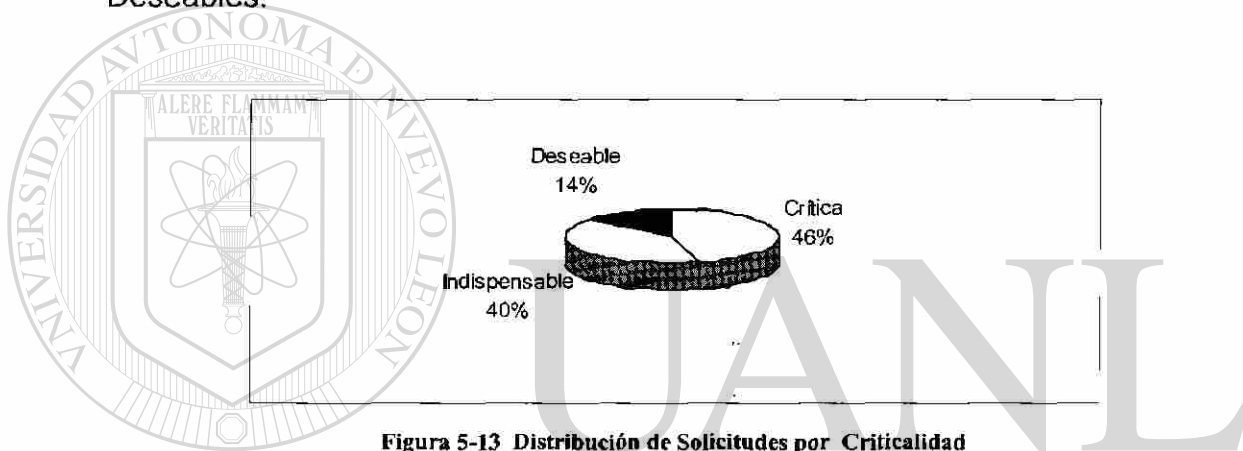


Figura 5-13 Distribución de Solicitudes por Criticalidad

Esto permite concluir que el nivel de calidad del sistema ABAPRO era muy bajo, ya que el 86% de las solicitudes alimentadas fueron Críticas ó Indispensables para la operación del sistema.

### 5.6.6 Calificación de las Solicitudes de Servicio .

Con respecto a las calificaciones que se otorgaron a los servicios proporcionados, los usuarios emplearon los criterios de Excelente, Bueno, Regular y Malo. En general, se puede decir que el servicio proporcionado fue bueno, como lo demuestra el 79.17% de la figura 5-14.

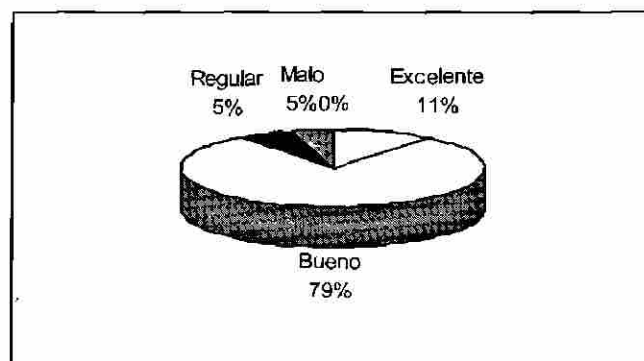


Figura 5-14 Calificación del Servicio Proporcionado

### 5.7 Conclusiones Generales.

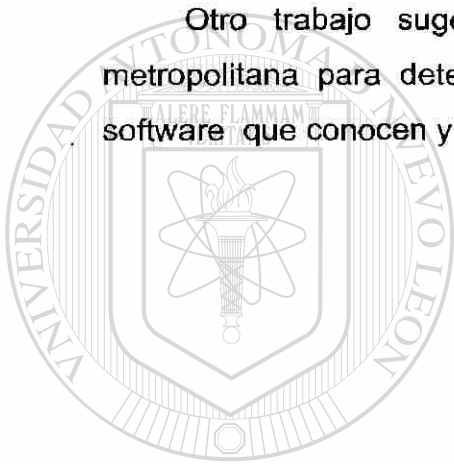
El resultado global del desarrollo de esta tesis ha sido positivo, ya que se ha logrado la comprobación de que la falta de calidad en los sistemas en el momento de su desarrollo impacta enormemente en las etapas posteriores de implementación y de operación; no solo en cuanto a costos, sino también en el nivel de calidad desde la perspectiva del usuario, quien finalmente determina si un sistema es funcional o no.

Se llegó a la conclusión de que la calidad para el usuario significa que sus requerimientos de información y de proceso sean cumplidos en el mayor grado posible; además de que un sistema con calidad es aquel que no requiere de un continuo mantenimiento por correcciones una vez que ha sido puesto en marcha. Aún queda mucho por hacer, este solo es un avance para lograr lo que todo analista y desarrollador de sistemas desea producir, un sistema de información con calidad.

## 5.8 Trabajos Futuros.

La inclusión del capítulo de métricas en este documento se incluyó como parte del esfuerzo que se debe realizar para impulsar esta área desatendida e importante para el desarrollo de software, por lo que sería interesante realizar el desarrollo de dos proyectos en paralelo, empleando y no empleando métricas de calidad; para posteriormente evaluar su calidad y hacer el análisis de los costos de cada uno.

Otro trabajo sugerido sería el de aplicar un estudio en el área metropolitana para determinar el porcentaje de empresas constructoras de software que conocen y utilizan las técnicas del SQA en sus productos.



# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



## REFERENCIAS BIBLIOGRAFICAS.

Cristales Pavón, Asley Alberto ; "Estudio Exploratorio de la Aplicación de Actividades de SQA en el Proceso de desarrollo de Software para lograr el Exito de los Sistemas"; Tesis de Maestría, I.T.E.S.M.,Monterrey N.L; 1996.

Crosby , Philip B. ; La Calidad No Cuesta, El Arte de Cerciorarse de la Calidad; Compañía Editorial Continental;México;1991.

De Millo, McCracken;Software Testing and Evaluation;Benjamin Cummings; USA ;1987.

Freeman, Peter; Software Perspectives: The System Is the Message; Addison Wesley; USA;1987.

Gómez Serrato, Eduardo ; "Métricas de Calidad y Productividad en la Función de Codificación de Software"; Tesis de Maestría, I.T.E.S.M., Monterrey N.L; 1995.

Jorgensen,Paul C. ; Software Testing A Craftsman's Approach; CRC;USA;1995.

Juran, J.M ; Quality Planning and Analysis; Mc Graw Hill; U.S.A ;1970.

Lowell ,Jay Arthur; Software Evolution: The Software Maintenance Challenge; Wiley- Interscience Publication;USA; 1988.

Pressman, Roger S.; Software Engineering: A Practitioner's Approach; Mc Graw Hill;USA; 1992.

Schulmeyer, Mc Manus; Handbook of Software Quality Assurance; VNR, U.S.A; 1987.

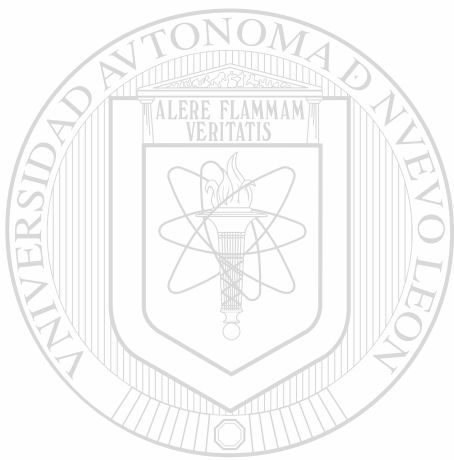
Sommerville, Ian; Software Engineering; Addison- Wesley Publishing Company ;USA;1989.

Stebbing, Lionel; Aseguramiento de la Calidad, El Camino a la Eficiencia y la Competitividad; Compañía Editorial Continental;México; 1991.

Uvalle del Castillo,Nancy; "Control de Calidad en Sistemas de Información"; Tesis de Maestría, I.T.E.S.M.; Monterrey N.L; 1991.

Villarreal Antelo, Brenda María; "Aseguramiento de la Calidad de Sistemas de Información"; Tesis de Maestría, I.T.E.S.M.; Monterrey N.L; 1995.

Wengerg, Gerald M.; Quality Software Management : Volume 1 Systems Thinking; Dorset House Publishing;USA; 1991.



UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS



## LISTA DE FIGURAS.

<b>Figura</b>	<b>Título</b>	<b>Página</b>
2-1	Generación de Software de Calidad	18
2-2	Clasificación de Métricas de Calidad según Roger Pressman	23
2-3	Factores de Calidad de Software de McCall	30
2-4	Flujo de Información de Pruebas	41
3-1	Distribución de los Tipos de Mantenimiento	49
3-2	Costos de Desarrollo y Mantenimiento	50
4-1	Modelo Propuesto de la relación de las Métricas de Calidad con el Éxito de los Sistemas	59
5-1	Organigrama del Departamento de Informática	62
5-2	Servidores de Uso Interno	64
5-3	Servidores Con Conexión Externa	65
5-4	Esquema General del Sistema ABAPRO	67
5-5	Nivel de Calidad del sistema ABAPRO en base a los Factores de Calidad de Freeman	73
5-6	Nivel de Calidad del sistema ABAPRO en base a los Factores de Calidad de Lowell Jay Arthur	75
5-7	Plan de Trabajo Original del proyecto ABAPRO en el año de 1997	76
5-8	Distribución de los Costos del ABAPRO	80
5-9	Comparación de los Costos de Desarrollo, Implementación y Mantenimiento	81
5-10	Distribución de las solicitudes en base a los Negocios de Grupo Cydsa S.A. de C.V.	82
5-11	Distribución de la Solicitudes en base al Enfoque del Departamento de Informática	87
5-12	Distribución de Solicitudes por Mantenimiento	88
5-13	Distribución de Solicitudes por Criticalidad	89
5-14	Calificación del Servicio Proporcionado	90

## LISTA DE TABLAS.

Tabla	Título	Página
2-1	Métricas Orientadas al Tamaño	25
2-2	Métricas de Puntos de Funcionalidad	26
2-3	Puntos de Función Computacionales	27
2-4	La Relación entre los Factores y Métricas de Calidad de McCall	32
2-5	Clasificación de los Factores de Calidad según Freeman	34
3-1	Costos Estimados en el desarrollo del software	46
5-1	Gastos generados durante el análisis y diseño del sistema ABAPRO	68
5-2	Evaluación de los Factores de Calidad de Freeman	72
5-3	Evaluación de los Factores de Calidad de Lowell Jay Arthur	74
5-4	Gastos generados durante la implementación del sistema ABAPRO	77
5-5	Gastos generados por el mantenimiento al sistema ABAPRO	79
5-6	Solicitudes recibidas por Tipo de Servicio	85
5-7	Reclasificación de Tipos de Servicio	86

DIRECCIÓN GENERAL DE BIBLIOTECAS

## LISTA DE ABREVIATURAS.

ABAPRO. Abastecimientos y Proveedores.

Cn. Coeficiente de Regresión.

CPU. Central Process Unit (Unidad Central de Procesamiento).

C.V. Capital Variable.

Cydsa. Celulosa y Derivados, Sociedad Anónima.

EDP. Electronic Data Processing (Procesamiento Electrónico de Datos).

Etc. Etcétera.

Fc. Factor de Calidad.

FPs. Puntos de Fusión.

HP. Hewlett Packard.

IBM. International Business Machine.

KLOC. Thousand Lines of Code (Miles de Líneas de Código)

LOC. Lines of Code (Líneas de Código).

MC. Métricas de Calidad.

MN. Moneda Nacional.

PC. Personal Computer (Computadora Personal).

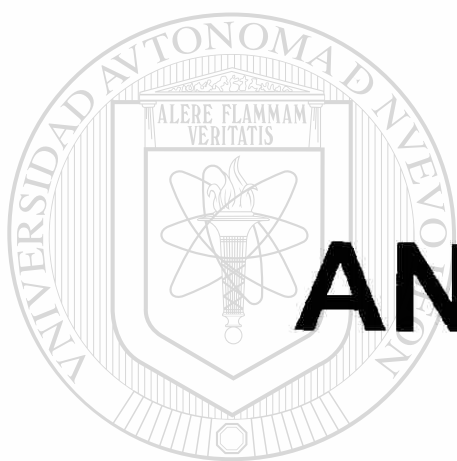
S.A. Sociedad Anónima.

SI. Sistemas de Información.

SIF. Sistema de Información Financiera

SQA. Software Quality Assurance (Aseguramiento de la Calidad del Software)

SQAM. Software Quality Assessment and Measurement (Aseguramiento y Medición de la Calidad del Software).



# ANEXOS

UANL

---

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

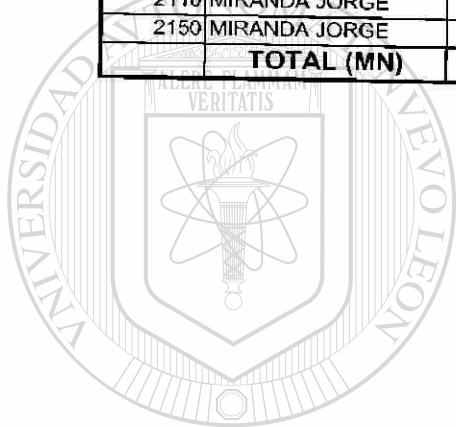
## ANEXO A

### Gastos Generados por Concepto de Viajes durante el periodo Ene-Dic 1997.

VIAJE	NOMBRE	GASTO	SALIDA	LLEGADA	DIAS	VIAJE
38	AVILA JOSE	4,211.53	05-Mar-97	08-Mar-97	3	GUAD. JAL. -
66	HDZ. NIÑO GERARDO	3,941.31	06-Mar-97	07-Mar-97	1	AGS. -
150	MIRANDA ABELARDO	4,897.55	11-Mar-97	14-Mar-97	3	MEX., D.F. -
165	HDZ. NIÑO GERARDO	2,745.30	12-Mar-97	14-Mar-97	2	MEX., D.F. -
167	AVILA JOSE	3,778.66	12-Mar-97	15-Mar-97	3	GUAD. JAL. -
193	MIRANDA ABELARDO	4,104.43	17-Mar-97	19-Mar-97	2	TAMPICO -
196	HDZ. NIÑO GERARDO	3,525.37	17-Mar-97	19-Mar-97	2	TAMPICO -
249	AVILA JOSE	7,003.22	19-Mar-97	26-Mar-97	7	GUAD. JAL. -
327	AVILA JOSE	3,448.49	02-Abr-97	04-Abr-97	2	GUAD. JAL. -
517	MIRANDA ABELARDO	4,676.45	16-Abr-97	18-Abr-97	2	TAMPICO -
540	AVILA JOSE	3,232.21	17-Abr-97	19-Abr-97	2	GUAD. JAL. -
585	AVILA JOSE	4,718.58	22-Abr-97	26-Abr-97	4	GUAD. JAL. -
608	HDZ. NIÑO GERARDO	4,155.11	23-Abr-97	25-Abr-97	2	MEX., D.F. -
703	MIRANDA JORGE	2,206.46	29-Abr-97	29-Abr-97	-	GUAD. JAL. -
743	MIRANDA JORGE	2,440.85	02-May-97	02-May-97	-	GUAD. JAL. -
755	AVILA JOSE	2,638.22	02-May-97	03-May-97	1	GUAD. JAL. -
797	AVILA JOSE	4,569.39	06-May-97	09-May-97	3	TAMPICO -
828	MIRANDA ABELARDO	3,868.04	07-May-97	09-May-97	2	TAMPICO -
898	AVILA JOSE	7,083.75	13-May-97	16-May-97	3	TAMPICO -
917	MIRANDA ABELARDO	4,159.19	14-May-97	16-May-97	2	TAMPICO -
948	VILLEGAS MATA JOSE	2,467.24	16-May-97	16-May-97	-	GUAD. JAL. -
993	MIRANDA ABELARDO	4,810.51	20-May-97	23-May-97	3	TAMPICO -
1022	AVILA JOSE	4,209.37	21-May-97	23-May-97	2	TAMPICO -
1049	VILLEGAS MATA JOSE	11,467.41	22-May-97	04-Jun-97	13	GUAD. JAL. -
1068	MIRANDA ABELARDO	3,993.77	27-May-97	30-May-97	3	TAMPICO -
1069	AVILA JOSE	8,084.64	27-May-97	06-Jun-97	10	TAMPICO -
1070	MIRANDA JORGE	5,213.19	27-May-97	30-May-97	3	TAMPICO -
1098	HDZ. NIÑO GERARDO	2,767.90	23-May-97	23-May-97	-	GUAD. JAL. -
1187	MIRANDA ABELARDO	8,203.14	02-Jun-97	09-Jun-97	7	TAMPICO -
1188	MIRANDA JORGE	10,014.16	02-Jun-97	13-Jun-97	11	TAMPICO -
1324	MIRANDA ABELARDO	3,775.24	11-Jun-97	14-Jun-97	3	TAMPICO -
1326	AVILA JOSE	4,136.04	11-Jun-97	14-Jun-97	3	TAMPICO -
1382	AVILA JOSE	5,942.56	17-Jun-97	20-Jun-97	3	TAMPICO -
1388	MIRANDA ABELARDO	4,142.79	17-Jun-97	19-Jun-97	2	TAMPICO -
1390	MIRANDA JORGE	5,030.66	17-Jun-97	20-Jun-97	3	TAMPICO -
1477	TREVIÑO IGNACIO	2,421.74	23-Jun-97	23-Jun-97	-	MEX., D.F. -
1479	MIRANDA ABELARDO	2,508.74	23-Jun-97	23-Jun-97	-	MEX., D.F. -
1524	AVILA JOSE	8,132.83	25-Jun-97	04-Jul-97	9	TAMPICO -
1526	MIRANDA JORGE	9,146.37	25-Jun-97	04-Jul-97	9	TAMPICO -
1659	MIRANDA JORGE	3,847.53	07-Jul-97	08-Jul-97	1	TAMPICO -
1663	TREVIÑO IGNACIO	4,414.16	07-Jul-97	09-Jul-97	2	AGS. -
1681	TREVIÑO IGNACIO	3,499.80	03-Jul-97	04-Jul-97	1	MEX., D.F. -
1739	MIRANDA ABELARDO	3,248.91	10-Jul-97	11-Jul-97	1	MEX., D.F. -
1740	AVILA JOSE	5,963.82	08-Jul-97	11-Jul-97	3	GUAD. JAL. -

## (Continuación ANEXO A)

VIAJE	NOMBRE	GASTO	SALIDA	LLEGADA	DIAS	VIAJE
1789	MIRANDA ABELARDO	4,911.51	15-Jul-97	18-Jul-97	3	MEX., D.F. -
1822	MIRANDA JORGE	3,739.53	17-Jul-97	18-Jul-97	1	MEX., D.F. -
1823	AVILA JOSE	5,994.13	15-Jul-97	18-Jul-97	3	GUAD. JAL. -
1853	MIRANDA ABELARDO	4,393.74	21-Jul-97	22-Jul-97	1	AGS. -
1963	MIRANDA JORGE	4,831.84	28-Jul-97	31-Jul-97	3	MEX., D.F. -
1975	TREVIÑO IGNACIO	4,128.79	30-Jul-97	01-Ago-97	2	MEX., D.F. -
2110	MIRANDA JORGE	4,852.77	07-Ago-97	10-Ago-97	3	MEX., D.F. -
2150	MIRANDA JORGE	90.00	24-Jul-97	24-Jul-97	-	MEX., D.F. -
<b>TOTAL (MN)</b>		<b>241,788.94</b>				



# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



## ANEXO B

### Gastos Generados por Concepto de Honorarios durante el periodo Sep-Dic 1996.

Nombre	Puesto	Sep	Oct	Nov	Dic	TOTAL 1996
Abelardo Miranda	LDP	9,146	11,890	11,890	11,890	44,816
Luis Enrique Santoy	LDP	9,708	9,708	9,708	9,708	38,832
Ignacio Treviño	Red	8,500	8,500	8,500	8,500	34,000
Héctor Gzz. Garza	Red	0	0	9,700	9,700	19,400
Jaime González	Red	9,000	9,000	9,000	9,000	36,000
Luis Manuel Gzz	Red	8,500	8,500	0	0	17,000
Mario Saucedo	Red	0	0	9,680	9,680	19,360
				<b>TOTAL (MN)</b>		<b>209,408</b>

LDP = Lider de Proyecto

Red = Rediseño

# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

®

## ANEXO C

### Gastos Generados por Concepto de Honorarios durante el periodo Ene-Dic 1997.

Nombre	Puesto	Ene	Feb	Mzo	Abr	Mary	Jun	TOTAL 1997
Abelardo Miranda	LDP	11,890	11,890	11,890	14,125	14,125	14,125	78,045
Luis E. Santoy	LDP	9,708	9,708	9,708	11,533	11,533	11,533	63,723
Ignacio Treviño	Red	9,775	9,775	9,775	9,775	9,775	9,775	58,650
Héctor Gzz.	Red	9,700	9,700	9,700	9,700	10,670	10,670	60,140
Jaime González	Red	9,000	0	0	0	0	0	9,000
Mario Saucedo	Red	9,680	9,680	9,680	9,680	10,648	10,648	60,016
Rosa Elva Gzz.	Red	0	3,500	3,500	3,500	3,500	3,500	17,500
Rubén Garza	Red	0	9,500	9,500	9,500	9,500	9,500	47,500
Mario Padrón	Red	0	0	8,400	8,400	8,400	8,400	33,600
Minerva Jiménez	Red	3,500	3,500	3,500	3,500	3,500	3,500	21,000
Ambrocio S.	Red	0	0	8,400	8,400	8,400	8,400	33,600
Hilda Linares	Red	0	0	6,500	6,500	6,500	6,500	26,000
						TOTAL (MN)		508,774

Nombre	Puesto	Jul	Ago	Sep	Oct	Nov	Dic	TOTAL 1997
Abelardo Miranda	LDP	14,125	14,125	14,125	14,125	14,125	14,125	84,750
Luis E. Santoy	LDP	11,533	11,533	11,533	11,533	11,533	11,533	69,198
Héctor Gzz.	Red	10,670	10,670	10,670	10,670	10,670	10,670	64,020
Mario Saucedo	Red	10,648	10,648	10,648	10,648	10,648	10,648	63,888
Rosa Elva Gzz.	Red	3,500	3,500	3,500	3,500	3,500	7,000	24,500
Rubén Garza	Red	9,500	9,500	0	0	0	0	19,000
Mario Padrón	Red	8,400	8,400	0	0	0	0	16,800
Minerva Jiménez	Red	3,500	3,500	0	0	0	0	7,000
Ambrocio S.	Red	8,400	8,400	0	0	0	0	16,800
Hilda Linares	Red	6,500	6,500	6,500	6,500	6,500	6,500	39,000
Silvia Reyna	Red	0	0	0	0	0	11,000	11,000
						TOTAL (MN)		415,956



## ANEXO D

### Gastos Generados por Concepto de Honorarios durante el periodo Ene-May 1998.

Nombre	Puesto	Ene	Feb	Mzo	Abr	May	TOTAL 1998
Abelardo Miranda	ADMON	14,125	14,125	14,125	15,820	15,820	74,015
Luis E.Santoy	LDP	11,533	11,533	11,533	0	0	34,599
Israel Villegas	Red	10,500	10,500	10,500	13,000	13,000	57,500
Mario Saucedo	LDP	11,500	11,500	11,500	11,500	11,500	57,499
Rosa Elva Gzz.	Red	7,000	7,000	7,000	7,000	7,000	35,000
Rubén Garza	Red	0	9,500	9,500	9,500	9,500	38,000
Mario Padrón	Red	0	0	9,000	9,000	0	18,000
Silvia Reyna	Red	11,000	11,000	11,000	11,000	11,000	55,000
Luis Hdz.	Red	0	0	0	0	10,500	10,500
SCI	Red	0	0	0	0	45,000	45,000
					<b>TOTAL</b>	<b>(MN)</b>	<b>425,113</b>

**ADMON = Administrador del Proyecto**

**LDP =Lider de Proyecto**

**Red = Rediseño**

DIRECCIÓN GENERAL DE BIBLIOTECAS



## ANEXO F

**Gastos Generados por Concepto de Honorarios durante el  
periodo Ene-Dic 1999.**

Nombre	Puesto	Ene	Feb	Mar	Abr	May	Jun	TOTAL 1999
Abelardo M.	ADMON	20,721	20,721	20,721	23,000	23,000	23,000	131,162
José Cramer	LDP	0	4,500	4,500	4,500	4,500	4,500	22,500
Israel Villegas	Red	14,562	14,562	14,562	16,164	16,164	16,164	92,178
Jorge M.	Red	13,085	13,085	13,085	13,085	13,085	13,085	78,509
Mario Saucedo	LDP	16,667	16,667	16,667	18,500	18,500	18,500	105,500
Rosa Gzz.	Red	12,250	12,250	12,250	12,250	13,084	15,000	77,084
Silvia Reyna	Red	13,000	13,000	13,000	13,000	13,000	15,000	80,000
Luis Hdz.	Red	12,000	12,000	12,000	12,000	12,000	0	60,000
Irma Uicab	Red	7,000	9,000	9,000	9,000	10,000	10,000	54,000
Fernando Arcos	Red	10,500	10,500	10,500	10,500	10,500	10,500	63,000
Isela Ledezma	Red	7,000	7,000	7,000	7,000	3,500	0	31,500
Víctor Cirilo	Red	19,500	19,500	19,500	19,500	19,500	20,000	117,500
Karina Uicab	Red	7,000	7,000	7,000	7,000	7,000	7,000	42,000
ASESOR		0	0	0	72,000	72,000	84,000	228,000
SCI	Red	20,000	20,000	20,000	20,000	20,000	20,000	120,000
SINTERMIL	Red	0	29,000	41,000	41,000	41,000	41,000	193,000
							<b>TOTAL (MN)</b>	<b>1,495,934</b>

Nombre	Puesto	Jul	Ago	Sep	Oct	Nov	Dic	TOTAL 1999
Abelardo M.	ADMON	23,000	23,000	23,000	23,000	23,000	23,000	138,000
José Cramer	Red	4,500	0	0	0	0	0	4,500
Israel Villegas	LDP	16,164	16,164	16,164	16,164	16,164	16,164	96,984
Jorge M.	Red	13,085	13,085	13,085	13,085	13,085	13,085	78,509
Mario Saucedo	LDP	18,500	18,500	18,500	18,500	18,500	18,500	111,000
Rosa Gzz.	LDP	15,000	15,000	15,000	15,000	15,000	16,500	91,500
Silvia Reyna	Red	15,000	17,000	17,000	17,000	17,000	17,000	100,000
Irma Uicab	Red	11,000	10,000	11,000	11,000	11,000	11,000	65,000
Fernando Arcos	Red	12,000	12,000	12,000	12,000	0	0	48,000
Víctor Cirilo	Red	22,000	22,000	22,000	22,000	0	0	88,000
Karina Uicab	Red	8,500	8,500	8,500	8,500	8,500	8,500	51,000
Mayra Torres	Red	0	5,000	5,000	5,000	5,000	0	20,000
Alberto Villegas	Red	0	0	9,000	9,000	9,000	0	27,000
ASESOR		96,000	48,000	48,000	0	0	0	192,000
SCI		100,000	100,000	100,000	100,000	100,000	100,000	600,000
							<b>TOTAL (MN)</b>	<b>1,711,492.6</b>

## ANEXO G

### Presupuesto de Inversión en Infraestructura.

Inversiones en USD\$	TOTALES		
	Corp.	Negocio	Total
<b>PROYECTO</b>			
RED VIRTUAL DE DATOS	\$597,703	\$0	\$597,703
<b>RED DE VOZ</b>			
HARRIS/ACTUALIZ. HARRIS	\$249,560	\$0	\$249,560
MULTILINEAS/ACTUALIZ. MULT.	\$22,789	\$0	\$22,789
CONTR. TRONC. DIGITALES	\$158,554	\$0	\$158,554
ADECUAR RED DE VOZ	\$25,000	\$0	\$25,000
	\$0	\$0	\$0
SUBTOTAL	\$1,053,606	\$0	\$1,053,606
<b>EQUIPO COMPUTO CENTRAL</b>			
ACTUALIZ. EQUIPO CENTRAL DEC	\$452,514	\$0	\$452,514
PROGRESS	\$146,668	\$0	\$146,668
<b>RÉDES LOCALES (LAN)</b>			
INFRAESTRUCTURA DE RED	\$0	\$0	\$0
PROCESADORES	\$0	\$0	\$0
LICENCIAS	\$0	\$0	\$0
PC's	\$0	\$0	\$0
<b>OTROS</b>			
CORREO DE VOZ CENTIGRAM	\$19,000	\$0	\$19,000
UPS	\$10,000	\$0	\$10,000
SUBTOTAL	\$628,182	\$0	\$628,182
TOTAL	\$1,681,788	\$0	\$1,681,788



# ANEXO I

## Solicitudes Colocadas en el Periodo Sep-Dic 1999.

Número Asignado	Fecha (DIA)	Autor	Tema	Tipo	Crit	Cal
AMB--S-1	29/12/99	FRANCISCO FDZ.	ABPR: CANCELACION DE ENTRADA	Falla Proc	Ind.	B
AMB--S-23	8/10/99	JAIME LOREDO	ABPR: ENTRADA	Falla Proc	Ind.	E
AMB--S-24	12/10/99	DOLORES TORRES	ABPR: contabilizacion por unidad de ne	Falla Contab	Des.	B
AMB--S-25	20/10/99	BENEDICTO A CASANOVA	ABPR: Modificación	Tiempo Resp.	Ind.	M
AMB--S-26	27/10/99	DOLORES TORRES	ABPR: Requisiciones de Cargo Directo	Falla Proc	Crit.	B
AMB--S-264	18/11/99	GUMERCINDO S.	ABPR: LEYENDA EN CONTRATO	Falla Proc	Ind.	B
AMB--S-265	22/11/99	GUMERCINDO S.	ABPR: TRASLAPE DE FOLIÓS DE CONTRATO	Falla Proc	Ind.	E
AMB--S-267	23/11/99	GUMERCINDO S.	ABPR: PEDIDO GSB012	Falla Proc	Ind.	E
AMB--S-268	24/11/99	DOLORES TORRES	ABPR: reporte de autorizacion de requi	Falla Rep	Des.	B
AMB--S-27	27/10/99	DOLORES TORRES	ABPR: Pendientes de ABAPRO en ATLATEC	Falla Proc	Crit.	B
AMB--S-275	1/12/99	DOLORES TORRES	ABPR: CUENTA-SUBCTA EN CARGO DIRECTO	Falla Contab	Des.	B
AMB--S-276	2/12/99	DOLORES TORRES	ABPR: ARCH. DE DESCARGA	Config	Ind.	B
AMB--S-279	14/12/99	DOLORES TORRES	ABPR: REQUISICIONES DE LA PLANTA 70 EN	Personaliz.Pta	Ind.	B
AMB--S-281	14/12/99	DOLORES TORRES	ABPR: REGISTRO DE FACTURAS DE CARGO DI	Recompilacion	Ind.	B
AMB--S-282	16/12/99	DOLORES TORRES	ABPR: REPORTE CON DATOS DE LA CIAPTA.	Personaliz.Pta	Ind.	B
AMB--S-284	22/12/99	GUMERCINDO S.	ABPR: Problema con requisiciones	Falla Proc	Ind.	E
AMB--S-29	29/10/99	DOLORES TORRES	ABPR: CORRECCION DE CTO-CTA-SCTA	Config	Ind.	B
AMB--S-32	11/11/99	GUMERCINDO S.	ABPR: ASIGNACION DE CONTRATO.	Falla Proc	Crit.	E
CIMA-S-192	25/10/99	JOSE LUIS PUENTE MEDINA	ABPR: ALIMENTACION DE IVA A PEDIDOS	Config	Ind.	B
CIMA-S-195	28/10/99	E. RIVERA TORRES	ABPR: CANCELACION DE ENTRADAS	Falla Proc	Ind.	B
CIMA-S-200	29/10/99	E. RIVERA TORRES	ABPR: DIF. EN SALDOS EN INVENTARIOS	Falla Contab	Crit.	B
CIMA-S-201	29/10/99	E. RIVERA TORRES	ABPR: CANCELACION DE ENTRADAS	Falla Proc	Des.	B
CIMA-S-212	4/11/99	MARIO A MORALES SUAREZ	ABPR: no aparece la requisicion al autor	Falla Proc	Ind.	B
CIMA-S-215	5/11/99	E. RIVERA TORRES	ABPR: DIFERENCIA CONTAB.VS.SISTEMA CTA	Falla Contab	Crit.	B
CIMA-S-230	6/12/99	E. BAUTISTA VERASTEGUI	ABPR: JABAPROCS FUERA DE SERVICIO.	Recompilacion	Crit.	M
CIMA-S-233	17/12/99	E. BAUTISTA VERASTEGUI	ABPR: CANCELACION DE ENTRADAS.	Falla Proc	Crit.	B
CIMA-S-235	21/12/99	E. BAUTISTA VERASTEGUI	ABPR: DATOS CAMBIOS EN ENTRADAS.	Config	Crit.	B
CIMA-S-237	22/12/99	JOSE CRUZ JIMENEZ LOPEZ	ABPR: ALIMENTACION DE COSTO STANDAR	Falla Proc	Ind.	B
CORP-S-1	17/12/99	GREGORIO HDZ.	ABPR: recuperacion de reportes del m	Reproceso	Crit.	B
CORP-S-1	22/12/99	GUILLERMO SILLER	ABPR: CORRECCION DE PROGRAMA DE IMPREC	Personaliz.Pta	Crit.	B

## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
CORP-S-231	3/11/99	ALEJANDRO ALMAGUER	ABPR: Archivo de datos para PRO5%	Config	Crit.	B
CORP-S-281	17/11/99	JORGE LUIS RODRIGUEZ	ABPR:	Falla Proc	Crit.	B
CORP-S-358	8/12/99	OLGA ISABEL MARTINEZ VEGA	ABPR: CARGA DE CONSUMOS EN PLANTAS PEN	Falla Proc	Ind.	B
CORP-S-364	9/12/99	ALEJANDRO GARCIA	ABPR: NO PUEDO GENERAR REPORTE DE	Duda	Ind.	B
CRYSEL-S-1	20/10/99	DIANA M ROMERO BLANCARTE	ABPR: REQUISICION ATRAPADA A42155	Falla Proc	Des.	E
CRYSEL-S-1	15/11/99	MARCO ANTONIO LOPEZ RODRIGUEZ	ABPR: PROBLEMA SIST. ABAPRO	Falla Rep	Crit.	B
CRYSEL-S-1	21/12/99	DAVID GARCIA JAUREGUI	ABPR: REPORTES DE AGOSTO	Reproceso	Crit.	B
CRYSEL-S-1	22/12/99	ALICIA ESTHER VERDIN MARTINEZ	ABPR: BAJA DE PROVEEDOR	Config	Ind.	B
CRYSEL-S-2	8/11/99	DIANA M ROMERO BLANCARTE	ABPR: ORDENAR EL REPORTE DE REQ. A COT	Personaliz. Pta	Des.	B
CRYSEL-S-2	17/11/99	MARCO ANTONIO LOPEZ RODRIGUEZ	ABPR: AUTORIZACION DE REQUISICIONES	Falla Proc	Crit.	B
CRYSEL-S-2	22/12/99	ALICIA ESTHER VERDIN MARTINEZ	ABPR: DAR DE BAJA ENTRADA DE ALMACEN	Falla Proc	Crit.	B
CRYSEL-S-3	17/11/99	MARCO ANTONIO LOPEZ RODRIGUEZ	ABPR: CONFIRMACION DE PEDIDOS	Falla Proc	Crit.	M
CRYSEL-S-3	13/12/99	DIANA M ROMERO BLANCARTE	ABPR: REQ. SIN AUTORIZAR DESAPARECIDAS	Falla Proc	Des.	B
CRYSEL-S-3	22/12/99	ALICIA ESTHER VERDIN MARTINEZ	ABPR: BAJA DE ENTRADA DE ALMACEN	Falla Proc	Crit.	B
CRYSEL-S-37	28/09/99	JESUS TAPIA ROBLES	ABPR: INCORPORAR OPCION CODIFICADORES	Config	Ind.	B
CRYSEL-S-39	22/10/99	JESUS TAPIA ROBLES	SISTEMA DE MANTENIMIENTO: solicitud de servicio al sistema de	Personaliz. Pta	Des.	B
CRYSEL-S-4	28/12/99	ALICIA ESTHER VERDIN MARTINEZ	ABPR: Dar de Baja la Entrada 59195	Falla Proc	Ind.	B
CRYSEL-S-43	2/11/99	JESUS TAPIA ROBLES	SISTEMA DE MANTENIMIENTO: REPORTE DE GASTOS DE MANTTO.	Personaliz. Pta	Ind.	B
CRYSEL-S-49	16/11/99	JESUS TAPIA ROBLES	ABPR:	Personaliz. Pta	Ind.	B
CRYSEL-S-52	19/11/99	JESUS TAPIA ROBLES	ABPR: REPORTE INCOMPLETO POR C.C.	Falla Rep	Ind.	B
CRYSEL-S-53	22/11/99	JESUS TAPIA ROBLES	ABPR: REPORTES DE ABASTOS	Tiempo Resp.	Des.	B
CRYSEL-S-54	22/11/99	JESUS TAPIA ROBLES	ABPR: base de datos a EXCEL	Config	Crit.	B
CRYSEL-S-56	24/11/99	JESUS TAPIA ROBLES	ABPR: FALLA EN CAPTURA DE COMPRADOR RE	Falla Proc	Ind.	B
CRYSEL-S-57	30/11/99	JESUS TAPIA ROBLES	ABPR: REQUISICIONES FANTASMAS DE CARGO	Falla Proc	Ind.	B
CRYSEL-S-63	8/12/99	JESUS TAPIA ROBLES	ABPR: IMPRESION UNITARIA DE REQ. CARGO	Nuevo Des	Ind.	B
CRYSEL-S-65	14/12/99	JESUS TAPIA ROBLES	ABPR: Agregar Reporte de Lenta Rotación	Config	Ind.	B
CRYSEL-S-66	20/12/99	JESUS TAPIA ROBLES	ABPR: Reporte de REQUISICIONES DEVUELT	Falla Rep	Ind.	E
CRYSEL-S-69	27/12/99	JESUS TAPIA ROBLES	ABPR:	Nuevo Des	Ind.	B



## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
CRYSEL-S-7	28/12/99	DIANA M ROMERO BLANCARTE	ABPR: CIERRE DE CONSIGNACION	Falla Proc	Crit.	E
CRYSEL-S-8	29/12/99	DIANA M ROMERO BLANCARTE	ABPR: CIERRE CONSIGNACION	Falla Proc	Crit.	B
CTT-S-11	22/10/99	E. ARAIZA GZZ.	ABPR: FACTURA PERDIDA EN EL SISTEMA	Falla Proc	Ind.	B
CTT-S-14	28/10/99	JAVIER HERRERA M.	ABPR: Insertar al inicio de la cadena	Config	Crit.	B
CTT-S-239	10/12/99	FRANCISCO PINEDO HDZ.	ABPR: Alta de Usuario	Config	Ind.	E
CTT-S-241	13/12/99	VERONICA SALAS	ABPR: ALTA DE USUARIO	Config	Ind.	B
CTT-S-243	21/12/99	FRANCISCO PINEDO HDZ.	ABPR: Revisión del sistema de abasteci	Falla Acceso	Ind.	E
CTT-S-245	28/12/99	VERONICA SALAS	ABPR: ALTA DE USUARIO	Config	Des.	B
CTT-S-247	30/12/99	VERONICA SALAS	ABPR:	Config	Des.	B
HILA-S-219	8/11/99	SALVADOR DE LUNA R.	SISTEMA DE MANTENIMIENTO: REPORTE DE DE GASTOS EN EL SISTEMA D	Personaliz.P ta	Ind.	B
HILA-S-337	23/12/99	ENRIQUE SANCHEZ	ABPR: Falla en sistema abastecimientos	Falla Acceso	Crit.	B
HILA-S-339	28/12/99	ENRIQUE GZZ.	ABPR:	Config	Des.	B
HILA-S-5	13/09/99	ENRIQUE GZZ.	ABPR: calificación de contratistas	Falla Proc	Des.	B
IQUISA MTY.-S-117	15/10/99	RUBEN Z FLORES	SISTEMA DE MANTENIMIENTO: Problemas con la obtención de report	Falla Rep	Crit.	B
IQUISA MTY.-S-159	6/12/99	RUBEN Z FLORES	ABPR: VERIFICACION EVALUACION CTTAS EN	Falla Proc	Ind.	B
IQUISA MTY.-S-167	17/12/99	LETICIA GARCIA ..	ABPR: Registro de facturas en el ABAPR	Falla Proc	Ind.	B
ISTMO-S-10	10/09/99	MAURICIO ORDUNA GZZ.	ABPR: FALLA EN SISTEMA DE ABAPRO	Falla Acceso	Crit.	B
ISTMO-S-11	10/09/99	MAURICIO ORDUNA GZZ.	ABPR: CORRECCION A PROGRAMA	Falla Proc	Crit.	B
ISTMO-S-118	19/10/99	JESUS URBINA MENDEZ	ABPR: CORREGIR FALLA EN MODULO DE CONT	Falla Proc	Crit.	B
ISTMO-S-119	19/10/99	JESUS URBINA MENDEZ	ABPR: REPORTE DE ANALITICOS DE ART. PO	Nuevo Des	Des.	M
ISTMO-S-121	20/10/99	MAURICIO ORDUNA GZZ.	ABPR: REPORTE DE REQUISICIONES DEVUELT	Falla Proc	Crit.	B
ISTMO-S-127	25/10/99	GUILLERMO TREJO BROCA	ABPR: Eliminación de fecha en formato	Personaliz.P ta	Ind.	B
ISTMO-S-132	27/10/99	MAURICIO ORDUNA GZZ.	ABPR: ACTIVO FIJO EN REQUISICIONES DE	Falla Proc	Crit.	R
ISTMO-S-137	29/10/99	MAURICIO ORDUNA GZZ.	ABPR: REPORTE DE CONSUMOS POR C. C. AB	Falla Rep	Crit.	B
ISTMO-S-14	10/09/99	MAURICIO ORDUNA GZZ.	ABPR: COLOCACION DE PEDIDOS	Recompilación	Crit.	B
ISTMO-S-143	2/11/99	JESUS URBINA MENDEZ	ABPR: COLOCACION DE PEDIDOS MATERIAS P	Personaliz.P ta	Crit.	E
ISTMO-S-144	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABAC0839 "Estadísticas d	Reproceso	Ind.	B
ISTMO-S-145	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABATB260 "Mantto. Tabla F	Config	Ind.	B
ISTMO-S-146	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABACG220 "Estatus de Pro	Personaliz.P ta	Ind.	B
ISTMO-S-147	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABAAR005 "Datos del Codif	Personaliz.P ta	Ind.	B



## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
ISTMO-S-148	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABACG220 "Estatus de Pro	Personaliz.P ta	Des.	B
ISTMO-S-149	4/11/99	JACINTO TRUJILLO RMZ.	ABPR: Calificación de Proveedores en C	Falla Proc	Des.	B
ISTMO-S-15	13/09/99	GUILLERMO TREJO BROCA	ABPR: PROBLEMAS EN EL SISTEMA DE ABAPR	Falla Acceso	Crit.	B
ISTMO-S-150	5/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABAP071 "Hoja de Trabaj	Personaliz.P ta	Des.	B
ISTMO-S-151	5/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABAAR005 "Datos del Codif	Validación	Ind.	B
ISTMO-S-152	5/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABASV010 "Separación de M	Depuración	Ind.	B
ISTMO-S-154	8/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción CYDAP050 "Mtto. Traspasos	Falla Proc	Crit.	B
ISTMO-S-155	8/11/99	MAURICIO ORDUNA GZZ.	ABPR: CARGO EN SEPARACION DE MATERIALE	Validación	Ind.	B
ISTMO-S-157	8/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABARP010 "Mantto. a Repos	Validación	Ind.	B
ISTMO-S-158	8/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABASV010 "Alimentación Se	Falla Proc	Crit.	B
ISTMO-S-163	15/11/99	MAURICIO ORDUNA GZZ.	ABPR: REPORTE DE CONSUMOS ABACO092	Falla Rep	Crit.	B
ISTMO-S-168	18/11/99	SOTERO RODRIGUEZ LEE	ABPR: cambio de empresa para elaborar	Config	Crit.	B
ISTMO-S-170	18/11/99	GUILLERMO TREJO BROCA	ABPR: ANTICIPO	Falla Proc	Crit.	B
ISTMO-S-171	18/11/99	GUILLERMO TREJO BROCA	ABPR: REQUISICIONES PERDIDAS	Falla Proc	Ind.	B
ISTMO-S-22	15/09/99	MAURICIO ORDUNA GZZ.	TESORERIA DE NEGOCIOS: REVISAR MODULO DE CANCELACION DE PAGOS	Interfase	Crit.	B
ISTMO-S-295	23/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABASV010 "Alimentación de	Falla Rep	Ind.	B
ISTMO-S-297	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABAAR005 "Consultas de Ar	Falla Cons	Ind.	B
ISTMO-S-298	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABARE055 "Generación de R	Falla Proc	Ind.	B
ISTMO-S-299	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABAAR005 "Ultimas 3 Entra	Falla Cons	Ind.	B
ISTMO-S-300	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABACO131 "Entradas del m	Config	Ind.	B
ISTMO-S-301	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Opción ABAIF010 "Generación de I	NA	Ind.	B
ISTMO-S-302	24/11/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABASA160 "Salidas y Devo	Falla Contab	Ind.	B
ISTMO-S-305	26/11/99	ANGEL PATRACA MOLINA	ABPR: cierre de proveedor en consignac	Falla Proc	Crit.	B
ISTMO-S-313	30/11/99	MAURICIO ORDUNA GZZ.	ABPR: GENERACION DE ARCHIVO DE DESCARG	Falla Proc	Crit.	B
ISTMO-S-314	30/11/99	JESUS URBINA MENDEZ	ABPR: CIERRE DE MES	Falla Proc	Crit.	E
ISTMO-S-320	2/12/99	ESTEBAN FLORES MAYEN	ABPR: ANTICIPO A PROVEEDOR	Falla Proc	Crit.	E
ISTMO-S-324	6/12/99	ADRIANA RMZ. LOPEZ	ABPR: SOLICITUDES DE CONTRATO	Falla Proc	Crit.	B
ISTMO-S-325	7/12/99	FRANCISCO RAFAEL MEDINA ESCUDE	ABPR: Cargos de equipos en el módulo d	Nuevo Des	Ind.	M

## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
ISTMO-S-327	7/12/99	FRANCISCO RAFAEL MEDINA ESCUDE	ABPR: Aviso de Requisiciones devueltas	Personaliz.P ta	Des.	B
ISTMO-S-331	13/12/99	MAURICIO ORDUNA GZZ.	ABPR: DEPURACION DE AMBIENTES	Depuración	Crit.	B
ISTMO-S-334	14/12/99	MAURICIO ORDUNA GZZ.	ABPR: CARGOS CONTABLES CON CEROS	Falla Contab	Crit.	R
ISTMO-S-34	22/09/99	MAURICIO ORDUNA GZZ.	ABPR: GENERACION DE ANTICIPOS	Falla Proc	Crit.	R
ISTMO-S-35	22/09/99	MAURICIO ORDUNA GZZ.	ABPR: ENTRADAS CON INSPECCION	Falla Contab	Crit.	B
ISTMO-S-376	15/12/99	ALFREDO DE LA CRUZ VARGAS	ABPR: CAMBIO CENTRO DE COSTO EN DESCAR	Config	Crit.	M
ISTMO-S-381	16/12/99	JUAN HECTOR HDZ. PALACIOS	ABPR: DIFERENCIAS EN REPORTES DE ABAST	Falla Contab	Crit.	B
ISTMO-S-382	16/12/99	JUAN HECTOR HDZ. PALACIOS	ABPR: ACTUALIZAR NOMBRE DE LOS TIPOS	Config	Des.	B
ISTMO-S-384	17/12/99	SEVERO MARIN MARTINEZ	ABPR: ELIMINAR MOVIMIENTOS DE LOS REPO	Depuración	Crit.	B
ISTMO-S-385	17/12/99	SEVERO MARIN MARTINEZ	ABPR: REPORTES DE MATERIALES Y REFACCI	Depuración	Crit.	B
ISTMO-S-386	17/12/99	MAURICIO ORDUNA GZZ.	ABPR: CAMBIO EN NO. DE ENTRADA	Config	Crit.	B
ISTMO-S-388	20/12/99	JACINTO TRUJILLO RMZ.	ABPR: Repóte ABAAR270 "Hoja de Análisis	Personaliz.P ta	Crit.	B
ISTMO-S-389	20/12/99	MAURICIO ORDUNA GZZ.	ABPR: DIVISA DE ADDENDUM	Config	Crit.	E
ISTMO-S-39	23/09/99	MAURICIO ORDUNA GZZ.	ABPR: DIFERENCIA DE INVENTARIOS	Falla Contab	Crit.	B
ISTMO-S-394	21/12/99	JAIME GALVAN LOPEZ	ABPR: NO SE GENERAN LOS AVISOS DE CONT	Personaliz.P ta	Ind.	B
ISTMO-S-395	22/12/99	JOSE ANTONIO GZZ. FLORES	ABPR: Requisiciones perdidas	Falla Proc	Ind.	B
ISTMO-S-396	22/12/99	JOSE ANTONIO GZZ. FLORES	ABPR: REQUISICIONES YA AUTORIZADAS	Falla Proc	Ind.	E
ISTMO-S-397	23/12/99	JOSE ANTONIO GZZ. FLORES	ABPR: REQUISICIONES	Falla Proc	Ind.	B
ISTMO-S-398	23/12/99	JOSÉ ANTONIO GZZ. FLORES	ABPR: REQUISICION YA AUTORIZADA	Falla Proc	Ind.	E
ISTMO-S-404	29/12/99	MARCELINO HDZ. CABRALES	ABPR: FALLA EN PROCESO DE RESPALDO DE	Falla Proc	Crit.	R
ISTMO-S-406	30/12/99	JESUS URBINA MENDEZ	ABPR: ANALITICO CON ASTERISCOS...	Falla Rep	Crit.	E
ISTMO-S-45	27/09/99	MAURICIO ORDUNA GZZ.	ABPR: CUENTAS DE CARGOS EN CARGO DIREC	Interfase	Crit.	B
ISTMO-S-46	27/09/99	MARCELINO HDZ. CABRALES	ABPR: ** "word2.p" was not found. (293	Recompilaci ón	Crit.	B
ISTMO-S-49	27/09/99	MAURICIO ORDUNA GZZ.	ABPR: REPORTE DE ESTADISTICAS DE PAGO	Falla Rep	Crit.	B
ISTMO-S-5	8/09/99	VICENTE ARTURO HDZ. LEON	ABPR: Abapro (IVA RETENIDO)	Nuevo Des	Ind.	B
ISTMO-S-51	28/09/99	MAURICIO ORDUNA GZZ.	ABPR: PROGRAMA DE TRASPASOS EN CONSIGN	Falla Proc	Crit.	B
ISTMO-S-53	29/09/99	MAURICIO ORDUNA GZZ.	ABPR: CUENTAS DE CARGOS EN REQUISICION	Validación	Crit.	B
ISTMO-S-54	29/09/99	MAURICIO ORDUNA GZZ.	ABPR: IMPORTES DE MATERIAL EN CONSIGNA	Falla Proc	Crit.	M

## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
ISTMO-S-57	29/09/99	MAURICIO ORDUNA GZZ.	ABPR: GENERACION DE ANTICPOS	Falla Proc	Crit.	B
ISTMO-S-68	6/10/99	MAURICIO ORDUNA GZZ.	ABPR: INVENTARIO FISICO	Personaliz.P ta	Crit.	B
ISTMO-S-70	8/10/99	MAURICIO ORDUNA GZZ.	ABPR: SALIDA DE SISTEAM DE ABAPRO	Personaliz.P ta	Des.	M
ISTMO-S-81	12/10/99	MAURICIO ORDUNA GZZ.	ABPR: COSTO ESTIMADO DE REQUISICIONES	Falla Proc	Des.	B
ISTMO-S-84	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: MODIFICACION DE PRECIOS EN CONTR	Validación	Des.	B
ISTMO-S-85	12/10/99	JACINTO TRUJILLO RMZ.	ABPR: MILESIMAS EN EL MODULO DE INVENT	Falla Proc	Crit.	B
ISTMO-S-86	12/10/99	JACINTO TRUJILLO RMZ.	ABPR: NO SE DESPLIEGA LA DESCRIPCION E	Falla Rep	Ind.	B
ISTMO-S-87	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: INCONSISTENCIA EN ARCHIVOS DE RE	Falla Proc	Ind.	B
ISTMO-S-88	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: FAMILIAS DE CARGO DIRECTO	Personaliz.P ta	Des.	B
ISTMO-S-90	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: QUE SE PUEDA BUSCAR ARTICULOS PO	Personaliz.P ta	Des.	B
ISTMO-S-91	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: REHABILITACION DEL REPORTE DE ES	Config	Ind.	B
ISTMO-S-92	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: DEPURACION DE SEPARACIONES DE MA	Depuración	Ind.	B
ISTMO-S-93	12/10/99	JACINTO TRUJILLO RMZ.	ABPR: HABILITAR EN CDE LA MODIFICACION	Personaliz.P ta	Des.	B
ISTMO-S-94	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: DESHABILITACION DE INSPECCIONES	Config	Des.	B
ISTMO-S-95	12/10/99	JACINTO TRUJILLO RMZ.	ABPR: AVISO DE ANTICIPOS EN LA CAPTURA	Personaliz.P ta	Des.	B
ISTMO-S-97	15/10/99	JACINTO TRUJILLO RMZ.	ABPR: Reporte ABACO220 Conciliacion de	Falla Contab	Ind.	B
MPK-S-121	19/10/99	ALEJANDRA RANGEL	ABPR: REPORTE	Personaliz.P ta	Crit.	B
MPK-S-122	20/10/99	JOSE MARIA CHAMU	ABPR: CHECAR STATUS CODIFICADOR	Configuración	Crit.	B
MPK-S-28	1/10/99	RUBEN VALDEZ	ABPR: IMPRESION REPORTES DE PROVEEDORE	Reproceso	Crit.	B
MPK-S-29	15/12/99	RUBEN VALDEZ	ABPR: CLAVE DE ACCESO ABASTECIMIENTOS	Personaliz.P ta	Crit.	B
MPK-S-30	10/10/99	CARLOS R TREVINO	ABPR: NUEVO REPORTE	Personaliz.P ta	Ind.	B
QUIMO-S-1	9/12/99	ANDRES G GZZ.	ABPR: REQUISICIONES	Falla Proc	Crit.	B
QUIMO-S-1	13/12/99	PABLO GARCIA FUENTES	ABPR: ADDENDUMS	Falla Proc	Ind.	R
QUIMO-S-120	11/10/99	JAVIER PADILLA MENDEZ	ABPR: Proceso de generación de requisi	Falla Proc	Ind.	B
QUIMO-S-125	25/10/99	JORGE BERRONES	ABPR: Captura de facturación.	Personaliz.P ta	Ind.	B
QUIMO-S-132	11/11/99	HUMBERTO PUENTE	ABPR: Falta al Ejecutar opcion de cance	Falla Proc	Crit.	B
QUIMO-S-133	12/11/99	LETICIA GARCIA	ABPR: Opción el sistema ABAPRO para im	Nuevo Des	Ind.	B
QUIMO-S-140	22/11/99	PABLO GARCIA FUENTES	ABPR: Addendums	Falla Proc	Ind.	B
QUIMO-S-154	1/12/99	JAVIER PADILLA MENDEZ	ABPR: CONSUMOS DE MESES ANTERIORES.	Reproceso	Crit.	M

## (Continuación ANEXO I)

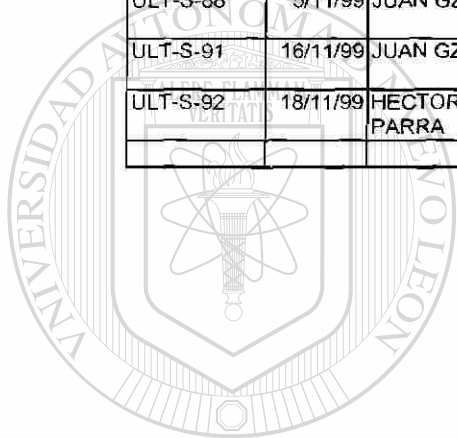
Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
QUIMO-S-155	1/12/99	JAVIER PADILLA MENDEZ	ABPR: REQS. DE SALIDA DE MATERIALES.	Falla Proc	Ind.	B
QUIMO-S-161	9/12/99	JAVIER PADILLA MENDEZ	ABPR: REQUISICIONES DE COMPRA.	Falla Proc	Crit.	M
QUIMO-S-162	10/12/99	JOSE L IBARRA	TESORERÍA DE NEGOCIOS: REPORTE SIN IMPORTE TOTAL	Falla Rep	Crit.	E
QUIMO-S-163	13/12/99	FELIX MAGALLANES DE LUNA	ABPR: Reqsiciones devueltas por abas	Falla Proc	Ind.	R
QUIMO-S-164	13/12/99	JAVIER PADILLA MENDEZ	ABPR: CIERRE DE CONSIGNACION PROVEE. 7	Falla Proc	Ind.	M
RAYON-S-11	6/09/99	GERARDO GUERRERO	ABPR: LISTADO CONSUOS O.T.	Falla Contab	Crit.	B
RAYON-S-118	18/10/99	YULIETTE CAMACHO	ABPR: NO SE PUEDEN AUTORIZAR CONTRATOS	Falla Proc	Crit.	B
RAYON-S-121	20/10/99	JOSE JUAN DEL RIO	ABPR: SUSPENSION DE PAGOS	Tiempo Resp.	Crit.	B
RAYON-S-130	28/10/99	GERARDO RMZ.	ABPR: HISTORIA DE CONSUMOS	Falla Rep	Crit.	B
RAYON-S-132	2/11/99	MARTIN LOPEZ ARREDONDO	ABPR: DEPURACION SISTEMA ANTERIOR DE A	Depuración	Ind.	B
RAYON-S-133	2/11/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LA CADENA DE AUTORIZAC	Configuración	Crit.	B
RAYON-S-134	4/11/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LA CADENA DE AUTORIZA	Configuración	Crit.	B
RAYON-S-136	4/11/99	ROGELIO RMZ.	ABPR: CONCILIACION LISTDOS DE ABAPRO "	Falla Contab	Crit.	B
RAYON-S-137	5/11/99	GERARDO GUERRERO	ABPR: CONSUMO O.T.	Nuevo Des	Crit.	B
RAYON-S-143	24/11/99	PEDRO PABLO DE LA GARZA	ABPR: Error Historia de Consumos	Falla Contab	Crit.	B
RAYON-S-146	30/11/99	ROGELIO RMZ.	ABPR: DESCARGA DE POLIZA DE ABASTO A C	Falla Contab	Crit.	B
RAYON-S-147	30/11/99	ROGELIO RMZ.	ABPR: LISTADOS DE CONSUMOS DEL MES ABA	Falla Contab	Crit.	B
RAYON-S-150	7/12/99	YULIETTE CAMACHO	ABPR: REQUISICIONES	Falla Proc	Des.	B
RAYON-S-151	7/12/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LA CADENA DE AUTORIZAC	Configuración	Des.	B
RAYON-S-153	8/12/99	YULIETTE CAMACHO	ABPR: REQUISICIONES	Configuración	Des.	B
RAYON-S-154	10/12/99	ROGELIO RMZ.	ABPR: CANCELACION DE ANTICIPO EN SISTE	Configuración	Ind.	B
RAYON-S-155	10/12/99	YULIETTE CAMACHO	ABPR: SEPARACION DE MATERIALES	Bloqueo	Ind.	B
RAYON-S-157	14/12/99	GERARDO GUERRERO	ABPR: CONSUMOS POR O.T.	Falla Contab	Crit.	B
RAYON-S-158	15/12/99	YULIETTE CAMACHO	ABPR: NO SE PUEDE DAR SALIDA A ARTICUL	Bloqueo	Crit.	B
RAYON-S-160	16/12/99	MINERVA SANCHEZ	ABPR: CANCELACION ENTRADA No.45689	Falla Proc	Crit.	B
RAYON-S-161	16/12/99	ROGELIO RMZ.	ABPR: CANCELACION DE ENTRADA	Falla Proc	Crit.	B
RAYON-S-163	17/12/99	YULIETTE CAMACHO	ABPR: SEPARACION DE MATERIAL	Bloqueo	Crit.	B
RAYON-S-165	20/12/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LAS CADENAS DE AUTORIZ	Configuración	Crit.	B

## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
RAYON-S-166	20/12/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LA CADENA DE AUTORIZAC	Configuración	Crit.	B
RAYON-S-167	21/12/99	YULIETTE CAMACHO	ABPR: CAMBIO EN LA CADENA DE AUTORIZAC	Configuración	Crit.	B
RAYON-S-168	21/12/99	MARTIN LOPEZ ARREDONDO	ABPR: CORRECCION DE VALIDACION	Configuración	Ind.	B
RAYON-S-170	22/12/99	YULIETTE CAMACHO	ABPR: SEPARACION DE MATERIAL	Bloqueo	Ind.	B
RAYON-S-173	23/12/99	ROGELIO TREVIÑO MEJIA	ABPR: CANCELACION DE ENTRADAS	Falla Proc	Ind.	B
RAYON-S-177	28/12/99	YULIETTE CAMACHO	ABPR: IMPRESIÓN DE PEDIDOS INCLUYENDO	Personaliz.P ta	Des.	B
RAYON-S-183	31/12/99	ROGELIO RMZ.	ABPR: LISTADO ABACO092 CON MATERIALES	Falla Rep	Crit.	R
RAYON-S-30	20/09/99	ROGELIO RMZ.	ABPR: CANCELACION ANTICIPOS	Falla Proc	Crit.	B
RAYON-S-42	4/10/99	PEDRO PABLO DE LA GARZA	ABPR:	Gen. Inf.	Crit.	B
RAYON-S-44	11/10/99	JOSE JUAN DEL RIO	ABPR: PROVEEDOR 42941 MGM. BUSQUEDA LE	Tiempo Resp.	Crit.	B
RAYON-S-45	12/10/99	ROGELIO RMZ.	ABPR: CONSULTA UNIVERSAL DE FACTURAS	Tiempo Resp.	Crit.	B
REX-S-19	27/09/99	CARLOS CABALLERO	ABPR: Conciliaciones en ABAPRO	Falla Contab	Ind.	B
REX-S-23	1/10/99	CARLOS CABALLERO	ABPR: Modificaciones Solicitudes de Co	Personaliz.P ta	Des.	B
REX-S-25	6/10/99	FACUNDO CANTU	ABPR: Descargas Pendientes	Falla Contab	Des.	E
REX-S-27	6/10/99	FACUNDO CANTU	ABPR:	Configuración	Des.	R
REX-S-358	17/12/99	JOSE L LAZARO GARCIA	ABPR: cancelar una entrada de material	Falla Proc	Ind.	B
REX-S-414	29/12/99	JOSE L LAZARO GARCIA	ABPR: ELIMINACION DE SOLICITUD DE ADEND	Configuración	Ind.	E
REX-S-415	29/12/99	JOSE L LAZARO GARCIA	ABPR: Agregar un digito entero en Pantalla	Falla Proc	Ind.	R
REX-S-47	15/11/99	FACUNDO CANTU	ABPR: Problemas con REQUISICIÓN	Duda	Ind.	R
ULT-S-17	20/09/99	JESUS LOPEZ SANCHEZ	ABPR: HABILITAR REPORTES EN ABASTECIMI	Configuración	Crit.	B
ULT-S-31	1/10/99	FERNANDO RONZON SEVILLA	ABPR: No se puede generar un adendum	Duda	Crit.	E
ULT-S-33	1/10/99	JUAN GZZ. AYALA	ABPR: IMPRESION DE REPORTES DEL MES	Duda	Ind.	E
ULT-S-338	14/12/99	ROSENDO HDZ.	ABPR: DEPURAR DIRECTORIO	Depuración	Ind.	E
ULT-S-341	15/12/99	CLAUDIA ALEMAN RUIZ	ABPR: Problemas con el Sistema de Aba	Falla Acceso	Crit.	R
ULT-S-343	15/12/99	FERNANDO RONZON SEVILLA	ABPR: EN LAS CLAVES MAM,PMA Y MAE NO S	Gen. Inf.	Crit.	E
ULT-S-349	23/12/99	JESUS LOPEZ SANCHEZ	ABPR: GENERACION DE REPORTE	Validación	Crit.	E
ULT-S-40	8/10/99	JUAN GZZ. AYALA	ABPR: DIF. EN LA CONCILIACION DEL SIST	Falla Contab	Ind.	B
ULT-S-50	20/10/99	JUAN GZZ. AYALA	ABPR: ABRIR UN NUEVO SUB ALMACEN	Configuración	Ind.	B
ULT-S-54	22/10/99	HECTOR CARDENAS PARRA	ABPR: Interfases MFG/PRO vs Abastecimi	Interfase	Crit.	B

## (Continuación ANEXO I)

Número Asignado	Fecha	(DIA) Autor	Tema	Tipo	Crit	Cal
ULT-S-55	28/10/99	CLAUDIA ALEMAN RUIZ	ABPR: Proceso de Costeo para cierre me	Duda	Crit.	B
ULT-S-60	25/10/99	ROSENDO HDZ.	ABPR: LEVANTADORES DE BASES DE DATOS	Falla Acceso	Ind.	E
ULT-S-63	26/10/99	ROSENDO HDZ.	ABPR: DESIGNACION DE AUTORIZACION ERRO	Duda	Crit.	B
ULT-S-68	28/10/99	ROSENDO HDZ.	ABPR: Levantadores de Bases de Datos	Falla Acceso	Ind.	B
ULT-S-77	1/11/99	JUAN GZZ. AYALA	ABPR: REPORTES DEL CIERRE DE MES DE OC	Configuración	Crit.	B
ULT-S-81	3/11/99	JUAN GZZ. AYALA	ABPR: CORRECCION DE REPORTE ABACO092	Reproceso	Ind.	B
ULT-S-85	4/11/99	ROSENDO HDZ.	ABPR: Manuales del Sistema de Abasteci	Sol. Manual	Ind.	R
ULT-S-88	5/11/99	JUAN GZZ. AYALA	ABPR: BAJAR A EXCEL REPORTE PROCO270	Duda	Ind.	B
ULT-S-91	16/11/99	JUAN GZZ. AYALA	ABPR: DIFERENCIA CONCILIACION OCT-99	Falla Contab	Ind.	B
ULT-S-92	18/11/99	HECTOR CARDENAS PARRA	ABPR: PROBLEMAS CON REQUISICIÓN	Falla Proc	Crit.	R



# UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



