

44
21



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**VENTANILLA ELECTRÓNICA DE
SERVICIO PARA EL SEGUIMIENTO Y
SOPORTE A CLIENTES**

**TESIS PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A N:
LEONARDO GÁLVEZ MEDINA
Y
SILVIA PEÑA GARCÍA**



DIRECTOR DE TESIS: ING. ADOLFO MILLÁN NÁJERA

MÉXICO, D.F.

JUNIO 1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PRÓLOGO

El tema a tratar en esta tesis es el desarrollo de un sistema de ventanilla electrónica de servicio para el seguimiento y soporte a clientes. La ventanilla electrónica consiste en un sistema electrónico de registro y seguimiento de solicitudes de servicio. Éste tipo de aplicaciones suele ser muy necesario en muchas empresas que requieren medir sus niveles de atención o bien desean alcanzar un nivel de servicio a clientes adecuado.

En nuestro caso partimos de una empresa que no cuenta con un modelo automatizado del servicio y sus procesos de atención son ineficientes. Con la herramienta a desarrollar se pretende dar el primer paso para mejorar y medir los procesos, los tiempos de servicio, atender a una demanda mayor, e integrar a otras empresas de servicio bajo esquemas de integración en las que participen como solucionadores del problema, esto es, la ventanilla de servicio es el punto donde los clientes se comunican para reportar problemas de su equipo, y serán empresas externas las que se encarguen de dar el servicio de último nivel.

Como una actividad paralela al sistema de información es necesario definir la infraestructura adecuada de hardware y software. Para lo cual empleamos una metodología, técnicas para el desarrollo de sistemas de base de datos, análisis costo beneficio entre las distintas alternativas, y técnicas para el control y administración de proyectos.

En el capítulo uno nos concentramos en la definición de la metodología de control de proyectos. En éste se lleva a cabo una propuesta de pasos ordenados que nos permitirán dar un mejor seguimiento al proyecto y detectar las fases y las actividades que involucran a cada una de éstas, todo con la finalidad de garantizar la calidad del sistema a desarrollar.

En el capítulo dos nos concentramos en la definición del proyecto. En él se estudia el problema y se hacen las primeras propuestas de solución, además se definen alcances, objetivos y se analizan los procesos e infraestructura actuales de la empresa.

En los capítulos tres y cuatro nos dedicamos al análisis, diseño y desarrollo del sistema. En el capítulo tres se lleva a cabo el análisis del sistema de información en base al diseño conceptual de las base de datos y al modelo entidad relación; Por otra parte también se lleva a cabo un análisis funcional del sistema, finalmente bajo éste método se integran funciones y datos para el desarrollo de la aplicación. El capítulo cuatro compara las diferentes alternativas de hardware y software requeridas para implantar la infraestructura de voz y datos en la empresa, se hace un análisis comparativo de sistemas operativos, topologías de red y los estándares en el mercado. Además se lleva a cabo un estudio y propuesta del número de líneas telefónicas, troncales y equipo óptimo para la operación y expansión a las necesidades del cliente.

En el capítulo cinco nos dedicamos a hacer un estudio de métodos para el control de proyectos (GANTT, PERT, etcétera) de esta forma se lleva a cabo también la aplicación de técnicas de seguimiento y planeación de proyectos. Además nos ayuda a presupuestar el trabajo a realizar en términos de horas hombre y a determinar la ruta crítica de nuestro proyecto. Para lo cual hicimos un caso práctico utilizando el programa MS Project que maneja los distintos métodos

Terminamos con las conclusiones y resultados obtenidos del desarrollo teórico práctico del sistema.

ÍNDICE

CAPÍTULO 1. METODOLOGÍA DE CONTROL DE PROYECTOS POR UTILIZAR.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ESTRUCTURA DE LA METODOLOGÍA.....	1
<i>Definición de proyecto</i>	1
<i>Desarrollo a la medida</i>	1
<i>Mantenimiento a sistemas</i>	1
1.3. DEFINICIÓN DE PROYECTO.....	2
<i>Fase I.A: Análisis preliminar</i>	2
<i>Fase I.B: Diseño conceptual</i>	2
1.4. DESARROLLO A LA MEDIDA.....	3
<i>Fase II: Diseño funcional y técnico</i>	3
1.5. DESARROLLO, PRUEBAS E INSTALACIÓN.....	5
<i>Fase III: Desarrollo</i>	5
<i>Fase IV: Pruebas</i>	6
<i>Fase V: Instalación</i>	6
1.6. MANTENIMIENTO A SISTEMAS.....	7
<i>Fase A: Recepción</i>	7
<i>Fase B: Organización</i>	8
<i>Fase C: Ejecución</i>	8
<i>Fase D: Mejora</i>	8
<i>Conclusiones de la metodología</i>	9
CAPÍTULO 2. DEFINICIÓN DE PROYECTO.....	12
2.1. ANTECEDENTES.....	12
2.2. OBJETIVOS Y ALCANCE DEL PROYECTO.....	12
<i>Objetivos</i>	12
<i>Alcance</i>	12
2.3. SITUACIÓN ACTUAL DE LA EMPRESA Y SU INFRAESTRUCTURA.....	13
<i>Procesos de operación de la empresa</i>	13
<i>Infraestructura</i>	14
2.4. ANÁLISIS Y PROPUESTAS A LA PROBLEMÁTICA ACTUAL.....	16
<i>Problemática</i>	16
<i>Análisis y Propuestas</i>	16
CAPÍTULO 3. DESARROLLO A LA MEDIDA: DISEÑO TÉCNICO Y FUNCIONAL.....	18
3.1. INTRODUCCIÓN.....	18
3.2. LEVANTAMIENTO DE INFORMACIÓN.....	18
<i>Módulo de apertura y seguimiento</i>	19
<i>Consultas y reportes</i>	19
<i>Administración del sistema</i>	19
<i>Utterías</i>	20
3.3. DISEÑO TÉCNICO.....	20
<i>Los diagramas de contexto</i>	20
<i>Diseño de modelo conceptual y vistas del sistema</i>	21
3.4. ESTRATEGIAS PARA EL DISEÑO DE ESQUEMAS.....	24
<i>Estrategia descendente</i>	24
<i>Estrategia ascendente</i>	24
<i>Estrategia centrífuga</i>	24
<i>Estrategia mixta</i>	24

3.5. ESQUEMA DE VISTAS A PARTIR DE REQUERIMIENTOS EXPUESTOS EN LENGUAJE NATURAL.....	26
<i>Diseño a partir del lenguaje natural</i>	26
3.6. VISTAS DEL SISTEMA DE LA VENTANILLA ELECTRÓNICA.....	28
3.7. ESQUEMA DE DATOS GENERAL DEL SISTEMA (DICCIONARIO DE DATOS).....	30
3.8. MEJORA DE LA CALIDAD DE UN ESQUEMA DE BASE DE DATOS.....	33
<i>Transformaciones de esquemas</i>	33
<i>Ciclos de Interrelaciones</i>	34
<i>Atributos derivados</i>	34
<i>Subconjuntos implícitos</i>	34
3.9. TRANSFORMACIONES PARA LOGRAR EXPRESSIVIDAD Y AUTOEXPLICACIÓN.....	35
<i>Eliminaciones de subentidades colgantes en las jerarquías de generalización</i>	35
<i>Eliminación de entidades colgantes</i>	35
<i>Creación de una generalización</i>	35
<i>Creación de un nuevo subconjunto</i>	35
3.10. TRANSFORMACIONES PARA LOGRAR LA NORMALIZACIÓN.....	36
<i>Dependencias funcionales</i>	36
<i>Anomalías de actualización</i>	36
3.11. ATRIBUTOS PERTINENTES DE LAS ENTIDADES E INTERRELACIONES.....	37
<i>Atributos pertinentes de una entidad</i>	37
<i>Atributos pertinentes de una interrelación</i>	37
<i>Primera forma normal</i>	37
<i>Segunda forma normal</i>	37
<i>Tercera forma normal</i>	38
<i>Forma normal de Boyce - Codd</i>	38
<i>Cómo razonar respecto a la normalización</i>	38
3.12. DISEÑO FUNCIONAL EN EL DISEÑO DE BASE DE DATOS.....	40
<i>Análisis funcional usando el modelo de flujo de datos</i>	40
<i>Primitivas para el análisis funcional</i>	41
3.13. ESTRATEGIAS PARA EL ANÁLISIS FUNCIONAL.....	42
<i>Estrategia descendente</i>	42
<i>Estrategia ascendente</i>	42
<i>Estrategia centrífuga</i>	42
<i>Estrategia mixta</i>	43
<i>Diseño de esquema funcional</i>	43
3.14. ANÁLISIS CONJUNTO DE DATOS Y FUNCIONES.....	44
<i>Esquemas externos para los diagramas de flujo de datos</i>	45
<i>Metodología para el análisis conjunto de datos y funciones</i>	45
<i>Refinamientos mutuos</i>	48
<i>Esquema de navegación para operaciones con bases de datos</i>	48
CAPITULO 4. INFRAESTRUCTURA DE HARDWARE POR APLICAR.....	49
4.1 EQUIPO REQUERIDO PARA EL PROYECTO.....	49
<i>Levantamiento de información</i>	49
4.2 ANÁLISIS Y DISEÑO DE LA INFRAESTRUCTURA DE VOZ.....	51
4.3 EQUIPO SUGERIDO PARA LA RED DE VOZ.....	53
<i>Propuesta 1</i>	53
<i>Propuesta 2</i>	54
4.4 ANÁLISIS Y DISEÑO DE LA INFRAESTRUCTURA DE DATOS.....	55
<i>Topología De La Red</i>	55
4.5. ESTÁNDARES LAN.....	57
<i>IEEE 802.3</i>	57
<i>IEEE 802.4</i>	59
<i>IEEE 802.5</i>	60

<i>Recomendaciones para la red de datos</i>	61
4.6. HERRAMIENTAS DE SERVICIO Y SOPORTE	63
<i>Antecedentes</i>	63
<i>Selección del sistema operativo de red adecuado</i>	63
<i>Desarrollo de un software a la medida</i>	65
4.7. ANÁLISIS DE RIESGOS Y SELECCIÓN DE SOFTWARE Y HARDWARE	66
<i>Conclusiones de la sección de voz</i>	66
<i>Conclusiones de la red de datos</i>	67
CAPÍTULO 5 PLANEACIÓN Y ADMINISTRACIÓN DE PROYECTOS	69
5.1 INTRODUCCIÓN	69
5.2 MÉTODOS DE PLANEACIÓN: PERT (PROGRAM EVALUATION REVIEW TECHNIQUE) Y CPM (CRITICAL PATH METHOD)	69
<i>Antecedentes</i>	69
<i>Actividades</i>	70
<i>Diagrama de red</i>	71
<i>Método PERT</i>	74
<i>Método CPM</i>	76
5.3 PROYECTOS DE SOFTWARE	79
<i>Ámbito del software</i>	80
<i>Riesgos</i>	81
<i>Presupuesto del proyecto de software</i>	82
<i>Descomposición</i>	83
<i>Herramientas automáticas de cálculo</i>	83
5.4 APLICACIÓN DE LOS CONCEPTOS DE ADMINISTRACIÓN DE PROYECTOS	83
CAPÍTULO 6 ASEGURAMIENTO DE LA CALIDAD Y MANTENIMIENTO DEL SISTEMA	85
6.1. ASEGURAMIENTO DE CALIDAD DEL SOFTWARE	85
<i>Definición</i>	85
<i>Factores que determinan la calidad del software</i>	85
<i>Factores de operación del producto</i>	86
<i>Factores de revisión del producto</i>	87
<i>Factores de transición del producto</i>	87
MÉTRICAS DEL SOFTWARE	87
<i>Actividades del QA</i>	90
6.2. REVISIÓN DEL SOFTWARE	91
REVISIONES TÉCNICAS FORMALES	91
<i>Registro e informe de la revisión</i>	92
<i>Directrices de una revisión</i>	92
6.3 PRUEBAS DE SOFTWARE	93
<i>Objetivos de la prueba</i>	94
<i>El flujo de información en la prueba</i>	94
<i>Diseños de casos de prueba</i>	94
<i>La prueba de la caja blanca</i>	95
<i>Prueba del camino básico</i>	96
<i>Prueba de la caja negra</i>	96
<i>Análisis de valores límite</i>	97
6.4 MANTENIMIENTO DE SOFTWARE	98
<i>Definición de mantenimiento de software</i>	98
<i>Características del mantenimiento</i>	99
<i>El mantenimiento estructurado</i>	99
<i>Costos del mantenimiento</i>	100
<i>Problemas</i>	100
<i>Facilidad de mantenimiento</i>	101

<i>Medidas cuantitativas</i>	101
<i>Tareas de mantenimiento</i>	102
<i>Organización de mantenimiento</i>	102
<i>Informes</i>	102
6.5 FLUJO DE SUCESOS PARA REALIZAR UN MANTENIMIENTO	103
<i>Registro de Información</i>	104
<i>Evaluación</i>	105
6.6 CONTROL DE VERSIONES.....	106
6.7 CONTROL DE CAMBIOS.....	107
CONCLUSIONES.....	109
BIBLIOGRAFÍA.....	112

ANEXOS

ANEXO 1 Diccionario de Datos

ANEXO 2 Directorio de Programas

ANEXO 3 Revisión Técnica Formal

ANEXO 4 Diagrama de Gantt y PERT

ANEXO 5 Pantallas del Sistema

CAPÍTULO 1. METODOLOGÍA DE CONTROL DE PROYECTOS POR UTILIZAR

1.1. INTRODUCCIÓN.

Uno de los factores que en mayor medida podrá influir en el éxito de las áreas de "Sistemas" será la habilidad que se tenga para estructurar y dar seguimiento al trabajo que realizan. Para lograrlo, es necesario el uso de diferentes herramientas. Entre ellas, una metodología para desarrollar y mantener sistemas. Algunas características que presenta la metodología se describen a continuación y son :

- Se puede aplicar independientemente de las herramientas que se utilicen, ya que únicamente proporciona la logística a seguir en el desarrollo de cualquier proyecto.
- Se deja a los líderes y consultores de proyectos la decisión de adaptar los ciclos de vida para cada proyecto en específico, eliminando aquellas actividades y/o productos que no apliquen. Incluso los proyectos "emergentes" podrán definir su propio ciclo de vida.

1.2. ESTRUCTURA DE LA METODOLOGÍA

La metodología está integrada por tres módulos que a continuación se detallan de manera general:

Definición de proyecto

- Alineación de proyectos a la Planeación Tecnológica .
- Evaluación de alternativas del proyecto: Reutilizar sistemas existentes, comprar un paquete, desarrollar un sistema nuevo.
- Justificación del proyecto.

Desarrollo a la Medida

- Modificaciones mayores y medias de sistemas en operación.
- Desarrollo de sistemas nuevos.

Mantenimiento a sistemas.

- Modificaciones menores de sistemas en producción.
- Atención de maquilas de otros proyectos.
- Atención de reportes de problemas y performance (rendimiento).
- Definición de versiones.

1.3. DEFINICIÓN DE PROYECTO

El módulo de definición de proyecto consta de dos fases: a) un análisis preliminar , y b) un diseño conceptual.

Fase I.A: Análisis preliminar

- Definición del alcance del proyecto
- Revisión de la situación actual
- Formación del equipo de trabajo que intervendrá en el análisis, desarrollo, instalación y pruebas del sistema durante el proyecto
- Revisión de las necesidades específicas del cliente contra los requerimientos del negocio

Fase I.B: Diseño conceptual

- Evaluación de las alternativas de desarrollo
- Análisis costo beneficio
- Plan de trabajo en fases

Resumiendo, el objetivo final de este módulo es elaborar el plan de trabajo para las fases de "Análisis preliminar " y "Diseño conceptual" y establecer el equipo de trabajo que ejecutará el plan. Posteriormente durante el desarrollo del diseño conceptual del sistema. El consultor debe evaluar las alternativas de solución:

- Reutilizar los sistemas existentes
- Comprar un paquete en el mercado, o bien
- Desarrollar un nuevo sistema (desarrollo a la medida) para el caso de que fuese necesario efectuar un
- Análisis de costo beneficio

Una vez definidos los alcances y expectativas del proyecto se obtendrá como producto final un diseño conceptual y un plan preliminar de trabajo para el siguiente módulo: Desarrollo a la medida. A continuación se presenta diagrama de flujo de la metodología en esta fase

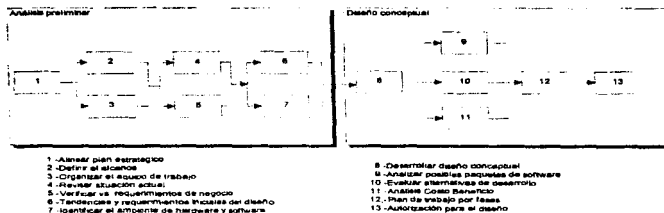


Figura 1.1 Definición de proyecto

1.4. DESARROLLO A LA MEDIDA

Fase II : Diseño funcional y técnico

Las fase de diseño buscan hacer una interpretación de las especificaciones del usuario para generar el diseño funcional y técnico por aplicar. En caso de que se haya optado por un desarrollo a la medida el consultor procederá entonces a seguir las siguientes fases de diseño.

Fase II.A : Diseño funcional Elaborar el "diseño del sistema de acuerdo con la información recopilada en la Fase de "Definición de Proyecto". Esta etapa es conocida también como el levantamiento de información. Al diseño funcional lo componen estos elementos y actividades

Organización de las fases de diseño

- Elaboración de un modelo general
- Plan y estándares de trabajo

Levantamiento de información

- Análisis de situación actual
- Identificar requerimientos funcionales
- Identificar requerimientos adicionales

La definición de hardware y software para integrar la plataforma

- Establecer criterios de selección
- Identificar alternativas
- Evaluación y selección preliminar
- Selección final

Diseño funcional

- Definición y desarrollo de Prototipos (si fuese necesario)
- Definición de entradas y salidas
- Definición y detección de políticas y procedimientos y generación de un diagrama estructural
- Verificar factibilidad técnica y autorización del usuario
- Diseño de los procesos del sistema.

Fase II.B : Diseño técnico. Preparar la infraestructura del medio ambiente para desarrollar el nuevo sistema. Al diseño técnico lo componen estos elementos y actividades:

Diseño técnico

- Diseño de arquitectura técnica
- Diseño de entidades de datos
- Diseño y pruebas de conversión
- Determinar requerimientos de recursos de operación

Plan y costos de desarrollo

- Cálculo de costos de desarrollo e instalación
- Confirmar costo/beneficio

Aprobación del diseño

- Publicación de especificaciones

Una vez que se cuenta con el diseño conceptual, los diseños técnicos y funcionales del sistema y la aprobación de los requerimientos del usuario, se puede proceder a desarrollar pruebas y ejecutar la instalación.

Es importante mencionar que estos requerimientos son indispensables y el contar con ellos en esta fase permitirá la adecuada administración de recursos humanos y técnicos necesarios para el desarrollo. Incluso el usuario puede contar en ese momento con un panorama completo del desarrollo en tiempo y costos, sin haber invertido grandes cantidades de dinero. A continuación se presenta diagrama de flujo de la metodología en esta fase.

Instalación de hardware y software

- Instalación de hardware y software

Generación de ambientes

- Generación de ambiente de desarrollo
- Generación de infraestructura aplicativa
- Desarrollo de ambiente de pruebas

Programación

- Establecer mecanismos de control del desarrollo
- Codificación
- Revisión del código
- Preparación de datos de prueba
- Pruebas unitarias internas

Fase IV : Pruebas

Pruebas

- Realización de pruebas modulares e integrales
- Revisar resultados a detalle
- Realizar pruebas con el usuario

Fase V : Instalación

Capacitación

- Elaborar documentación del usuario
- Capacitación técnica aplicativa

Instalación

- Documentar para la liberación
- Acuerdos para liberación
- Creación y aplicación de un ambiente de producción
- Soporte seguimiento e instalación
- Documentación y mejoras al sistema
- Revisión posterior a la instalación.

A continuación se presenta diagrama de flujo de la metodología en esta fase

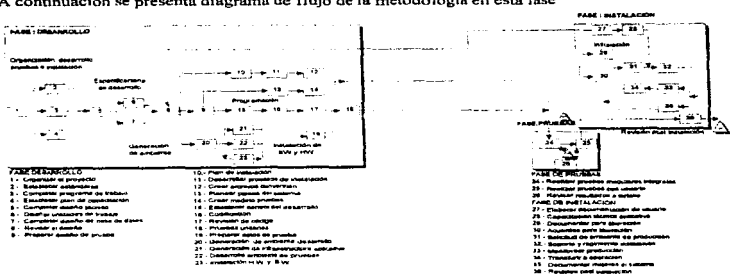


Figura 1.3 Desarrollo, pruebas e instalación

1.6. MANTENIMIENTO A SISTEMAS

Es muy importante mencionar que dentro del ciclo de vida de un sistema se debe prever un módulo de mantenimiento. Esto se requiere generalmente cuando el usuario hace alguna solicitud de cambio o de nuevo requerimiento, mantenimiento correctivo o de (rendimiento) o bien de apoyo en caso de contingencia, para lo cual generalmente se establece algún mecanismo de control y aseguramiento de atención a problemas. Al módulo podremos dividirlo en cuatro fases:

Fase I: Recepción

Solicitudes de apoyo al usuario

- Coordinación con el usuario
- Recepción y registro
- Análisis y clasificación

Reporte de problemas de rendimiento

- Recepción y registro
- Establecer operación

Solicitudes de Maquila

- Involucración previa
- Recepción y registro

Fase B: Organización

Jerarquización de Solicitudes

- Diagnóstico de solicitud
- Establecer prioridades de atención

Plan de trabajo a nivel de área

- Calendarización de solicitudes
- Seguimiento de la capacidad de respuesta
- Análisis de cumplimiento a tiempos objetivo

Fase C: Ejecución

Diseño detallado

- Especificaciones de programación
- Plan de trabajo personal

Modificaciones al sistema

- Complementar especificaciones de programación
- Realizar cambios
- Actualizar manual de usuario
- Actualizar documentación técnica
- Pruebas

Instalación

- Capacitación al personal
- Catalogación en producción

Fase D: Mejora

- Definición de áreas de oportunidad
- Decisión de optimización.

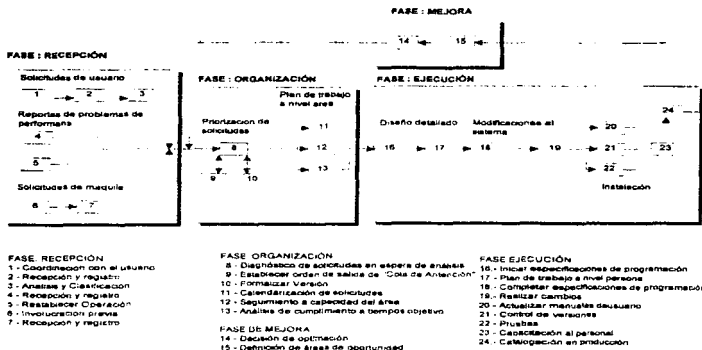


Figura 1.4 Mantenimiento de sistemas

Conclusiones de la metodología

Como se puede ver la metodología presentada busca proporcionar al usuario los mecanismos de control para el buen fin del proyecto, y por medio del cual el consultor podrá llevar a cabo un mejor seguimiento a su proyecto, sin olvidar ningún detalle asegurando de esta forma la calidad del producto por desarrollar. El diagrama general de la metodología se muestra a continuación, en forma de módulos y fases. En el próximo capítulo nos centraremos en la definición y alcance del proyecto a llevar a cabo

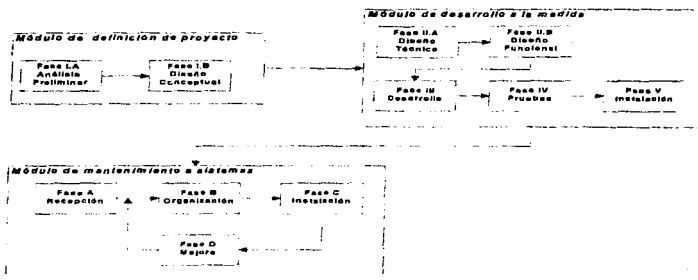


Figura 1.5 Ciclo de vida del software

FALTA PAGINA

NO. 11

CAPITULO 2. DEFINICIÓN DE PROYECTO

2.1. ANTECEDENTES

La empresa SERVICOMP es una entidad privada que se dedica a proveer insumos y dar mantenimiento a equipos terminales punto de venta (T.P.V) de una institución bancaria.

SERVICOMP está muy interesada en mejorar la atención a su cliente, debido a que ha recibido quejas del servicio. Su cliente comenta que ha notado deterioro en el mismo pues éste en ocasiones se ha vuelto lento e ineficiente según el consenso medido por las áreas de reclamaciones. Por lo anterior, SERVICOMP nos solicita un rediseño de la infraestructura y automatización del área de atención al cliente a fin de mejorar la operación del negocio en el DF que cuenta con alrededor de 3,000 afiliados a los que hay que dar servicio a sus terminales cuando éstos lo requieran.

2.2. OBJETIVOS Y ALCANCE DEL PROYECTO

Objetivos

Implantar un infraestructura que agilice y mejore el registro, atención y distribución de solicitudes de servicio. Para lo cual se sugiere organizar:

1. Una infraestructura de comunicaciones eficiente para atender las llamadas del cliente (teléfonos, conmutadores, troncales necesarias, etcétera).
2. Una infraestructura de cómputo adecuada para almacenar información y modernizar los procesos de interacción entre las áreas de atención al cliente y de mantenimiento e instalaciones (red de cómputo, servidor, cableado, módems, etcétera).
3. Una herramienta de software a la medida que permita intercambiar información con las áreas ya mencionadas, y la gerencia permitiendo, obtener información estadística oportuna para facilitar la toma de decisiones.

Alcance

Estos objetivos se alcanzarán a partir de las siguientes premisas:

- El cliente requiere que las llamadas recibidas sean atendidas en un 90 por ciento sin importar la demanda a lo largo del día.
- Infraestructura de comunicaciones y de cómputo que resuelva sus necesidades a corto plazo y de pie a un crecimiento plancado si la demanda de servicio aumenta.
- Requiere contar con una herramienta informática que le permita verificar sus niveles de servicio y que agilice la atención a sus clientes.

2.3. SITUACIÓN ACTUAL DE LA EMPRESA Y SU INFRAESTRUCTURA

Procesos de operación de la empresa

Actualmente SERVICOMP cuenta con un área de atención a clientes que consta de una planta de tres personas para satisfacer las solicitudes de servicio. Estas tres personas levantan los reportes en formatos de papel. Dichos reportes recolectados son enviados a las doce del día y a las cinco de la tarde al área de mantenimiento ubicada en otro edificio. Pero como dos personas se quedan a recibir llamadas, se reduce la posibilidad de atender a los clientes.

El área de mantenimiento está integrada por un jefe y cinco técnicos que atienden el área Metropolitana. El jefe de la misma recibe el primer lote de reportes a las doce del día y en ese momento lo distribuye a los técnicos disponibles. Posteriormente, en el transcurso del día, el jefe de mantenimiento realiza un reporte de las solicitudes de servicio recibidas la víspera y lo turna a la gerencia para que ésta decida qué y cuándo se hace. Al término del día el jefe de mantenimiento recibe el segundo lote de reportes y traza rutas de trabajo para que al siguiente día los técnicos las cubran. A continuación se muestra el diagrama de flujo del proceso descrito

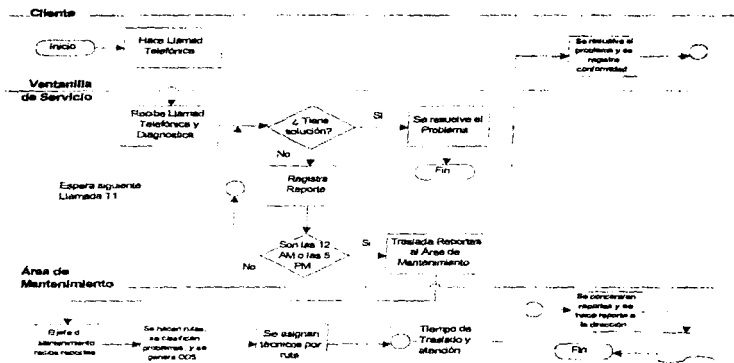


Figura 2.6 Flujo actual de la operación

Infraestructura

Sección de voz. La empresa SERVICOMP cuenta con esta infraestructura:

Siete líneas directas; tres de las cuales son asignadas al área de atención de reportes y las otras cuatro sirven al resto de la empresa.

Sección de datos

Actualmente carecen de alguna infraestructura para atender los reportes, debido a que como ya se mencionó son llenados a mano y transportados físicamente por una persona.

Las tres computadoras personales con que cuenta el cliente manejan únicamente graficadores, procesadores de texto y hojas de cálculo que sirven para generar informes y reportes pero ninguna está conectada en red ni es utilizada por el área de atención a clientes. Sus características técnicas son:

EQUIPO	CARACTERISTICAS
Tres computadoras personales	Microprocesador: 486 a 66 Mhz RAM : 8 Mb Disco Duro : 200 Mb Monitor Super VGA

A continuación mostramos un esquema de la situación actual del cliente:

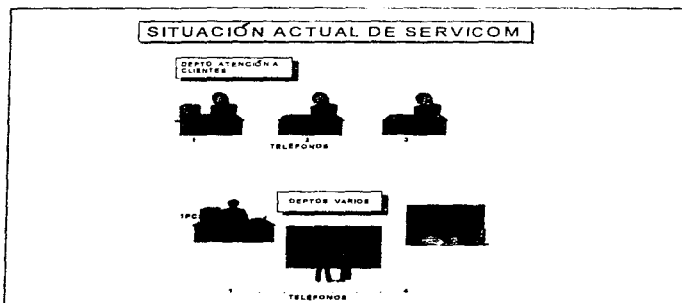


Figura 2.7 Situación actual e infraestructura de la empresa

2.4. ANÁLISIS Y PROPUESTAS A LA PROBLEMÁTICA ACTUAL

Problemática.

SERVICOMP, como se apuntó en la sección anterior, y debido a que cuenta con poca infraestructura encara los siguientes problemas:

- El proceso de recolección de llamadas del área de atención a clientes llega a ser ineficiente debido a que se presentan casos en que la información (reportes de servicio) se pierde.
- No cuenta con información oportuna para tomar las decisiones pertinentes en cada caso.
- Muchos clientes están insatisfechos porque las líneas telefónicas siempre están ocupadas.
- La atención de las llamadas se retrasa hasta más de 24 horas, debido a que actualmente no se tiene una forma eficiente de hacer llegar con oportunidad los reportes al área de mantenimiento de la empresa.

Debido a lo anterior, la empresa no está en una situación competitiva respecto de otras empresas del ramo.

Análisis y Propuestas.

El proyecto reclama contar con una infraestructura de equipo y con un software desarrollado a la medida de las necesidades del cliente.

Debido a que se ha presentado bloqueo de llamadas en la compañía es necesario realizar un análisis de tráfico telefónico con el fin de determinar cuál es el equipo de comunicaciones apropiado. Por el momento podemos recomendar que el equipo del conmutador debe contar con facilidades de ACD (Attention Call Distribution) para distribuir oportunamente las llamadas al área de atención. Posteriormente se realizará un estudio más detallado, mismo que nos permitirá definir el número de extensiones y troncales necesarios para cubrir como mínimo al 90 por ciento de las solicitudes de servicio.

Con base en lo anterior se definirá el tamaño y la cantidad de estaciones de trabajo requeridas para la red, puesto que debe corresponder al mismo número de extensiones destinadas al área.

El sistema propuesto para el manejo de información debe cumplir con requisitos de organización muy específicos. Por tal razón será necesario excluir la posibilidad de adquirir un software disponible en el mercado, y nos enfocaremos al desarrollo de un sistema, y en paralelo a la instalación de la infraestructura necesaria.

Puesto que éste sistema servirá para registrar datos y obtener información estadística a nivel administrativo, sugerimos que el desarrollo se haga en un manejador de base de datos como Foxpro for Windows, pues el usuario desea contar con la posibilidad de hacer uso de multitareas. Otra razón más es que contamos ya con la experiencia en el uso de éste manejador de base de datos, mismo que permite ejecutar desarrollos rápidos y de buena calidad. Otros puntos que nos llevaron a desarrollar en esta herramienta son:

1. Es una herramienta ideal para ambientes multiusuarios bajo Windows
2. Es una herramienta de desarrollo visual y que permite obtener desarrollos en corto plazo
3. Es fácil de administrar el proyecto con ella. Además facilita y agiliza el mantenimiento
4. Xbase es su lenguaje nativo. Lo cual nos ayuda a contar con mano de obra que pueda darle un mantenimiento eficiente con manos experimentadas en otros manejadores de bases de datos para computadoras personales tales como Dbase y Clipper.
5. El volumen de datos que va a manejar el sistema no es muy grande por lo que el manejador propuesto es suficiente para soportar la cantidad de información esperada

Lo anterior beneficiará al cliente, puesto que se podrán efectuar modificaciones rápidas que harán más versátil al sistema debido a que se podrá adecuar a sus necesidades futuras.

CAPITULO 3. DESARROLLO A LA MEDIDA: DISEÑO TÉCNICO Y FUNCIONAL

3.1. INTRODUCCIÓN

A finales de 1960, cuando las bases de datos entraron por primera vez al mercado del software, los diseñadores de bases de datos actuaban como artesanos, con herramientas muy primitivas: Diagramas de bloques y estructuras de registros eran los formatos comunes para las especificaciones. La tecnología de base de datos cuenta ya con un marco teórico que incluye la teoría relacional de datos, procesamiento y optimización de consultas, control de concurrencia, gestión de transacciones y recuperación, etcétera. Sin embargo, la mayoría de las organizaciones y los diseñadores individuales confía muy poco en la metodología para llevar a cabo el diseño de las bases de datos y esto con frecuencia se considera una de las principales causas de fracaso en el desarrollo de los sistemas de información.

El sistema de información de una empresa es un conjunto de actividades que regulan la distribución y el comportamiento de la información y el almacenamiento de los datos relevantes para administrar la empresa. *Una base de datos es cualquier conjunto de datos estructurados y almacenados en una computadora.* Los sistemas de gestión de base de datos se utilizan para almacenar, manipular y recuperar datos. Las bases de datos son tan sólo un elemento de los sistemas de información, que incluyen programas, interfaces, etcétera.

El diseño de un sistema de información es una actividad complicada y elaborada que incluye la planificación, la especificación y el desarrollo de cada componente del sistema, así como un estudio de factibilidad, levantamiento de información, diseño, elaboración de prototipos, implantación, validación, pruebas y operación.

3.2. LEVANTAMIENTO DE INFORMACIÓN

Un estudio de factibilidad trata de determinar la rentabilidad de las distintas alternativas de diseño del sistema de información y las prioridades de los diversos componentes del sistema. Por otra parte el levantamiento de información se ocupa de la recolección de análisis y requerimientos, ocupándose de la comprensión de la misión del sistema de información. Esta centra su interacción con los usuarios, quienes a su vez describen sus necesidades a los diseñadores, que las almacenan en lo que se denomina especificación de requerimientos. En general las especificaciones son más bien informales y desorganizadas y se expresan comúnmente en lenguaje natural o semiformal (claves y lenguaje natural)

En nuestro caso, el sistema que requiere nuestro cliente deberá consistir en un lenguaje informal, automatizar y el enviar reportes de manera eficiente y rápida a las distintas

filiales, para que éstas atiendan a los comercios afiliados al servicio de terminales punto de venta (TPV) en el área metropolitana. Por tanto, sus necesidades se verán resueltas con un sistema que se estructure mediante los siguientes módulos:

1. Módulo de apertura y seguimiento de reportes
2. Módulo de consultas y reportes.
3. Módulo de Administración y mantenimiento a catálogos de comercios, resolvidores, usuarios, equipos, localidades, etcétera.
4. Módulo de utilerías del sistema, funciones de respaldo, borrado, reindexación, etcétera.

Módulo de apertura y seguimiento

Tiene como alcance y misión recolectar solicitudes de servicio de los diferentes comercios afiliados a lo largo del día. Para lo cual el sistema debe solicitar el número afiliación del comercio y la localidad a la que corresponde, después el sistema asignará un número de folio.

También será necesario para que el usuario envíe la información a quien atiende los reportes¹ generados durante el día para que sean atendidos. Una vez hecho esto el sistema también debe permitir el cierre y la atención del reporte anotando la solución correspondiente

Consultas y reportes

Le servirá al usuario para llevar a cabo la generación de reportes estadísticos y de control de niveles de servicio de sus resolvidores, así como el comportamiento y necesidades frecuentes del cliente a lo largo del tiempo.

Administración del sistema

Sirve para contar con un sistema dinámico en el que podrán variar, aumentar, disminuir o modificarse los datos administrativos de usuarios, comercios afiliados, equipos, localidades, resolvidores, etcétera.

¹ posteriormente los denominaremos como "Resolvidores"

Utilerías

Sirven para dar mantenimiento al sistema en caso de que el performance, y la capacidad del mismo se vea disminuida por el tamaño de las bases de datos; también debe permitir la reindexación y respaldo de información

A grosso modo el sistema requiere de todos los puntos mencionados. Esta información deberá ser presentada al usuario, así como una lista de los programas y funciones que se realizarán para desarrollar el sistema de base de datos. Esto deberá ser validado y firmado por el usuario antes de iniciar el desarrollo para evitar así trabajos de más y desacuerdos con el usuario.

3.3 DISEÑO TÉCNICO

En el diseño detallado del sistema el desarrollados puede hacer uso de diversas herramientas cómo son el uso de "*Diagramas de contexto*" y "*el Diseño de esquemas*", que le permitan interpretar en forma sencilla y gráfica la realidad. Es decir el concepto que ha sido solicitado por el usuario en su etapa inicial.

Los diagramas de contexto

Permiten al analista definir y detectar con el usuario las entradas, salidas y actividades del diseño conceptual. Inicialmente con la ayuda de éstos se considera la metodología cómo una caja negra y se examinan sus entradas y salidas, luego se mira adentro de la caja y se presentan las principales fases de diseño que deben pertenecer a una metodología para el diseño conceptual de bases de datos. El siguiente diagrama muestra las entradas y salidas típicas de una metodología para diseño conceptual.



Figura 3.1 Diagrama de contexto

1. Entradas. Son los requerimientos necesarios para efectuar el proceso de diseño, y se pueden clasificar en varias formas. En nuestro diagrama se clasifican:

- **Requerimientos de datos.** Describen la estructura de datos que se debe almacenar en la base de datos
- **Requerimientos de funciones.** Describen la estructura dinámica del sistema de información, identificando funciones o actividades del sistema y flujos de información
- **Requerimientos de aplicación.** Describen operaciones sobre los datos tales como modificaciones, inserciones, etcétera

2. Salidas. Las salidas producidas incluyen:

- **Esquema conceptual de datos** describen todos los datos presentes en los requerimientos.
- **Esquemas de funciones.** Describen funciones y actividades del sistema de información y los flujos de información entre ellos
- **Las especificaciones de alto nivel de la aplicación** describen las operaciones efectuadas con la base de datos.
- **Documentos de diseño** éstos ofrecen información adicional sobre cada una de las salidas anteriores.








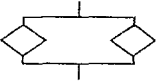

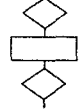



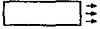
Una vez con la información anterior el analista podrá iniciar el esquema conceptual. El objetivo principal será dar a los requerimientos una estructura que facilite posteriores actividades de modelado.

Diseño de modelo conceptual y vistas del sistema

Antes de iniciar la exposición de los modelos de esquema se debe definir el concepto de primitivas. El diseño de un esquema conceptual es el resultado de un análisis complejo de los requerimientos del usuario. En consecuencia, es producto de un proceso interactivo. Durante ese proceso, se empieza con una versión preliminar de esquema misma que sufre una serie de transformaciones que dan lugar a la versión definitiva. Para lograr esto, el analista se apoya de las transformaciones primitivas. Es decir en las transformaciones con una estructura simple que pueden descomponerse en otras más sencillas.

Las primitivas se clasifican en dos grupos: ascendentes y descendentes. Las primeras se caracterizan por el refinamiento de un concepto inicial con el objeto de obtener una descripción más detallada del concepto. Mientras que las descendentes por su parte introducen conceptos nuevos y propiedades que no aparecían en versiones anteriores al sistema. A continuación se muestran las tablas de las primitivas ascendentes y descendentes.

Figura 3.2 Clasificación de las primitivas descendentes

<p>T1: Entidad \rightarrow Entidades Relacionadas</p>		
<p>T2: Entidad \rightarrow Generalización (Entidad - Subconjunto)</p>		
<p>T3: Entidad \rightarrow Entidades no Relacionadas</p>		
<p>T4: Interrelación \rightarrow Interrelaciones paralelas</p>		
<p>T5: Interrelación \rightarrow Entidad con Interrelaciones</p>		
<p>T6: Desarrollo de atributos</p>	 	 


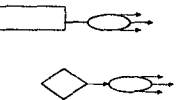

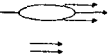


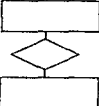

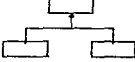


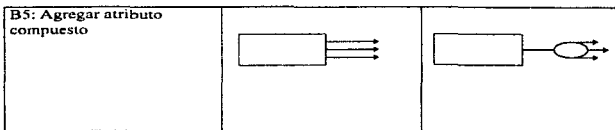
T7: Desarrollo de atributos compuestos		
T8: Refinamiento de atributos		

Figura 3.3 Clasificación de las primitivas ascendentes

B1: Generación de Entidad		
B2: Generación de Interrelación		
B3: Generación de Generalización		
B4: Incluyendo de atributos		



3.4. ESTRATEGIAS PARA EL DISEÑO DE ESQUEMAS

Para diseñar esquemas se distinguen cuatro estrategias fundamentales: descendente, ascendente, centrífuga y mixta.

Estrategia descendente

Se caracteriza por el uso de primitivas de refinamiento descendentes. A partir de éste esquema cada primitiva introduce nuevos detalles al esquema. El proceso termina cuando se representan todos los requerimientos. Es importante señalar que en una descendente pura, los conceptos representados al final deben estar presentes en cada plano del refinamiento.

Estrategia ascendente

Se obtiene aplicando sólo las primitivas de refinamiento ascendentes. En esta estrategia se parte de conceptos elementales para construir conceptos más complejos. En ella los requerimientos se descomponen, se conceptualizan de manera independiente y finalmente se fusionan en un esquema global.

Estrategia centrífuga

Es un caso especial de estrategia ascendente. En ella se parte de fijar los conceptos más importantes y evidentes y se inicia la expansión de conceptos y funciones que giran alrededor de ella, que se propagan hacia afuera a partir de un punto central, a manera de las ondas seleccionando, primero los conceptos más cercanos a la idea inicial, y navegando después hacia los más distantes.

Estrategia mixta

La estrategia mixta aprovecha tanto la estrategia ascendente, como la descendente, al permitir particiones controladas de los requerimientos. Cuando el dominio de la aplicación es muy complejo, el diseñador divide los requerimientos en subconjuntos, que más tarde se consideran por separado.

A continuación se muestra una tabla de comparación de las estrategias para el diseño de esquemas.

Descendente	Los conceptos se refinan progresivamente	No hay efectos secundarios indeseables	Requieren de un diseñador hábil con alta capacidad de abstracción desde el comienzo
Ascendente	Los conceptos se crean a partir de componentes elementales. Decisiones locales de diseño sencillas	Necesidad de una reestructuración después de aplicar cada primitiva ascendente	
Centrífuga	Los conceptos se construyen con un efecto de mancha de aceite	Facilidad para descubrir nuevos conceptos cercanos a los anteriores	La visión global del dominio de la aplicación se tiene solo al final
Mixta	Partición descendente de los requerimientos; integración ascendente usando un esquema de armazón	Enfoque divide y vencerás	Requiere de decisiones cruciales sobre el esquema de armazón al inicio del proceso de diseño

Figura 3.4 Tabla de comparación

Con la ayuda de estas herramientas podemos iniciar el diseño de vistas del sistema. Su objetivo principal es crear un esquema conceptual partiendo de una descripción informal de los requerimientos del usuario. Se utiliza el término vista para referirse a la percepción de una base de datos o de los requerimientos de datos de una aplicación, tal como lo ve un usuario. El diseño de vistas abarca dos actividades distintas:

1. Análisis de los requerimientos, para captar el significado de los objetos de interés de la aplicación, su agrupación en clases, sus propiedades, etcétera.
2. La representación de éstos objetos, clases y propiedades utilizando los conceptos del modelo entidad relación

Nuestro diseño partirá del diseño de vistas a base de lenguaje natural. Es decir, a partir de la síntesis de información de un lenguaje coloquial proveniente del usuario. Existen tres tipos de requerimientos: lenguaje natural, formularios y declaración de registros.

3.5. ESQUEMA DE VISTAS A PARTIR DE REQUERIMIENTOS EXPUÉSTOS EN LENGUAJE NATURAL

Las descripciones en lenguaje natural se hacen normalmente por escrito; así se deduce la información sobre la estructura de la base de datos a partir de un análisis textual de los requerimientos. El diseño de nuestro sistema es un claro ejemplo de esta situación, pues a partir de esta información se obtiene un diseño conceptual de datos, funcionalidad, objetivos y alcances del sistema de información.

Diseño a partir del lenguaje natural

En éste caso nuestro usuario nos ha definido sus requerimiento a partir de lenguaje natural. A continuación se hace una descripción de sus requerimientos:

El área de atención a clientes requiere levantar reportes a sus clientes. Exige que cada reporte sea asignado a un "resolvidor" quién a su vez haya sido recibido por un "representante" del área. El "representante" tomará el "tipo de equipo", "el sintoma", "el nombre de la persona que reporta" y "la causa" que resulte al tipo de diagnóstico detectado, para clasificarlo en una categoría de causas.

El sistema por su parte deberá detectar si el comercio que solicita existe, y además le debe permitirle dar de alta un nuevo "comercio" si no existe en el sistema. También podrá dar de alta comercios "prospecto" de instalación" para asignarlo al "resolvidor de ventas."

"El representante" debe poder actualizar el estatus del reporte y hacer los "comentarios" correspondientes o los datos generales del "comercio". El usuario podrá llegar al reporte ya sea por "el número de folio del reporte" o por "el número de afiliación del comercio."

El área de atención a clientes quiere enviar a través de un disco vía módem todos aquellos reportes que se generen a lo largo del día, en cualquier momento. Dichos reportes deben haberse suscitado a lo largo del día, estar abiertos, y pertenecer al "resolvidor" al que se les desea enviar.

Además debe poder cerrar los reportes en forma automática a partir de un disco que emita el "resolvidor" en el formato de base de datos (*.DBF); El reporte se cerrará a partir del "número de folio" que esté cerrado y que traiga la descripción de la solución del mismo, así como "la fecha y la hora de atención".

El sistema trabajará en red y será multiusuario, por lo que se debe contar con un algoritmo de asignación de folios que no los repita. Además un "representante" sólo puede estar activo en una sola terminal. El "password" del representante deberá estar oculto o almacenado de tal forma que no se pueda descifrar por cualquier persona.

- El sistema debe contar con un módulo que permita dar mantenimiento a los diferentes catálogos de resolutores, causas, etcétera.
- El desarrollo debe contemplar un módulo de seguridad y mantenimiento de la base de datos.
- Se debe incluir un módulo de mantenimiento a las bases de datos: Reindexación, borrado y respaldo:
- Por último se debe contar con un módulo de reportes y consultas que permita la explotación de la información.

Una conceptualización inicial de los requerimientos del usuario permite definir el siguiente diagrama estructural

	Comercial	Administración	
Registro	Clientes con más de 3 llamadas	Catálogos de:	Utilleras:
Consultas	Distribución por códigos de causa	a) Causas	a)Reindexación
Prospectos	Reportes abiertos	b) Ciudades	b)Respaldo
Envío y recepción	Prospectos	c) Comercios	c)Borrar
		d) Resolutores	d)Exportar
		e) Tipos de equipo	Salida
		e) Usuarios	

Figura 3.5 Diagrama estructural

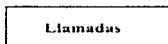
Cómo se puede observar hasta el momento se cuenta con un diseño general y no detallado del sistema. "El diseño detallado" se obtiene a partir del uso de herramientas como los diagramas de contexto y el diseño de vistas basados en los requerimientos obtenidos de lenguaje natural.

A continuación se presenta el diseño de vistas que se desprende del análisis, definición y conceptualización del modelo de datos del sistema planteado.

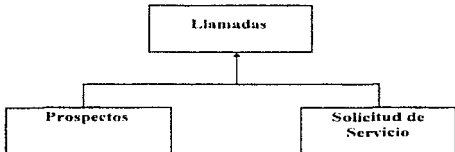
3.6. VISTAS DEL SISTEMA DE LA VENTANILLA ELECTRÓNICA

Partiendo de las especificaciones en lenguaje informal proporcionadas por el usuario podemos darnos cuenta que la entidad principal es la llamada o solicitud de servicio. En torno a ésta giran todos los conceptos. Aplicando la estrategia centrífuga tenemos que:

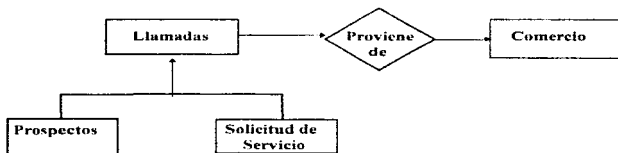
1. Entidad principal: *Llamadas*



2. *Llamadas* se descompone en dos subconjuntos : a) *Prospectos*; y b) *Solicitudes de Servicio*



3. Toda *Llamada* proviene de un *Comercio o Cliente Afiliado*.



4. Las *Llamadas* son atendidas por un *Representante* y son asignadas a un *Resolveror*. Todas tienen una *Causa*, reportan un *Equipo* y provienen de un *Comercio* que está en una *Localidad*.

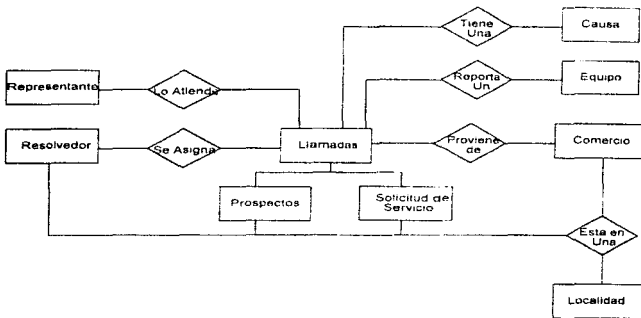


Figura 3.6 Modelo de datos preliminar (Vista inicial)

Cómo se puede ver partimos de una entidad y obtuvimos la vista final que contiene todos los elementos entidad - relación del sistema de información que el usuario prevé.

A Continuación se hace una descripción de cada entidad, por medio de la cual se podrán observar sus atributos individuales

3.7. ESQUEMA DE DATOS GENERAL DEL SISTEMA (DICCIONARIO DE DATOS)

Este esquema se obtiene de la detección de las necesidades de información de las diferentes entidades del modelo de datos. Aquí se podrá definir el diccionario de datos, posteriormente se podrá obtener una mejor calidad en las estructuras utilizando técnicas para lograrlo y que veremos más adelante en este capítulo. Por lo pronto el esquema de datos obtenidos es el que sigue:



Entidad: LLAMADAS

Atributos Compuestos

Generales:

Folio	07(9)
Afiliación	07(x)
Centro_Regional	02(x)
Comercio:	35(x)
Dirección	35(x)
Teléfono	11(x)
Código_postal	05(x)
Cve_Representante	02(x)
Cve_Equipo	02(x)

Atributos compuestos: Datos del Reporte

Cve_Resolver	02(x)
Sintoma	35(x)
Fecha_reporte	08(d)
Hora del Reporte	05(x)
ReportadoX	35(x)

Atributos compuestos: Datos de la solución

Codigo_de_causa	02(x)
Hora_de_Asigna	05(x)
Fecha_Asigna	08(d)
Fecha_solucion	08(d)
Hora_Solucion	05(x)
Seguimiento	(Texto)
Estatus	02(x)

Identificadores

key1: Afiliación + Centro_Regional
Key2: Folio

Entidad: PROSPECTO**Atributos Compuestos**

Generales :	Folio	07(9)
	Centro_Regional	02(x)
	Comercio:	35(x)
	Dirección	35(x)
	Teléfono	11(x)
	Código_postal	05(x)
	Cve_Representante	02(x)
	Cve_Equipo	02(x)
	Tipo_call	01(9)
	Cta_de_Cheques	12(x)
	Cve_Resolvedor	02(x)

Entidad: COMERCIO

Afiliacion	07(x)
Centro_Regional	02(x)
Comercio	35(x)
Dirección	35(x)
Teléfono	11(x)
Código_Postal	05(x)
Prioridad	01(9)
Colonia	12(x)

Identificadores

key1 : Afiliacion + Centro_Regional
key 2: Afiliacion

Entidad: LOCALIDAD

Centro_regional	02(x)
Localidad	35(x)
Dirección	35(x)
División	03(x)
Zona	03(x)

Identificadores

keyL1: Centro_Regional

Entidad: CAUSA

Código_de_Causa	02(x)
Descripción	35(x)

Identificadores

keyCI:Codigo_de_Causa

Entidad: EQUIPO

Nombre	30(X)
Modelo	30(X)
Cve_equipo	02(X)

Identificadores

keyE1: Cve_Equipo

Entidad: REPRESENTANTE

Cve_Representante	02(X)
Password	15(X)
Menu1	05(X)
Menu2	06(X)
Menu3	05(X)
Menu4	04(X)
Nombre	35(X)

Identificadores

keyRPI: Cve_representante

Entidad: RESOLVEDOR

Cve_Resolvedor	02(X)
Nombre	35(X)
Dirección	35(X)
Teléfono	11(X)

Identificadores

keyRS1: Cve_resolveror

Con ayuda de la metodología para el modelo de datos tenemos un diseño rápido y eficiente de los elementos de datos que intervienen en el desarrollo del sistema. Con esta técnica se tienen la información indispensable para la estructura de datos.

3.8. MEJORA DE LA CALIDAD DE UN ESQUEMA DE BASE DE DATOS

Transformaciones de esquemas

Las transformaciones de esquemas se aplican a un esquema de entrada S1 y producen un esquema resultante S2.

Una forma de comparar el contenido de información de dos esquemas S1 y S2 es comparar su capacidad para responder a consultas: se dice que S1 y S2 poseen el mismo contenido de información si, por cada consulta Q que pueda expresarse sobre S1, existe una consulta Q' que pueda expresarse sobre S1, y existe una consulta Q'' que puede expresarse sobre S2 dando la misma respuesta y viceversa.

De forma similar, se dice que el contenido de información de un esquema A es mayor que el de un esquema B si existe alguna consulta Q sobre A que no tenga una consulta correspondiente sobre B, pero no viceversa. Ahora estamos en condiciones de clasificar las transformaciones:

1.- **Transformaciones que preservan la información:** el contenido de información del esquema no es alterado por la transformación.

2.- **Transformaciones que cambian la información,** clasificadas más en detalle cómo sigue:

- a) **Transformaciones de aumento:** el contenido de información del esquema resultante es mayor que el del esquema de entrada.
- b) **Transformaciones de reducción:** el contenido de información del esquema resultante es menor que el del esquema de entrada.
- c) **Transformaciones no comparables:** en cualquier otro caso.

La comprobación de la calidad hace uso frecuente de las transformaciones que preservan la información: el contenido de información del esquema conceptual no debe variar, pero la organización de los conceptos debe mejorar. Las transformaciones de reducción no sólo se usan cuando el esquema conceptual contiene conceptos superfluos que no están expresados en los requerimientos y, por tanto, deben eliminarse. La tabla resume los tipos de transformaciones usadas más comúnmente durante el diseño conceptual.

Aumento	Diseño descendente
Aumento	Diseño ascendente
No comparable	Resolución de conflictos
Aumento	Aumento de propiedades interesquemáticas
Preservación	Reestructuración para lograr minimalidad, legibilidad y normalidad

Figura 3.14 Tipos de transformaciones

La redundancia en los esquemas Entidad Relación (ER²) puede surgir por varias razones. Una de las más comunes es que *“los requerimientos poseen una redundancia inherente, y ésta a veces se comunica al esquema”*. También, en una metodología ascendente, la redundancia puede surgir cuando se expresan en diferentes esquemas conceptos relacionados, y se hace evidente cuando los esquemas se fusionan. La redundancia en esquemas conceptuales ER está inmersa en los siguientes aspectos de los esquemas.

Ciclos de Interrelaciones

La redundancia se da cuando una Interrelación R1 entre dos entidades posee el mismo contenido de información que una ruta de Interrelaciones R2, R3,... Rn que conecta exactamente los mismos pares de casos de entidades que R1.

A pesar de que la redundancia en los ciclos de Interrelaciones depende del significado, existen obvias comprobaciones sintácticas que pueden realizarse para detectar la redundancia.

Atributos derivados

La redundancia puede deberse a la existencia de un algoritmo para calcular los valores de datos derivados de los otros datos; de aquí que los datos derivados puedan omitirse de un esquema ER mínimo.

Subconjuntos implícitos

Después de la integración de esquemas, algunos de los subconjuntos podrían derivarse de otros subconjuntos presentes en el esquema.

En resumen, es tarea del diseñador decidir si acepta la redundancia en el esquema conceptual o la elimina. En cualquier caso, la redundancia puede ser fuente de anomalías en la administración de datos. Por ello debe estar claramente indicada en el esquema.

² ER Entidad - Relación

3.9.TRANSFORMACIONES PARA LOGRAR EXPRESIVIDAD Y AUTOEXPLICACIÓN

La autoexplicación se logra cuando las propiedades de los datos se expresan usando solamente el propio modelo conceptual, en vez de anotaciones adicionales. La expresividad se realiza simplificando el esquema.

Eliminaciones de subentidades colgantes en las jerarquías de generalización.

Puede suceder que el diseñador cree una generalización en el proceso de asignar diferentes propiedades a entidades de la jerarquía. Si, al final del proceso de diseño, las subentidades no se distinguen por ninguna propiedad específica, se pueden reducir a la superentidad. La diferencia entre las distintas entidades se expresa entonces mediante un atributo.

Eliminación de entidades colgantes

Considérese colgante una entidad E si posee pocos (posiblemente uno) atributos A_i y una conexión con otra entidad (la entidad principal) a través de una Interrelación R ; en éste caso, puede convenir simplificar el esquema eliminando la entidad colgante y la Interrelación de conexión, pasando los atributos A_i de la entidad colgante a la entidad principal.

Creación de una generalización

Esta transformación se aplica cuando se descubren dos entidades distintas con propiedades similares, que pertenecen en realidad a la misma jerarquía de generalización. La adición de una generalización proporciona sencillez y concisión al esquema resultante mediante el uso de la propiedad de herencia.

Creación de un nuevo subconjunto

Esta transformación destaca el papel de una entidad. Se puede aplicar a entidades con cardinalidad mínima de cero en una Interrelación. Esto significa que la interrelación se aplica a un subconjunto de los casos de la entidad.

3.10. TRANSFORMACIONES PARA LOGRAR LA NORMALIZACIÓN

En el modelo relacional, la normalización de relaciones es un proceso de aplicación de transformaciones progresivas para lograr la forma normal deseada. El proceso está guiado por las dependencias funcionales.

Dependencias funcionales

Existe una dependencia funcional (DF¹) entre dos atributos monovalentes, A1 y A2, de una entidad E o de una Interrelación R, si cada valor de A1 corresponde exactamente a un valor de A2.

Decimos que A1 determina funcionalmente a A2, lo cual se denota también como A1 → A2; el atributo a la izquierda de la DF se llama el *determinante*. Las DF se establecen de forma similar entre conjuntos de atributos. Las dependencias funcionales se establecen, asimismo, entre atributos de Interrelaciones, exactamente con la misma interpretación.

Estas dependencias dicen que si se asigna el valor de los atributos determinantes, se encontrará en la base de datos un valor de los atributos determinados.

Anomalías de actualización

Si bien las dependencias funcionales que corresponden a identificadores no causan problemas, otras dependencias que pueden existir en una entidad pueden causar las llamadas anomalías de actualización.

- 1.- Anomalía de inserción
- 2.- Anomalía de eliminación
- 3.- Anomalía de actualización

Todas estas anomalías se relacionan con la presencia de una dependencia funcional indeseable. El proceso de normalización es una progresiva detección y eliminación de esas dependencias no deseadas.

¹ Dependencia Funcional

3.11. ATRIBUTOS PERTINENTES DE LAS ENTIDADES E INTERRELACIONES

Las mayores dificultades para extender el tratamiento de la normalización del modelo relacional al modelo ER surgen de las distintas formas de expresar la identificación. Los identificadores externos de una entidad proveen dependencias funcionales entre atributos de la entidad.

Atributos pertinentes de una entidad

Considérese una entidad E, para cada identificación externa a partir de una entidad E1, se incorpora en el conjunto de atributos pertinentes de E un identificador interno de E1. Este proceso es interactivo si a su vez posee identificadores externos.

Atributos pertinentes de una interrelación

Sea R una Interrelación binaria entre las entidades E1 y E2; m1 y m2 denotan las cardinalidades máximas de E1 y E2, respectivamente, en R. Se consideran tres casos:

- 1.- **Interrelación de uno a uno ($m1 = 1$ y $m2 = 1$).** Se incorpora en los atributos pertinentes de R un identificador elegido arbitrariamente, sea de E1 ó de E2.
- 2.- **Interrelación de uno a muchos ($m1 = 1$ y $m2 = n$).** Se incorpora en los atributos pertinentes de R un identificador de E1.
- 3.- **Interrelación de muchos a muchos ($m1 = n$ y $m2 = n$).** Se incorpora en los atributos pertinentes de R un identificador de E1 y uno de E2.

Estas reglas se extienden con facilidad a las Interrelaciones n - arias.

Primera forma normal

Se deben quitar campos duplicados o que tengan más de un valor, en una nueva tabla.

Segunda forma normal

Antes de introducir la segunda forma normal, es necesario definir los atributos primos: un atributo no es primo si no pertenece a ningún identificador; es primo en caso contrario. Una entidad o interrelación está en segunda forma normal si no existe ninguna DF cuyo determinante esté propiamente contenido en un identificador y cuyo atributo del lado derecho sea no primo.

Es decir si existe una DF entre campos y llaves (indicadores) lo mejor es reestructurar la entidad o descomponerla mediante una transformación para lograr la normalización. Otra forma de verlo es que se deben mover a una tabla distinta aquellos campos que no dependen de la llave primaria de la tabla en cuestión.

Tercera forma normal.

Antes de presentar la tercera forma normal, hay que definir las dependencias transitivas. Una DF, $A \rightarrow C$, es transitiva si existen dos dependencias $A \rightarrow B$, $B \rightarrow C$, tales que A, B y C sean grupos diferentes de atributos. Entonces, la dependencia $A \rightarrow C$ se puede inferir como una combinación de $A \rightarrow B$, $B \rightarrow C$; como tal, esta dependencia es redundante y una causa de anomalías.

Una entidad o Interrelación esta en tercera forma normal si se encuentra en segunda forma normal y no tiene ninguna DF transitiva.

Para este caso se deben agrupar entidades en una sola entidad partiendo de DF transitivas. En otras palabras, se deben eliminar todos aquellos campos que sean derivados de otros ya existentes.

Forma normal de Boyce - Codd.

Otra forma normal muy conocida en el modelo relacional es la forma normal de Boyce - Codd. Una entidad o interrelación está en la forma normal de Boyce-Codd (BCNF), si cada determinante de sus DF es un identificador.

Si las DF de una entidad son identificadores en todos sus casos entonces se tiene una forma Boyce -Codd

Cómo razonar respecto a la normalización.

Las formas normales subsiguientes requieren la identificación y el empleo de nuevos tipos de dependencias (como las dependencias multivaluadas, de dominio, de reunión y de inclusión). El enfoque utilizado anteriormente puede ampliarse para tratar éstos tipos de dependencias, pero más bien, se evaluará la importancia de la normalización en el marco del diseño conceptual de base de datos.

La normalización es una herramienta para validar la calidad del esquema, más que un método para diseñar el esquema. Esto es discutible, porque varios enfoques proponen la normalización como el único método aceptable para diseñar bases de datos. Nuestro enfoque sugieren el uso del propio modelo relacional como modelo de diseño; suponen un conjunto de DF como entrada al proceso de diseño y producen una serie de relaciones en una forma normal dada como resultado.

Aunque creemos que la normalidad es una propiedad importante de un esquema, pensamos que las dependencias son medios inadecuados y poco claros para captar los requerimientos en el dominio de la aplicación. Preferimos los mecanismos de abstracción que ofrece el modelo ER, pues son fáciles de entender y se pueden representar en forma natural mediante diagramas.

El objetivo de la normalización, hasta la forma normal de Boyce - Codd, es mantener las DF separadas, al asociar un elemento del modelo con cada grupo de DF (entidad o relación), que tenga los determinantes de las DF cómo identificadores. Así, cada concepto del dominio de aplicación corresponde exactamente a un concepto del esquema.

Esta separación de conceptos es el resultado natural del uso de abstracciones de clasificación, agregación y generalización, y de la aplicación de una metodología descendente para producir un esquema mediante refinamientos sucesivos. Nuestro modelo de datos después de aplicar las técnicas para mejorar la calidad es el siguiente.

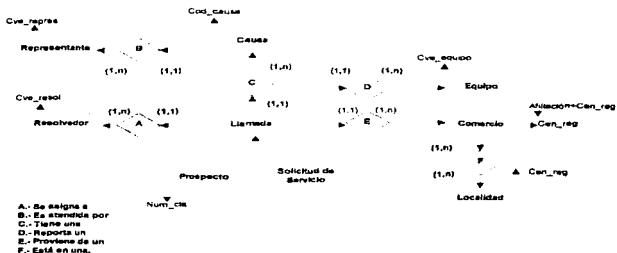


Figura 3.8 Modelo de datos depurado (Vista final)

3.12. DISEÑO FUNCIONAL EN EL DISEÑO DE BASE DE DATOS

El análisis funcional es necesario para el modelado de trabajo dentro de una empresa. Una función es sólo una porción de la empresa. El análisis funcional se centra en el entendimiento de cómo cada función usa la información y cómo es intercambiada entre las funciones. El análisis funcional es el primer paso hacia la especificación y el diseño de los programas de aplicación que operan sobre la base de datos.

La especificación de funciones y flujos de información es muy importante para el diseño conceptual de datos. Permite verificar que todos los datos que requieren las funciones estén contenidos en la base de datos y todas las operaciones en la base de datos son realizadas por algunas funciones. Además ayuda a desarrollar una visión procedimental de la manera en que se usa la base de datos.

Análisis funcional usando el modelo de flujo de datos.

El análisis funcional se encarga del modelado de un sistema de información en términos de actividades o procesos y flujos de información entre ellos. El resultado del último análisis funcional es un esquema funcional, que contiene una representación de actividades, flujos de información y otros rasgos. El esquema funcional incluye una representación de aplicaciones de base de datos y las interacciones entre ellas. Para éste se usa la misma progresión y organización de argumentos que para el modelo conceptual de datos. Primero se introduce un modelo para el análisis funcional y progresivamente se aplican transformaciones primitivas para la construcción de esquemas funcionales, estrategias para dominar el diseño y mantenimiento de esquemas funcionales.






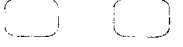

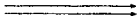
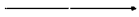



El modelo de flujo de datos sustenta los conceptos de proceso, flujo de datos, almacén de datos e interfaz:

1. **Proceso.** Representa una actividad dentro del sistema de información. El proceso puede actualizar, destruir y en general manipular la información.
2. **Flujo de datos.** Es un intercambio de información entre procesos. Éstos indican datos que fluyen hacia adentro o hacia afuera del proceso.
3. **Almacén de datos.** Es un depósito de información. Los archivos temporales, tablas, formularios de papel, formularios electrónicos, etcétera.
4. **Interfaz.** Usuario externo de un sistema de información, que puede ser emisor o receptor de los flujos de datos.

Para describir un diagrama de flujo de datos (DFD) se toma la perspectiva de la organización que usará el sistema de información, por lo que todas las actividades que contribuyen a la manipulación de datos dentro de la organización se representan con procesos. Las interfaces representan las fronteras del sistema de información, esto es los generadores y receptores de la información dentro del sistema.

Primitivas para el análisis funcional

El análisis funcional tiene el objetivo de producir un esquema funcional de las funciones (procesos) de la empresa. El diseño de un DFD se desarrolla iterando varias transformaciones de esquemas. De la misma forma que en el diseño conceptual de datos las transformaciones más simples se les denomina primitivas, las cuales pueden clasificarse cómo ascendentes o descendentes, que son:

T1: Descomposición de proceso con flujo intermedio		
T2: Descomposición de proceso con almacén intermedio		
T3: Descomposición de proceso sin conexiones		
T4: Descomposición de Flujo		
T5: Refinamiento de flujo		
T6: Descomposición de almacén		

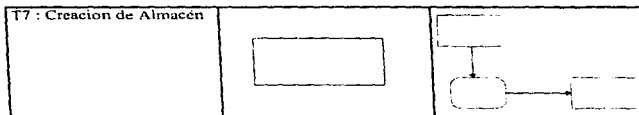


Figura 3.9 Primitivas para el análisis funcional

3.13. ESTRATEGIAS PARA EL ANÁLISIS FUNCIONAL

Al igual que en el diseño de datos se cuenta con estrategias de análisis para alcanzar un buen nivel de refinamiento, y de entre las que destacan:

Estrategia descendente

En éste caso consiste en la aplicación de primitivas descendentes puras; cada aplicación produce un refinamiento de un solo concepto. Como el concepto de proceso es el más importante en los DFD, es vital indicar cómo se realizan la descomposición de un proceso en los subprocesos. El criterio más importante se denomina independencia funcional. La idea clave es descomponer cada proceso en subprocesos bien diferenciados que sean, en lo posible, independientes entre sí. En otras palabras: cada proceso debe identificarse claramente y las conexiones entre los subprocesos deben ser débiles.

Estrategia ascendente

Con una estrategia ascendente, los DFD se construyen a partir de una colección de conceptos elementales, forjando las conexiones entre ellos. Los pasos a seguir son:

- Inicialmente se incluyen en el DFD todos los conceptos elementales
- Después se incluyen todas las interfaces. También se conectan éstas a los procesos que intercambian información con ellos
- Finalmente, se interconectan los procesos mediante flujos de datos. En particular, se introduce un almacén de datos cuando la información se registra permanentemente.

Estrategia centrífuga

Se puede aplicar naturalmente al análisis funcional, por que corresponde a seguir el tratamiento de información que realizan progresivamente los procesos. El término más apropiado para esta estrategia es, quizá, centripeta. Es decir, de afuera hacia adentro, ya que la aplicación más natural de ésta comienza con las interfaces y determina progresivamente los procesos que participan en la producción de los flujos intercambiados con las interfaces. Éste proceso se realiza hacia adelante si se parte de las interfaces que dan alguna información como entrada al sistema; se realiza hacia atrás si se comienza con las interfaces que reciben información que es una salida del sistema. La estrategia hacia atrás se denomina también "orientada a la salida", y es típica de muchas metodologías para el análisis funcional.

Estrategia mixta

Una estrategia mixta implica producir inicialmente un esquema almacén que incluya los principales procesos y luego realizar un diseño descendente de cada uno de esos procesos para producir varios DFD pequeños. Por último, los DFD se integran en uno solo. El proceso de integración es dirigido por el esquema almacén. Como en el análisis de datos, una estrategia mixta es conveniente sobre todo si el problema inicial es bastante complejo, porque permite un enfoque de "Divide y Vencerás".

Diseño de esquema funcional

Una metodología para el análisis funcional debe emplear una combinación apropiada de las estrategias expuestas. En particular, debe usar la estrategia mixta para el modelo inicial de un DFD Almacén, con el fin de descomponer el problema. Cada proceso del esquema almacén se debe entonces diseñar usando la estrategia más apropiada; se sugieren las descendentes o las centrifugas como las más convenientes. La metodología se puede esbozar así:

1. **Identificar interfaces.**
2. **Identificar flujos de entrada/salida entre interfaces y el sistema**
3. **Diseñar un esquema de almacén**
4. **Repetir**
5. **Refinar los procesos del esquema almacén por medio de actitudes de diseño descendentes, ascendentes centrifugas o mixtas, hasta que todos los conceptos de los requerimientos se hayan expresado en el esquema, sin describir rasgos procedimentales.**
6. **Verificar las cualidades del esquema: Independencia, corrección, exactitud, legibilidad y minimalidad**

Se considerará brevemente la condición de terminación de análisis funcional. En principio se puede proceder refinando los procesos hasta un nivel elemental. Si se consideran los procesos que en última instancia serán ejecutados por la computadora, el refinamiento podrá repetirse hasta que cada proceso concuerde con un procedimiento o incluso una instrucción elemental. Obviamente, esto no llevará a un buen esquema funcional, por razones metodológicas y prácticas. "En primer lugar, el análisis debe concentrarse en definir que hace el sistema de información, no en cómo opera dicho sistema". Tales consideraciones se dejan para la fase subsecuente del diseño funcional. Por lo tanto, el esquema funcional resultante del análisis no debe contener aspectos procedimentales, que indican cómo se realiza un proceso. En segundo lugar no es práctico usar redes demasiado complejas de procesos pequeños, pues no son útiles porque no proporcionan una visión global manejable del sistema de información. La primera consideración sugiere un criterio para terminar el análisis funcional: no seguir refinando los procesos cuando ello introduce una descripción procedimental.

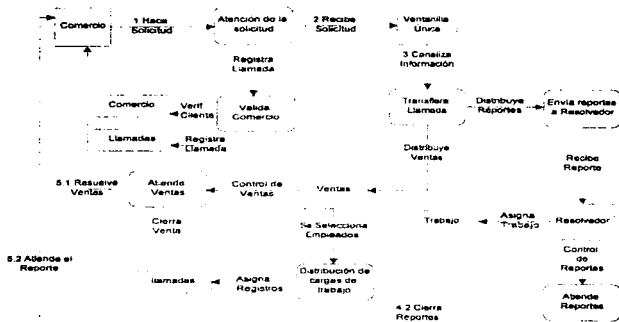


Figura 3.10 Diagrama de Flujo de Datos

3.14. ANÁLISIS CONJUNTO DE DATOS Y FUNCIONES

Se usan los términos de esquema D para designar un esquema conceptual del modelo de la base de datos, y un esquema F para denotar un esquema de funciones. La idea básica de la metodología de diseño conjunta es alternar los refinamientos del esquema D y el F, una verificación de congruencia mutua asegura que las dos especificaciones son coherentes. La prueba de la congruencia mutua se facilita por el uso de esquemas externos, que describen la estructura de los almacenes de datos y los requerimientos de datos de los procesos.

En seguida, los procesos se analizan para determinar las operaciones de base de datos, que son pequeñas unidades de interacción con la base de datos; por otro lado una operación de base de datos puede ser usado por varios procesos. Las operaciones se especifican usando esquemas de navegación: esquemas ER anotados con símbolos especiales que muestran cómo se obtiene el acceso a las entidades y cómo se recorren las relaciones.

Esquemas externos para los diagramas de flujo de datos

Un esquema externo se usa para designar una vista particular de la base de datos que se presenta en una aplicación específica o en un grupo de aplicaciones. Así, en la terminología ER, un esquema externo puede omitir porciones enteras del esquema conceptual e incluir solamente aquellas entidades e interrelaciones que se utilizan en la aplicación.

El uso de esquemas externos es propuesto durante el diseño funcional. En este marco, un esquema externo es un esquema ER que incluye únicamente los datos que conciernen a una porción específica de un DFD; en concreto, se propone asociar un esquema externo con alguno de los procesos y almacenes de datos de un DFD.

Cada esquema externo se representa con un hexágono conectado con uno de los elementos de DFD. La introducción de esquemas externos durante el análisis tiene varias ventajas:

1. Ayuda a identificar los datos que deben llegar a formar parte de la base de datos tal conocimiento puede influir sobre los refinamientos del esquema D.
2. Los esquemas externos sirven para realizar cuidadosas verificaciones de Compleción del esquema D, pues asegura que todos los conceptos de los esquemas externos estén también mencionados en el esquema D.
3. Los esquemas externos pueden llegar a ser útiles para una fase subsiguiente del diseño: la identificación y especificación de las operaciones de base de datos.

Metodología para el análisis conjunto de datos y funciones

La idea básica de esta metodología de diseño es alternar los refinamientos del esquema conceptual (de base de datos) y el esquema funcional. Los pasos de esta metodología son tres:

1. Se extienden los diagramas de flujo para incorporar los a los esquemas externos.
2. Introducir una metodología conjunta para el diseño de datos y funciones.
3. Finalmente se muestra el uso de esquemas externos como punto de partida para especificar operaciones de base de datos esenciales para el diseño lógico y físico.

Cuando realizamos los refinamientos es importante efectuar una prueba de congruencia mutua para asegurarnos de que las dos especificaciones son coherentes y proceder así a la realización de operaciones de bases de datos. Estas se especifican mediante esquemas de navegación que utilizan los esquemas externos como punto de partida.

Esquemas externos para los diagramas de flujo de datos

Los esquemas externos deben incluir únicamente los datos concernientes a la aplicación o aplicaciones consideradas. Generalmente los esquemas externos se definen para bases de datos en operación es decir, no se usan durante el diseño conceptual de base de datos.

En el ejemplo que utilizaremos en esta sección el esquema externo se usa durante el diseño funcional. En éste marco un esquema externo es un esquema de entidad - relación que incluye solamente los datos que concierne a una porción específica de un diagrama de flujo de datos. A continuación mostramos un diagrama de flujo y su esquema externo respectivo.

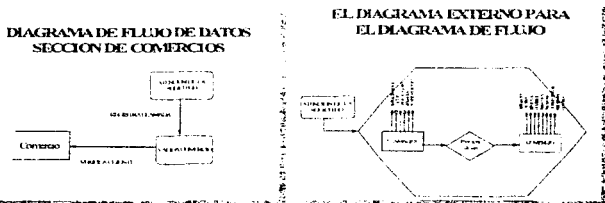


Figura 3.11 Diagrama del análisis conjunto de datos y funciones

Cómo observamos el esquema externo está enmarcado por un hexágono conectado con uno de los elementos del diagrama de flujo. Introducir esquemas externos durante el análisis funcional tiene varias ventajas:

- Ayuda a identificar los datos que llegan a formar parte de la base de datos. Esto puede influir sobre los refinamientos de esquema de datos.
- Los esquemas externos sirven para realizar cuidadosas verificaciones de Compleción del esquema de datos debido a que aseguran que todos los conceptos de los esquemas externos estén mencionados en el esquema de datos.
- Los esquemas externos pueden llegar a ser útiles para una fase subsecuente del diseño: la identificación y especificación de las operaciones de la base de datos.

Metodología para el análisis conjunto de datos y funciones

La razón primordial para un análisis conjunto es que cada planteamiento puede apoyar y completar al otro. El proceso de analizar un sistema de información se realiza por medio de refinamientos progresivos y éstos se simplifican si las especificaciones de los datos y las funciones se producen conjuntamente. Así que los refinamientos de los esquemas de datos pueden sugerir refinamientos a los esquemas funcionales y viceversa.

Una razón muy importante para adoptar un enfoque integrado es que nos ayuda a alcanzar la completación mutua. Es decir, podemos estar seguros de que los esquemas de datos y funciones son mutuamente completos haciendo una comparación cruzada de su contenido de información. A continuación mencionamos las fases de una metodología de análisis de datos y funciones:

1. **Diseño inicial.** Determinar los diagramas de flujo y de funciones, teniendo prioridad el diagrama de funciones y concentrando la atención en la descripción de entradas - salidas del sistema, identificando todas las interfaces y flujos de entrada - salida.
2. **Refinamientos progresivos descendentes.** Realizarlos en paralelo para los diagramas de datos y de funciones. El proceso de identificar los refinamientos de cada esquema está influido por las versiones previas de ambos.
3. **Versiones de Completación mutua.** Frecuentemente realizadas durante el diseño conjunto (quizá después de cada refinamiento). La completación mutua se define cómo:
 - El diagrama de datos está completo en relación con el esquema F si cada concepto expresado implícitamente en los flujos y almacenes de datos del esquema de funciones aparece en el esquema de datos. Para demostrar la completación del esquema de datos resulta muy útil considerar los esquemas externos y verificar que cada concepto mencionado en cualquiera de ellos también esté en el esquema de datos.
 - El esquema de funciones está completo si cada operación de recuperación o manipulación que deba ser realizada con los datos es en verdad ejecutada por un proceso del esquema de funciones. En particular, para cada elemento de datos almacenados debe haber un proceso encargado de su creación y se espera que algún proceso se ocupe de su recuperación y eliminación.
4. **Análisis final de los esquemas de datos y de funciones.** Aquí probamos las cualidades de los esquemas y puede ser que se realice la reestructuración de algunos esquemas.

Refinamientos mutuos

La principal ventaja del diseño conjunto de esquemas de datos y funciones es la posibilidad de alternar los refinamientos de cualquiera de los esquemas de acuerdo con el estado actual de ambos. Se consideran los cambios elementales que pueden ocurrir en cualquiera de los esquemas y sugerirán cambios elementales que pueden ocurrir en cualquiera de los esquemas y sugerir refinamientos que correspondan al otro.

Esquema de navegación para operaciones con bases de datos

Los esquemas de navegación son útiles para las fases subsecuentes del diseño de base de datos, particularmente para el diseño lógico. A continuación mostramos la metodología para producir esquemas de navegación consistente en dos etapas:

1. Determinar las operaciones de base de datos (interrelaciones individuales con la base de datos).
2. Para cada operación.
 - a) Determinar el esquema de la operación, que es el subconjunto del esquema que contiene todos los datos mencionados por la operación.
 - b) Determinar el esquema de navegación (nuevo esquema con flechas) que indican las operaciones de acceso de base de datos realizadas.

La primer etapa consiste en identificar las operaciones de base de datos. Como sabemos, las operaciones se realizan como parte de las actividades que tiene lugar dentro de un proceso. En consecuencia, para identificar las operaciones se necesita detectar los procesos que las producen. Para cada operación identificada se dibuja un esquema de operación. Este contiene todos los elementos del esquema externo en el esquema de especificación o requeridos para la navegación. Todo esquema debe incluir:

1. Atributos que se usan para las condiciones que gobiernan el acceso a las entidades e interrelaciones, o su elección
2. Atributos que usan las operaciones de recuperación o actualización
3. Los símbolos R (Leer), M (Modificar), I (Insertar) y D (Suprimir) dentro de las entidades o interrelaciones.

Éstos esquemas son un medio de indicar gráficamente que es lo que esta implicando en una operación de alto nivel.

CAPITULO 4. INFRAESTRUCTURA DE HARDWARE POR APLICAR

4.1 EQUIPO REQUERIDO PARA EL PROYECTO

Levantamiento de información

La realización de este proyecto requiere hacer un levantamiento de información que nos permita obtener datos de la situación actual del cliente, con que se podrá medir posteriormente la infraestructura de voz y datos requerida. Para obtener esta información necesitamos un cuestionario que llamaremos Check List, el cual se describe a continuación:

Figura 4.1 Check List

CHECK LIST

1. Ubicación de la empresa y su anexo

La sede de la empresa se encuentra en :

Prof. Aldama #188, C-58, MZ. 5

Colonia: Mistones de la Noria.

Delegación: Xochimilco.

México D.F.

Cuenta con un anexo en :

Av. México Xochimilco, 3500

Tercer piso.

Colonia la Noria

Delegación: Xochimilco.

México D.F.

El proyecto únicamente pretende incorporar la infraestructura necesaria para atender llamadas en el centro de soporte a clientes.

SECCIÓN DE VOZ

2. ¿Con qué frecuencia realizan llamadas de Larga distancia?

Esporádicamente.

3.- Comportamiento de su tráfico de llamadas.

3.1 Tráfico entrante provocado: Cero.

3.2 Cuál es la Hora Pico? La Hora Pico es de 11 a 12 horas.

CHECK LIST

4. Número de llamadas en la hora pico y flujo que tienen

A) Para la parte de Operación del departamento de Soporte a Clientes: Número de llamadas es 80. El flujo de éstas es únicamente de entrada y son externas. La duración de la llamada es de aproximadamente de tres minutos.

B) Para la parte de control y administración del departamento de Soporte a Clientes. Son 30 llamadas, y de tiempo indefinido.

El Flujo de éstas de Entrada y Salida (Internas y Externas) es el siguiente:

- Entrada Interna: 0
- Salida Interna: 0

- Entrada Externa: 15
- Salida Externa: 15

Duración de la llamada aproximadamente de siete minutos.

5. ¿Cuántas líneas directas y/o troncales tiene contratadas e instaladas en el sitio?

Actualmente contamos con siete líneas directas.

6. Marcas y modelos de los equipos instalados:

Son Ericsson, de los que instala TELMEX.

SECCIÓN DE DATOS

7. ¿Actualmente cuentan con una red en esa localidad?

No, actualmente se cuentan con tres PC, de las cuales una la utiliza el contador, otra el ejecutivo de control operativo y la última el gerente del área.

8. ¿Cuáles son los flujos de información?

El flujo de información se lleva a cabo inicialmente entre las áreas de mantenimiento e instalaciones y la de soporte a clientes, para la operación diaria que van del edificio principal al anexo y viceversa.

9. ¿Cuáles es la hora pico de la transferencia de información?

Las horas pico son: a las 12 y a las 17 horas

10. ¿En qué medio y en que tiempo viaja la información?

La información se envía mediante documentos, y es enviada mediante mensajería interna.
El tiempo de entrega es de aproximadamente 15 minutos como máximo.

4.2 ANÁLISIS Y DISEÑO DE LA INFRAESTRUCTURA DE VOZ

Partiendo de la información obtenida realizaremos los cálculos para darle dimensión a la infraestructura de voz de las dos secciones del área de soporte a clientes. A continuación calcularemos para la parte de " *Operación del departamento* " , sabemos que la cantidad de llamadas que se esperan recibir en la hora pico es de 80 y estas únicamente son de tráfico externo, de entrada¹. Además sabemos que la duración promedio de las mismas es aproximadamente es tres minutos, por lo que mediante una multiplicación obtenemos el tráfico telefónico².

80 llamadas X 3 minutos. / llamada = 240 minutos.

Tráfico Telefónico actual

Sabemos que la unidad del tráfico mínima de un recurso telefónico es un C.C.S. (Centum Call Seconds). Esta unidad consiste en 100 segundos de utilización telefónica. También sabemos que:

1 CCS = 1.67 minuto de tráfico

Para poder calcular el número de troncales, es necesario manejar las unidades de medida del tráfico telefónico, por lo que de los 240 minutos obtenidos necesitamos conocer su equivalente en CCS.

1 CCS es a 1.67 minutos

X CCS es a 240 minutos

Entonces $X = 240 \text{ min} \cdot \text{CCS} / 1.67 \text{ min} = 143.71 \text{ CCS}$

El tráfico telefónico debe estar expresado en la unidad telefónica de Erlang³, para poder encontrar su equivalencia en Troncales. Sabemos que:

36 CCS = 60 minutos de tráfico = 1 Erlang. Por lo que si :

36 CCS = 1 Erlang

¹Tráfico Externo de Entrada. Es el volumen de utilización de un recurso proveniente de una llamada del exterior.

²Tráfico Telefónico: Es el volumen acumulado de la utilización de un recurso telefónico (Lineas, Troncal o Extensión), considerando su duración y número de llamadas.

³Erlang. Es una Hora de utilización telefónica, la cual equivale a 36 CCS.

$$143.71 \text{ CCS} = Y \text{ Erlang}$$

$$\text{Entonces } Y = 143.71 \text{ CCS} \cdot \text{Erlang} / 36 \text{ CCS} = 3.99 \text{ Erlangs}$$

Buscando la equivalencia en tablas y tomando un factor de bloqueo de 1% tenemos que 3.99 Erlangs equivale a 10 troncales (Todas las tablas se presentan en el anexo 1).

1	0.01	0.02
2	0.15	0.22
3	0.45	0.59
4	0.86	1.07
5	1.35	1.62
6	1.89	2.23
7	2.48	2.88
8	3.10	3.55

Figura 4.2 Tabla de factor de bloqueo

Para la parte de *control y administración del departamento* de Soporte a Clientes, aplicamos los mismos criterios y obtenemos que:

$$\text{Tráfico telefónico } 30 \text{ llamadas} \times 7 \text{ minutos} / \text{llamada} = 210 \text{ minutos}$$

$$1 \text{ CCS} = 1.67 \text{ minutos}$$

$$\times \text{ CCS} = 210 \text{ minutos}$$

Entonces

$$X = 210 \text{ min} \cdot \text{CCS} / 1.67 \text{ min} = 125.7 \text{ CCS}$$

$$1 \text{ Erlang} = 36.0 \text{ CCS}$$

$$Y \text{ Erlang} = 125.7 \text{ CCS}$$

$$\text{Entonces } Y = 125.7 \text{ CCS} \cdot \text{Erlang} / 36 \text{ CCS} = 3.49 \text{ Erlangs}$$

Buscando la equivalencia en tablas y tomando un factor de bloqueo de uno por ciento, tenemos que 3.49 Erlangs equivale a nueve troncales. Uniendo los resultados obtenidos se encontró que son requeridas 19 troncales para toda el área de Soporte a Clientes incluyendo sus dos departamentos.

4.3 EQUIPO SUGERIDO PARA LA RED DE VOZ

Como concluimos en la sección anterior, se requieren 10 Troncales (TK⁴) para el departamento de operación y nueve TK para los departamentos de control y administración. A partir de esta información se harán las propuestas que consideramos cubran los requerimientos de SERVICOMP en cuestión de voz.

Propuesta 1

Consiste en la utilización de un conmutador (PBX: Privated Branch Exchange), con facilidades de ACD⁵ (Automatic Call Distribution) al cual se conectarán las diez extensiones para el centro de operación y las nueve restantes para la parte de control y administración de la empresa, como se muestra en este esquema:

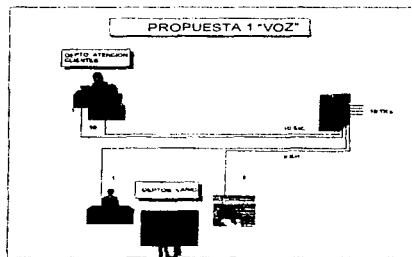


Figura 4.3 Propuesta 1 de la red de voz

La figura anterior muestra el esquema que se tendría para la parte de comunicación de voz. Cabe mencionar que el PBX tiene la facilidad de ACD. Lo que quiere decir que las llamadas se distribuirán a todas las operadoras en forma equilibrada. De manera que las llamadas sean atendidas a la brevedad posible y para que la carga de trabajo para las operadoras sea uniforme. Además se podrán generar estadísticas para efectuar la toma de decisiones.

⁴TK = Troncal

⁵ACD (Automatic Call Distribution): Aditamento del conmutador que distribuye las llamadas para que la distribución de trabajo sea en forma más equitativa y automática

Propuesta 2

Esta propuesta consiste en utilizar un equipo multilínea en lugar de un conmutador. El multilínea deberá estar dedicado a la parte de operación del área de atención a clientes, mientras que para la parte de control y administración se conectarán líneas directas, como muestra este esquema:

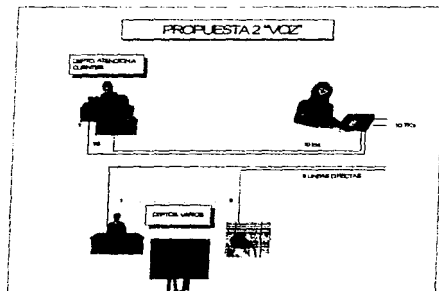


Figura 4.4 Propuesta 2 de la red de voz

Como se observa en la figura anterior, la operadora tendrá que recibir la llamada y enviarla con las operadoras desocupadas, a fin de ser atendida. La distribución se realiza de manera manual dando pie a que el tiempo de atención por llamada se incremente.

Teniendo las alternativas mencionadas, sólo resta realizar un análisis para mostrar la opción que nosotros recomendamos y que será de enorme utilidad para el buen funcionamiento de la empresa, dicho análisis se explica en la sección 4.7 de este capítulo.

4.4. ANÁLISIS Y DISEÑO DE LA INFRAESTRUCTURA DE DATOS

Para proponer una red de datos lo principal por evaluar son los siguientes dos factores:

- Topología de la red.
- Características que debe tener el equipo para soportar la aplicación adecuadamente.

Topología De La Red

Para seleccionar la tecnología de red por utilizar, tenemos que conocer a qué tipo de red pertenece. En la actualidad existen tres tipos de redes que son:

Redes Locales: LAN	<ol style="list-style-type: none">1.- Tecnología homogénea : El protocolo que se utilice en la red y el medio de transmisión a lo largo de la misma deberá ser la idéntica en toda la red.2. Alta velocidad: (0.1 a 25 Mbps.)3. Distancias cortas: 0.1 a 25 Km.4. Baja Frecuencia de Errores.- $10^{\wedge}(-8)$ a $10^{\wedge}(-12)$5. Una sola administración6. Privadas
Redes de área amplia: WAN	<ol style="list-style-type: none">1. Tecnología heterogénea.2. Distancias mayores a 25 Km.3. Velocidades bajas (1 a 2 Mbps)4. Frecuencia media de errores (menor a $10^{\wedge}(-8)$)5. Administración distribuida6.- Privadas o públicas

Redes de área metropolitana: MAN	<ol style="list-style-type: none"> 1. Tecnología heterogénea. 2. Distancias: cobertura que no exceda el tamaño de la ciudad en la que se instaló. 3. Velocidades bajas (1 a 2 Mbps) 4. Frecuencia media de errores (menor a $10^{(-8)}$) 5. Administración distribuida 6. Privadas o públicas
Redes Interconectadas: Redes de Redes	<ol style="list-style-type: none"> 1. Tecnología heterogénea 2. Distancias: cobertura que exceda el tamaño de la ciudad en la que se instaló 3. Velocidades: hasta 150,336 Mbps 4. Frecuencia media de errores: $10^{(-12)}$ 5. Administración: a) Distribuida (en el aspecto de que cada persona puede administrar su dirección IP proporcionada) b) Una sola Administración: existe sólo un organismo "NIC" que administra las direcciones IP por asignar. 6. Públicas.

Figura 4.5 Tabla de tipos de redes y sus características

La red que diseñaremos estará en un rango de distancia no mayor a los 25 kilómetros, por lo que deberá ser de tipo LAN. Como se sabe, esta red estará formada por un conjunto de dispositivos (Hardware) que se interconectarán para formar la red.

Los estándares son un conjunto de recomendaciones que tienen como meta lograr la integración de Hardware y Software sin importar quién lo provea. Es decir, los estándares son un conjunto de protocolos y especificaciones escritas y aprobadas por algún mecanismo regulador. Las organizaciones más reconocidas en la tarea de estandarización son: CCITT¹, ISO², IEEE³, OSF⁴, entre otras.

4.5. ESTÁNDARES LAN

Para nuestro caso nosotros nos enfocaremos a los estándares de redes LAN, ya que la red por instalar estará en una área no mayor a 25 kilómetros. Los estándares de las redes locales son conocidos bajo el nombre de norma IEEE 802, algunos de los cuales se muestran a continuación:

ESTÁNDAR	DESCRIPCIÓN
802.3	Define un bus utilizando CSMA/CD
802.4	Define un bus utilizando un Token
802.5	Define un anillo utilizando un Token

Figura 4.6 Tabla de estándares IEEE

En seguida mostramos los estándares más utilizados, que son el 802.3, 802.4 y 802.5, de los que daremos una breve explicación.

IEEE 802.3

El estándar 802.3 describe una familia completa de redes locales de datos que operan a velocidades de 1 a 10 Mbps en varios medios de transmisión y definen un bus utilizando un control de Acceso al medio llamado CSMA/CD. Cabe mencionar que todos los desarrollos hechos con la norma 802.3, incluyendo Ethernet, utilizan una codificación conocida como "Manchester", directamente en el medio de transmisión.

¹ Protocolo: Conjunto de reglas mediante las cuales se da un servicio

² CCITT: Comité Consultivo Internacional de Telegrafía y Telefonía

³ ISO: International Standards Organization

⁴ IEEE: Institute of Electrical and Electronics Engineers

⁵ OSF: Open Software Foundation

ALTERNATIVAS DEL 802.3 AL NIVEL FÍSICO

La red especificada por Xerox, Intel y DEC corresponde a una sola aplicación de Ethernet, en la que se basa el estándar 10 Base 5 de la norma 802.3. Posteriormente varios fabricantes han decidido poner en uso variantes de la norma, respondiendo a diferentes necesidades de los usuarios. Los parámetros por comparar para cada estándar basados en esta norma son principalmente tres:

- La velocidad de transmisión.
- La técnica de señalización.
- El tamaño de segmento de red.

El cuadro que se muestra a continuación ilustra las principales diferencias entre algunos estándares de la norma 802.3.

10 Mbps	En Banda Base	500 Mts.
10 Mbps	En Banda Base	100 Mts. ¹¹
10 Mbps	En Banda Base	200 Mts.
10 Mbps	En Banda Ancha	1,800 Mts.

La norma 802.3 se configuran en las siguientes topologías:

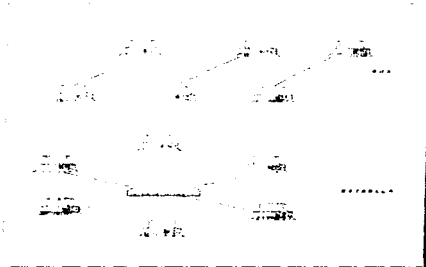


Figura 4.7 Estándar 802.3

¹¹ Antes era a 200 Mts., pero se redujo a 100 Mts., debido a que se utiliza cable delgado y es más propenso al ruido (Revista Soluciones Avanzadas Agosto 1993).

IEEE 802.4

El estándar 802.4 define una red local de datos utilizando un acceso al medio Token passing para una topología tipo bus. El Token viaja en toda la red en forma de anillo lógico. Es decir, a las estaciones se les asignan posiciones lógicas en una secuencia ordenada y, por consiguiente a continuación de la última estación encontramos la primera.

Otras definiciones nos dicen que en este estándar las estaciones van conectadas al bus o estrella. Esto es así debido a que la falta de espacio nos impide tender el bus. Como se sabe, se deben tener consideraciones de distancia para conectar una terminal de otra (para obtener parámetros físicos como impedancia, voltaje, etcétera) por lo que se requieren utilizar dispositivos que nos proporcionen los parámetros requeridos. A estos dispositivos los llamamos concentradores y cuando en una red en lugar de un bus utilizamos un concentrador decimos que la topología "es estrella".

Generalmente a este estándar se llama Token Bus y como mencionamos anteriormente, lógicamente forma un anillo. Por esta razón el orden físico de las estaciones en el medio no tiene que ver con el orden lógico.

Existe un paquete o ficha de control llamado "Token", que controla el uso del medio. A la estación que tiene el Token se le otorga un periodo de tiempo en el que se pueden transmitir uno o más paquetes y recibir respuestas. Cuando termina el tiempo o no tiene más que transmitir pasa el Token a la siguiente estación.

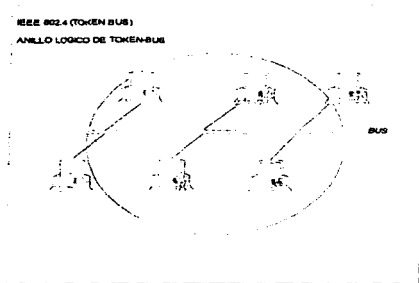


Figura 4.8 Estándar 802.4

IEEE 802.5

El estándar 802.5 define una red local de datos basada en el método de acceso al medio Token passing a 4 ó 16 Mbps para una topología tipo anillo (el más común es el diseñado por IBM llamado Token Ring).

A continuación se describe el funcionamiento de una red Token Ring con las características siguientes:

Una red con topología de anillo está integrada por un conjunto de estaciones conectadas en serie, por medio de enlaces (unidireccionales) punto a punto, de manera que se forma una trayectoria cerrada o anillo que permite su intercomunicación. El protocolo usado es el Manchester.

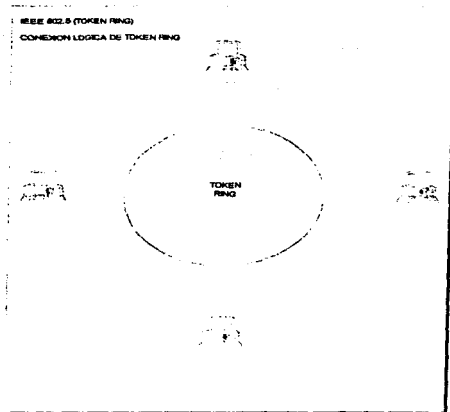


Figura 4.9 Estándar 802.5

Recomendaciones para la red de datos

Debido a que la carga de trabajo de la red no es mucha y que además se requiere de una velocidad de transmisión mayor o igual a 4 Mbps, nosotros recomendamos instalar una red Ethernet, aunándole a las ventajas que comentamos a continuación:

Desde los inicios de las redes locales, Ethernet ha dominado el panorama. En 1992 por estadísticas abarcaba el 49 por ciento del mercado total, le sigue Token Ring con el 36 por ciento; y las otras tecnologías en conjunto cubren el 15 por ciento y FDDI el siete por ciento restante. Hace muy pocos años, las diferencias eran mucho mayores, pues Ethernet ocupaba el 79 por ciento del total. Su liderazgo se debe principalmente a estas razones:

- Es una tecnología madura, bien adaptada a las necesidades que se tenía en los ambientes de oficina.
- Es relativamente simple de aplicar. Su protocolo de acceso al medio ofrece un comportamiento aceptable en cargas de trabajo ligeras.
- La política de apertura seguida por Xerox y la gran aceptación de esta tecnología por parte de los fabricantes de equipo, han permitido que Ethernet sea el tipo de red más aceptada en ambientes heterogéneos.
- Es sumamente flexible a los cambios en la configuración de la red.

4.6. HERRAMIENTAS DE SERVICIO Y SOPORTE

Antecedentes

Este apartado se orienta a detectar las herramientas de software necesarias para el proyecto, incluyen sistemas operativos, ambiente de red, programas de aplicación y desarrollo de software.

Por otra parte ya mencionamos que SERVICOMP se divide en dos. La primera parte es un área de control de operaciones y administración. En esta última son necesarios programas de aplicación para elaborar documentos y manipular información estadística, elaborar propuestas y presentar resultados a la gerencia. Con base en lo anterior lo recomendable es montar una red donde se puedan cargar aplicaciones tales como Word, Excel, Power Point, Access y cualquier otra aplicación para PC. También es necesario mencionar que se deben programar planes de capacitación para que los usuarios aprendan a usar todas estas herramientas, pues se ha detectado una carencia importante de cultura computacional en el área.

Selección del sistema operativo de red adecuado

El servidor que se instalará es un Servidor NT¹², las estaciones de trabajo que se quieren adquirir son PC en una Red local (LAN¹³) bajo una arquitectura Cliente - Servidor¹⁴. Con base en esto lo recomendable es un sistema operativo de red que permita el control adecuado del servidor cuyas principales características son:

- Multitareas
- Microprocesador de 32 bits
- Máquina con multiprocesador
- Es posible la Interacción con otros sistemas operativos
- Sistema operativo de red

A partir de esto el sistema operativo del servidor debe cubrir todos estos puntos. Claros ejemplos de esto son Net Ware, Windows 95 ó Windows NT, que cubren la totalidad de estos requerimientos.

¹² NT: New Technologie. Windows NT es un sistema operativo de 32 bits portable, multiproceso y de procesamiento distribuido

¹³ LAN (Local Area Network) : Red de área Local

¹⁴ Arquitectura Cliente - Servidor. En este tipo de arquitectura las máquinas servidoras normalmente corren los procesos que ofrecen uno o varios de los siguientes servicios: Servidor de archivos, impresión, correo electrónico, etcétera; mientras que las máquinas clientes son quienes ejecutan los programas de los usuarios y desde donde se solicita servicio a los equipos servidoras.

Por otra parte, cada estación de trabajo deberá tener a su vez su propio sistema operativo que puede ser Windows 3.1, Windows para Trabajo en Grupo, Windows 95, DOS, .

Una excelente opción son LAN Manager o Net Ware dentro de las aplicaciones comerciales existentes. El que proponemos es el LAN Manager, a partir de las siguientes consideraciones:

- Primer sistema operativo de red para cómputo Cliente - servidor en el mercado
- El Servidor no es un sistema operativo dedicado, pues corre como una aplicación por encima del sistema operativo huésped (UNIX, VMS, OS/2 y Windows NT)
- El cliente o estación de trabajo puede correr bajo DOS o Windows, OS/2, Apple System 7 y Windows NT
- Con Windows NT, LAN Manager es un sistema operativo de RED Peer - to- Peer¹¹
- Arquitectura de LAN : es una colección de estaciones de trabajo cliente y uno o varios servidores que forman un grupo lógico (dominio).
- Seguridad : Ofrece mecanismos para controlar el acceso al sistema y a los dispositivos compartidos por parte de los usuarios.
- Protección contra pérdida de datos ofreciendo discos espejo, duplicación de archivos y soporte de Unit Power Supply (UPS).

Estas son algunas de las principales razones para adoptar este sistema operativo de red pero tiene las limitaciones de no ser un sistema operativo como Windows 95 que integra conceptos como Computación Móvil, Multimedia y el estándar Plug and Play¹² ó CDFS (CD ROM File Systems), etcétera. Mas, sin embargo, para los objetivos de nuestra aplicación no serán necesarios.

En síntesis se previó para esta fase integrar en la red un sistema operativo de Red con un Windows NT en el Servidor y LAN Manager para el controlar las estaciones de trabajo. Además se contará con las siguientes aplicaciones:

- Excel
- Word
- Power Point
- Project Manager
- Microsoft Office
- Fox Pro For Windows
- Aplicación para el control y atención de llamadas y servicio al cliente (sistema por desarrollar).

¹¹ Per to per: Cada máquina puede jugar tanto el rol de cliente como el de servidor

¹² Plug and play: Esta tecnología se caracteriza por la facultad de configurar dispositivos de un equipo en forma automática. La asignación de IRQ's (Interruption request) se hace mediante software sin necesidad de configurar el dispositivo en forma manual.

Desarrollo de un software a la medida

Se ha podido observar que la recopilación estadística de datos del área de soporte a clientes es inadecuada, debido a que aunque es confiable presenta estos problemas

- El acopio de información es lento
- Se llega a perder información
- La información estadística parcial o por un periodo es difícil de obtener
- No es muy preciso el cálculo y la obtención de niveles de servicio
- La toma de decisiones se lleva a cabo lentamente
- Medir el comportamiento del proceso y la productividad del área es algo muy subjetivo
- Es necesario medir la productividad de los empleados de atención al cliente
- Es necesario automatizar los procesos de envío de documentos y el intercambio de información

Aunque para los requerimientos señalados es posible que ya exista una solución del software de aplicación en el mercado, cabe mencionar que el usuario requiere de un software que se ajuste a estas necesidades y a reglas de negocio muy específicas, propios de la empresa y con la capacidad de expandir sus requerimientos conforme las circunstancias lo reclamen.

4.7. ANÁLISIS DE RIESGOS Y SELECCIÓN DE SOFTWARE Y HARDWARE

Conclusiones de la sección de voz

Mediante una tabla se mostrarán los puntos débiles y fuertes de cada una de las propuestas mencionadas:

Transferencia de llamadas automáticamente.	SI	NO requiere una operadora	La desventaja que se tiene al tener una operadora aumentara el tiempo de atención de la llamada
Filtrado de llamadas a los distintos departamentos o extensiones	SI Programando el conmutador para que entren las llamadas para el departamento adecuado, lo impone el uso de un software conocido como Integrated Voice Response (I.V.R)	SI Mediante la operadora	
Generan Estadísticas.	SI	NO	
Distribuye automáticamente las llamadas.	SI	NO	Si no cuentan con la opción de ACD, la distribución se hará de modo manual. Por lo que si dos extensiones están desocupadas, la operadora será quien transfiera la llamada
Costo	ALTO	BAJO en comparación con respecto a la otra propuesta	El análisis detallado de costo se hace con las cotizaciones correspondientes y queda a consideración del cliente
OTRAS OBSERVACIONES:		Probablemente se requerirán 2 ó 3 operadoras para distribuir las llamadas	

Figura 4.10 Tabla de comparación de las propuestas de la red de voz

Nosotros recomendamos la opción dos ya que satisface las necesidades de operación de la empresa a fin de brindar un excelente servicio a sus clientes; pero la desventaja de esta es que no se obtendrán históricos del equipo instalado, que en este caso es un multilínea. Esta desventaja se solucionaría solicitando a TELMEX que haga un reporte monitoreo de sus líneas y así podrán analizar su comportamiento.

Conclusiones de la red de datos

Sistema operativo

<p>Las ventajas de instalar un sistema operativo de red como Windows NT en el Servidor son las de las posibilidades de crecimiento y control de óptimo y de punta dispositivos, recursos y archivos. Manejo de Multitareas, multiprocesadores y la posibilidad interaccionar con otros sistemas operativos, etcétera.</p> <p>La inclusión de LAN Manager aporta el beneficio de contar con terminales DOS controladas por el servidor en forma independiente y se podrán compartir recursos.</p>	<p>Se requiere un entrenamiento para manejar el software y conocimientos de redes para el controlador de la red. Es necesario contratar los servicios de un administrador de red o, en su caso, capacitar personal, contratar mantenimiento y soporte para sistema lo que debe ser previsto todo, lo cual tiene un costo adicional.</p>
--	---

Por las ventajas antes mostradas seleccionamos el sistema operativo de red Windows NT en el servidor, además de incluir a LAN Manager.

Desarrollo a la Medida

<p>Desarrollo adecuado a las necesidades y reglas de negocio específicas del cliente</p> <p>Posibilidad de crecer en requerimientos a mediano y largo plazos.</p> <p>Se tiene un respaldo de sistemas por parte de la empresa con una garantía de servicio y apoyo al sistema desarrollado</p> <p>La licencia del producto es de uso único para el usuario.</p>	<p>Es más caro que las aplicaciones existentes en el mercado.</p> <p>El tiempo de entrega del desarrollo es mayor que si se adquiere la aplicación comercial</p>
---	--

Debido a la peculiaridad del negocio de la empresa SERVICOMP, nosotros recomendamos desarrollar su propio software y cubrir así, de manera creciente las necesidades de la empresa.

Este software deberá correr en una LAN Ethernet, con un servidor de 32 bits con Dieciséis Megas de memoria RAM, 1 Giga en disco duro, y un procesador pentium. El monitor que recomendamos debe ser Super VGA, y Windows NT instalado.

Debido a que el sistema operativo LAN Manager no maneja 32 bits, éste deberá de ser instalado en cada máquina de usuario final, los equipos deben contar con un disco duro de hasta 200 MB, de 4 a 8 MB en RAM, monitor Super VGA, Procesador 386 ó 486 de preferencia, tarjeta de red, y con una versión de Windows de 3.1 ó mayor.

Con la recomendaciones antes mencionadas garantizamos que la aplicación correrá en forma adecuada.

CAPITULO 5. PLANEACIÓN Y ADMINISTRACIÓN DE PROYECTOS

5.1 INTRODUCCIÓN

Los proyectos modernos arrancan desde la construcción de un centro comercial suburbano hasta poner un hombre en la luna; son impresionantemente grandes, complejos y costosos. Completar éstos proyectos a tiempo y dentro del presupuesto no es tan fácil. En particular veremos que los problemas complicados surgen al programar dichos proyectos. Con frecuencia son estructurados mediante la interdependencia de las actividades. Normalmente algunas de las actividades no se pueden realizar antes de que hayan efectuado otras. Es posible que en el trabajo se abarquen miles de relaciones de dependencia, por lo que es necesario que los administradores busquen métodos efectivos de análisis. A continuación analizaremos algunos de éstos.

5.2 MÉTODOS DE PLANEACIÓN: PERT (Program Evaluation Review Technique) Y CPM (Critical Path Method)

Antecedentes

La PERT fue desarrollada a finales de la década de los 50's por la Navy Special Projects Offices en colaboración con la empresa de consultoría administrativa de Booz Allen y Hamilton. La técnica recibió una considerable publicidad favorable para su uso en el programa de ingeniería de desarrollo del misil Polaris, un complicado proyecto que tenía 250 contratistas primarios y 9000 subcontratistas. Este proyecto tuvo éxito y fue poniendo en práctica la herramienta PERT.

El CPM fue desarrollado en 1957 por J. E. Kelly, de Remington Rand y por M. R. Watker, de DuPont. Se diferencia de la PERT en principio por los detalles de como se maneja el tiempo y el costo. En realidad, las diferencias entre PERT y CPM en la instrumentación efectiva se han ido borrando en cuanto las empresas han integrado las mejores características de ambos sistemas en sus esfuerzos por manejar con eficiencia sus proyectos.

Métodos

El primer paso del proceso para ambos métodos es definir las actividades del proyecto y establecer las relaciones de procedencia adecuadas. Este primer paso es importante puesto que los errores y omisiones pueden conducir a una programación de eventos demasiada inexacta. A continuación mostramos una tabla la cual ejemplifica una lista de actividades, cabe hacer mención que en la tabla se muestran cosas por venir como el *tiempo y recursos*.

A	Elegir local de oficinas	-	3	
B	Crear el plan financiero y organizacional	-	5	
C	Determinar requerimientos de personal	B	3	
D	Diseñar medios	A,C	4	
E	Construcción del Interior	D	8	
F	Elegir personal a mudar	C	2	
G	Contratar nuevos empleados	F	4	
H	Mudar registros, personal clave	F	2	
I	Hacer arreglos financieros	B	5	
J	Entrenar nuevo personal.	H,E,G	3	

Figura 5.1 Tabla de actividades

Un procedimiento gráfico usado en la práctica es la "Gráfica de Gantt" misma que es usada para abordar problemas de programación de un proyecto.

La gráfica de Gantt fue desarrollada por Henry L. Gantt en 1918, a la fecha sigue siendo utilizada para la producción y programación de proyectos. Su simplicidad y su claro desarrollo gráfico lo han establecido como un recurso usual para los problemas simples de programación, la cual consiste en hacer una lista de actividades en el eje vertical. El eje horizontal es el tiempo y representa tanto la duración prevista como la verdadera de cada actividad mediante una barra de la longitud adecuada. La gráfica indica también el tiempo de inicio más temprano posible para cada actividad. Por ejemplo, la actividad "C" no puede empezar antes del momento 5 ya que de acuerdo con el cuadro anterior se debe terminar la actividad "B" antes que pueda comenzar la "C". cuando se completa cada actividad se sombrea la barra correspondiente. La gráfica que a continuación mostramos es un ejemplo de la Gráfica de Gantt.

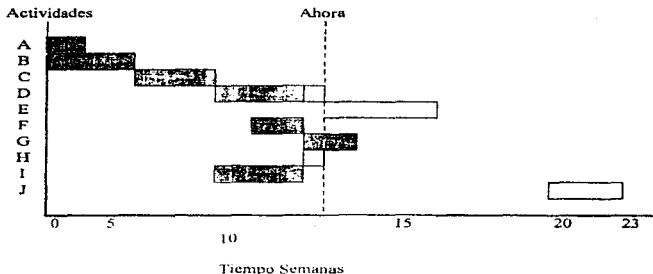


Figura 5.2 Diagrama de Gantt

En esta se muestra que las actividades D, E y H van fuera de tiempo, en tanto que la actividad G ya ha sido completada en forma total (esta sombreada totalmente) y por lo tanto va "adelante de tiempo".

La cuestión importante en un proyecto es definir si con éstos "Fuera de Tiempo" el proyecto va demorado o no en términos de una fecha de terminación como objetivo. La Gráfica de Gantt falla en revelar algo tan importante como esto. Por ejemplo la gráfica de Gantt no revela que actividades son antecesores inmediato de otras. El tipo de información de "antecesores inmediatos" es lo que se debe usar para deducir el efecto del tiempo de conclusión sobre el proyecto total, esta información la encontramos en el diagrama de red.

Diagrama de red

En un diagrama de red PERT cada actividad se representa mediante una flecha llamada arco o rama. El principio y el fin de cada actividad se representa mediante un círculo llamado nodo. Un evento representa la conclusión de las actividades que conducen al nodo. En relación a las actividades de la tabla mostrada anteriormente, vemos que "elegir local para oficina" se designa como actividad "A". Cuando se completa esta actividad, ocurre el evento "se seleccionó lugar para la oficina".

En el diagrama de red, cada actividad debe empezar en el nodo en que termina su antecesor inmediato. Por ejemplo, en el diagrama de red la actividad "C" comienza en el nodo 3, porque ahí termina su antecesor inmediato (la actividad "B").

A continuación mostramos un diagrama de red para las actividades "A" y "C". Cabe hacer mención que los números asignados a los nodos son arbitrarios (sólo se usan para identificar eventos) y no indican nada en referencia a su procedencia.

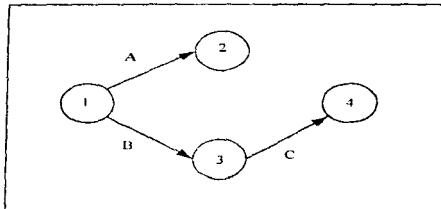


Figura 5.3 Diagrama de red para las actividades "A" y "C"

Vemos sin embargo que surgen complicaciones al agregar la actividad "D" al diagrama de red. Las actividades "A" y "C" son antecesoras inmediatas de "D" y dado que requerimos que cada actividad como "D" aparezca sólo una vez en el diagrama, se combinarán los nodos 2 y 4 para que "D" empiece en ese nuevo nodo. En la figura que a continuación mostramos, el nodo 3 representa ahora el evento del que han concluido tanto la actividad "A" como la "C".

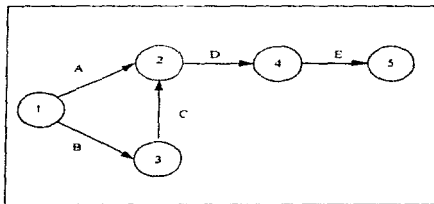


Figura 5.4 Diagrama de red parcial

Obsérvese que la actividad "E" que tiene sólo a "D" como antecesor inmediato puede agregarse sin dificultad. Sin embargo, cuando aparece un nuevo problema cuando tratamos de agregar la actividad "F" que tiene a "C" como antecesor inmediato.

Este problema de diagramación se resuelve introduciendo una actividad *ficticia* que se representa mediante una línea punteada como se muestra a continuación. Ésta no requiere tiempo ni recursos.

Esta figura indica que la actividad "D" sólo puede comenzar una vez finalizadas las actividades "A" y "C". En forma similar, la actividad "F" se puede iniciar cuando haya terminado la actividad "C".

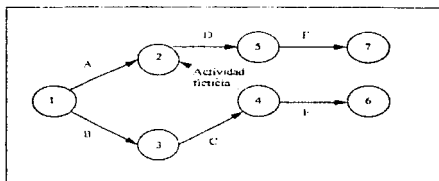


Figura 5.5 Introduciendo una actividad ficticia

Las actividades de la ruta crítica son aquellas cuya holgura es nula. Esto si lo observamos en un diagrama de red completo detectamos que las actividades críticas son: B, C, D, E, J. Es decir, las otras actividades (no críticas) pueden retardarse o tener una holgura sin provocar una demora en la terminación del proyecto. Cabe hacer mención que el tiempo mínimo de terminación de un proyecto es indicado por la duración del la ruta crítica. A continuación mostraremos el diagrama de red completo el cual contiene: ES (tiempo temprano de inicio de actividad), EF (tiempo más temprano de conclusión de una actividad), LS (Tiempo de inicio tardío) y LF (tiempo de terminación tardío).

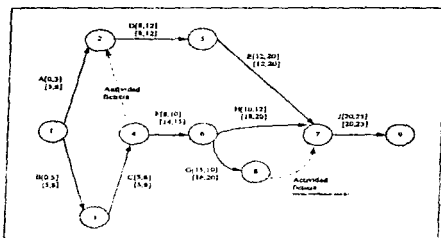


Figura 5.6 Diagrama de red (Ruta crítica)

Este análisis partió de tiempos fijos de las actividades y por lo que podemos observar para poder realizar el proyecto requerimos de 23 semanas. ¿Qué sucedería si el dueño de la empresa lo requiere en 22 semanas? Es evidente que algo debe cambiar si queremos alcanzar el objetivo. Existen dos formas básicas para lograrlo, que son:

1. **Una análisis estratégico:** En este punto podemos sugerir arreglos para efectuar algunas de estas actividades en distinta forma o ruta de como aparecen en la ruta crítica.
2. **Un tratamiento táctico:** En este método respetamos el diagrama de red y trabajamos para reducir el tiempo de ciertas actividades de la ruta crítica asignando más recursos, aunque probablemente el costo tienda a incrementarse.

El ejemplo que hemos manejado contiene tiempos de alguna actividad, y sabemos que esos tiempos están sujetos a cambios por lo que la PERT utiliza un procedimiento para calcular la probabilidad de que se efectúen las actividades en los tiempos considerados.

Método PERT

Primero, el sistema PERT para calcular el tiempo de actividad requiere de alguien que conozca bien la actividad en cuestión, a fin de producir tres presupuestos del tiempo de ésta.

1. **Tiempo optimista** (denotado por a): Es el tiempo mínimo. Todo tiene que marchar a la perfección para alcanzarlo.
2. **Tiempo más probable** (denotado por m): Es el tiempo normal, el que se necesita en circunstancias ordinarias.

3. **Tiempo pesimista** (denotado por b): Es el tiempo máximo. Una versión en la ley de Murphy. Es el tiempo que toma en cuenta todos los factores y contratiempos posibles.

Para el ejemplo de este capítulo, si queremos calcular la actividad "E" (construir el interior) y el contratista nos dio los siguientes valores: $a=4$, $m=7$ y $b=16$ y además sabemos que el cálculo del tiempo de actividad esperado se basa en una probabilidad que tiene una distribución beta unimodal, que para el tiempo esperado será de:

$$\text{Tiempo esperado de actividad} = \frac{a + 4m + b}{6}$$

Para nuestro ejemplo:

$$\text{Tiempo esperado de actividad} = \frac{4 + 4(7) + 16}{6} = 8$$

La desviación típica del tiempo de actividad se calcula a partir de que hay seis de ellas entre el tiempo optimista y el pesimista, es decir

$$\text{La desviación típica del tiempo de actividad} = (b - a)/6 \quad \text{y que la varianza} = ((b-a)/6)^2$$

De los datos anteriores podemos obtener la probabilidad de concluir el proyecto a tiempo. Es decir en 22 semanas (tiempo propuesto por los directivos) por lo que partiremos de las siguientes premisas:

$$\text{Prob } \{T \leq 22\}$$

donde T, es el tiempo a lo largo de la ruta crítica, y ésta tiene una distribución aproximadamente normal.

Tomando en cuenta que se hayan realizado ajustes en la ruta crítica y que esta sea "B", "C", "D" y "E", sumando el tiempo de éstas, tenemos que la duración del proyecto se puede efectuar en 20 semanas (sin tomar en cuenta la probabilidad), o sea $5 + 3 + 4 + 8 = 20$ semanas. Entonces encontraremos primero la varianza de T.

$$\text{Var T} = \Sigma \text{ varianza de las actividades (B, C, D y E)}$$

Sabemos que la Desviación estándar en la raíz cuadrada de las varianzas :

$$T = (\text{Var T})^{1/2} = 2.357$$

Recordando que la medida es 20 semanas y que el tiempo esperado de conclusión tomada por la dirección es de 22 semanas tenemos que:

$$(22 - 20) / 2.357 = 0.8485$$

Revisando las tablas obtenemos que habrá un 80 por ciento de probabilidad es que se complete en un tiempo menor o igual al indicado por los directivos.

Como acabamos de explicar, la PERT proporciona un medio útil para analizar problemas de planeación. Frente a la incertidumbre en relación con tiempos de actividad. Tal incertidumbre será más frecuente en los casos de proyectos nuevos únicos para los que haya poca experiencia sobre tiempos y costos que pueda ser aprovechada. Es interesante un trabajo más explícito con los costos, en el sentido de alcanzar posibilidades de desviar recursos con el objeto de reducir el tiempo de terminación.

Método CPM

El concepto de que exista un compromiso entre el tiempo que se tardará en completar una actividad y el costo de los recursos que se le destinan, es la base del modelo que fue parte original del método CPM. El modelo presupone que el costo es una función lineal del tiempo como lo muestra la siguiente gráfica:

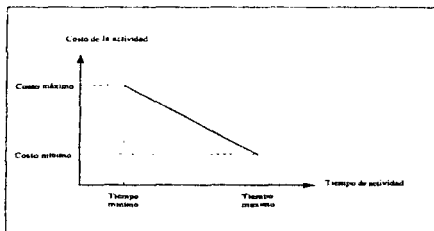


Figura 5.7 Función lineal tiempo - costo

Como se observa, el administrador del proyecto tiene la oportunidad de escoger el tiempo de la actividad en cualquier punto dentro de un mínimo y un máximo. La elección de un tiempo implica un costo en la actividad. Claro está que la elección de los tiempos de actividad afectan al tiempo de conclusión del proyecto. Ahora resta definir los tiempos más apropiados de las actividades para que se produzcan en tiempos deseados y a un costo mínimo, para lo cual se utilizará el CPM.

Para ejemplificar el método CPM supondremos que requerimos evaluar tomas potenciales (gasolineras) según la ubicación y otras características. En este método también partiremos realizando una lista de actividades.

DPI	---	32	\$ 640	20	\$ 800	12	\$ 13.33
EPI	---	40	\$ 480	30	\$ 720	10	\$ 24.00
DPA	DPI	50	\$ 1,000	30	\$ 1,200	20	\$ 10.00
EPA	---	24	\$ 288	15	\$ 360	9	\$ 8.00
INT	EPI,EPA	120	\$ 4,800	70	\$ 5,600	50	\$ 16.00

TOTAL

\$ 7,208

DPI :Diseñar procesador de información.
 EPI :Escribir procesador de información.
 DPA : Diseñar paquete de análisis.
 EPA: Escribir paquete de análisis.
 INT :Introducir sistema.

Figura 5.8 Lista de actividades

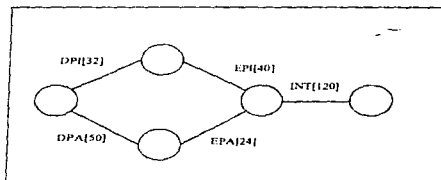


Figura 5.9 Diagrama de red para el proyecto de análisis financiero

El tiempo que aparece en la lista anterior se refiere al tiempo normal, es decir, consideramos que se pueden definirlos con precisión tiempos de las actividades. Como observamos en el diagrama anterior, la ruta más larga de la red es la DPA-EPA-INT y por lo tanto es la ruta crítica por lo que el tiempo mínimo de concluir un proyecto es de 194 horas. Cabe mencionar que la quinta columna muestra el máximo de horas de compresión que se define como:

Máximo de horas = tiempo normal - tiempo de compresión de compresión

El sistema CPM esta basado en un insumo de datos de cuatro factores para cada actividad:

1. Tiempo normal: El tiempo máximo de la actividad
2. Costo normal: El costo requerido para lograr el tiempo normal
3. Tiempo de compresión: El tiempo mínimo de la actividad
4. Costo de urgencia: El costo requerido para lograr el tiempo de compresión

A continuación daremos un ejemplo: Si queremos reducir el tiempo del DPI una hora, el costo de esta actividad será de $\$ 640 + \$ 13.33 = \$ 653.33$. La compresión del proyecto mediante CPM se realizará de la siguiente manera.

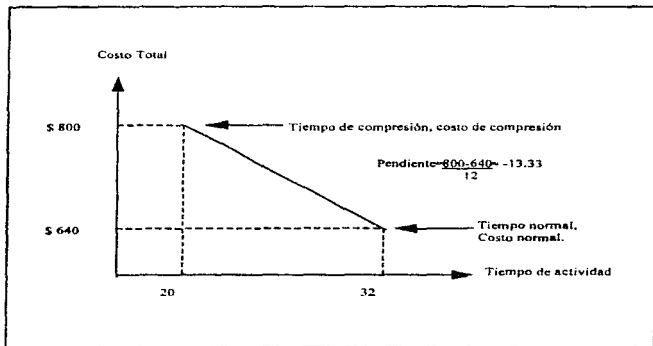


Figura 5.10

Retomando el punto inicial, el tiempo normal de la ruta crítica es de 194 horas, si nosotros necesitaráramos entregar el proyecto un día antes, es decir, en 193 horas, los pasos a seguir serían:

1. De las actividades de la ruta crítica (DPA-EPA-INT) debemos comparar, cuál es el costo por hora de urgencia que nos conviene pagar, es decir cuál de ellas nos sale más barata. Al realizar la comparación observamos que :

DPA = 10, EPA = 8 y INT = 16.

por lo que nos conviene comprimir 2 horas a EPA para alcanzar un tiempo proyectado de 192 horas con un costo total del proyecto de: $\$ 7208 + 8 = \7216

Si requiriéramos reducir el tiempo a 191 horas haríamos lo mismo; pero si lo reducimos a 190 observamos que tenemos dos rutas críticas, las cuales tendríamos que comprimir. Estas rutas son DPI-EPI-INT y DPA-EPA-INT, se muestran a continuación:

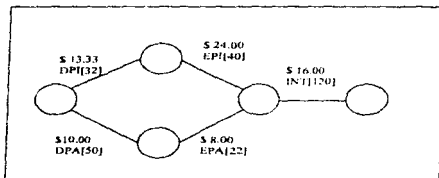


Figura 5.11 Rutas críticas

Ahora tendremos que analizar las actividades de cada ruta crítica que nos resulta más económico comprimir y como se observa en la gráfica son la actividad EPA y la actividad DPI, por lo que ahora el costo de comprimir una hora será de: $\$ 13.33 + \$ 8 = \$ 21.33$.

Otra alternativa sería comprimir INT con un costo de \$ 16.00, por lo que consideramos será la más apropiada.

Tanto el modelo de PERT como el CPM son modelos utilizados para proyectos generales, a continuación analizaremos el caso de nuestro proyecto de software.

5.3 PROYECTOS DE SOFTWARE

Retomando la metodología señalada en el capítulo 1, tenemos el punto de partida para llevar a cabo un proyecto de software. Sin embargo, aún con una metodología para el control de proyectos, el ingeniero en computación debe ser una persona con la habilidad de prever lo que irá mal antes de que ocurra, y con el coraje para hacer cálculos cuando el futuro no esté claro.

La estimación de recursos, costos y tiempos para el esfuerzo de desarrollo de software requiere experiencia, acceso a una buena información histórica y coraje para confiar en las medidas cuantitativas, cuando todo lo que existe son datos cualitativos. El cálculo conlleva un riesgo inherente. Existen varias factores importantes para la realización un proyecto:

1. *La complejidad del proyecto*, tiene gran efecto sobre la incertidumbre y va muy ligada con la experiencia del ingeniero de software.
2. *El tamaño del proyecto*, puede afectar a la precisión y al presupuesto debido a que a medida de que aumenta el tamaño, la interdependencia entre los distintos elementos de software crece rápidamente; éstos podemos descomponerlos en elementos más pequeños, pero pueden ser todavía enormes.
3. *El grado de estructuración del proyecto* crea riesgo sobre la estimación. En este contexto, la estructuración se refiere a la facilidad con la que las funciones pueden ser compartidas y a la naturaleza jerárquica de la información que debe ser procesada. En la figura que a continuación mostramos utiliza la medida recíproca de la estructuración, es decir, a medida que aumenta el grado de estructuración, la posibilidad de estimar aumenta y del riesgo disminuye.

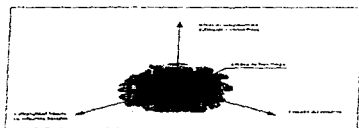


Figura 5.12 Estructuración de un proyecto

Como sabemos, el riesgo se mide por el grado de incertidumbre de las estimaciones cuantitativas establecidas para los recursos, los costos y los tiempos por lo que el planificador del software debe tener una completa información sobre la función, el rendimiento y las definiciones de la interfaz (contenidas en la especificación del sistema). El planificador y sobre todo el cliente deben tener presente que cualquier cambio en los requisitos del software significa variación en los costos y en el tiempo de entrega del sistema.

Al iniciar un proyecto, nos enfrentamos con un dilema que son las estimaciones cuantitativas, un análisis detallado de los requisitos del software proporcionaría la información detallada pero este puede durar semanas o meses, por lo que el objetivo de la planificación del proyecto de software es suministrar una estructura que permita hacer estimaciones razonables de recursos, costos y prioridades. A continuación mencionaremos las actividades asociadas con la planificación del proyecto de software.

1. Ámbito del software.
2. Recursos.
3. Estimación del proyecto de software.
4. Descomposición.
5. Herramientas automáticas de Estimación.

En seguida mencionaremos una explicación breve de los puntos antes mencionados.

Ámbito del software

Aquí establecemos explícitamente los datos cuantitativos como número de usuarios simultáneos, tamaño de la lista del correo y tiempo máximo de respuesta posible, tomando en cuenta limitaciones o restricciones como lo son el costo del producto, el hardware donde correrá la aplicación del software como lo son las interfaces, la fiabilidad del rendimiento (tiempo de respuesta) etcétera.

Recursos

La segunda tarea de la planificación del desarrollo de software es el cálculo de los recursos requeridos para realizar el desarrollo del software, en la figura que mostramos a continuación encontramos que en la base se encuentran las herramientas existentes (software y hardware) que darán soporte al desarrollo del software y en el nivel más alto encontramos los recursos humanos ya que ellos implementaran y manejaran el sistema.

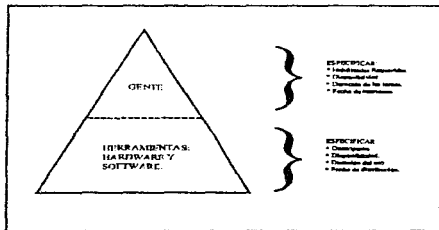


Figura 5.13 Recursos para la realización de un proyecto

Recursos Humanos.- Necesitamos que evaluar el ámbito y seleccionar las habilidades técnicas que se requieren para llevar a cabo el desarrollo con objeto de seleccionar al personal capacitado para llevar a cabo las tareas a ejecutar, también debemos definir el número de personas requeridas para el proyecto.

Recursos de Hardware.- Durante la planificación del proyecto de software deben considerarse tres categorías de hardware:

- a) El sistema de desarrollo (sistema anfitrión). Equipo por utilizar para desarrollar el software incluyendo sus periféricos.
- b) La máquina objetivo. Es el dispositivo donde se procesará el software desarrollado.
- c) El resto de los elementos que intervienen en el sistema.

Recursos de software.- Los ingenieros de software utilizan un conjunto de herramientas como el "CASE" (Ingeniería de software asistida por computadora). Además de esta existen otras como son: Herramientas de planificación de sistemas comerciales; de gestión de proyectos; de soporte, análisis y diseño, etcétera.

Reusabilidad.- Creación y reutilización de bloques constructivos de software. Tales bloques previamente serán catalogados, validados y estandarizados para ejecutar con ellos una fácil referencia, aplicación e integración.

Presupuesto del proyecto de software

El cálculo del costo y del esfuerzo del software nunca será algo exacto, por estar conformado de un extenso número de variables (humanas, técnicas, de entorno etcétera) que intervienen en el costo final del proyecto; pero para obtener una aproximación podemos recurrir a varias opciones:

1. Dejar el cálculo al final del proyecto.
2. Utilizar técnicas de descomposición relativamente sencillas para generar estimaciones de costos y esfuerzos del proyecto.
3. Desarrollar un modelo empírico para el cálculo de costos y esfuerzos del software. Este modelo se sustenta en la experiencia y toma como base:

$$d = f(V_i)$$

Donde:

d: valor estimado (esfuerzo, costo, duración del proyecto).

V_i: Son parámetros independientes Líneas de Código y los puntos de función

4. Adquirir una o varias herramientas automáticas de cálculo (aplicar una o varias técnicas de descomposición o modelos empíricos).

Descomposición

Los seres humanos hemos desarrollado un método natural para resolver problemas. Si el problema es demasiado complicado, tendemos a subdividirlo hasta lograr pequeñas partes manejables.

Herramientas automáticas de cálculo

Las técnicas de descomposición y los modelos empíricos de cálculo son aplicados con software. Estas herramientas permiten al planificador a calcular costos y esfuerzos, así como llevar a cabo pruebas del sistema. Todas estas requieren de uno o más de los siguientes datos:

1. Un cálculo cuantitativo del tamaño del proyecto (en líneas de código) o de la funcionalidad (datos sobre puntos de función).
2. Características cualitativas del proyecto, tales como la complejidad, la fiabilidad requerida o el grado crítico del negocio.
3. Alguna descripción del personal de desarrollo y/o del entorno desarrollado.

5.4 APLICACIÓN DE LOS CONCEPTOS DE ADMINISTRACIÓN DE PROYECTOS

En nuestro caso combinaremos la metodología con los puntos mencionados al inicio de este capítulo. Cabe decir que los factores citados en el apartado anterior influyen en la estrategia y costos del proyecto, pues el ingeniero en computación puede encontrar en situaciones en que tuviese que colaborar con otra empresa dentro del proyecto, o bien el sistema debe ser entregado en cierta fecha, lo que afecta directamente en las necesidades de contratar más recursos, etcétera.

Para ello la metodología de control de proyectos nos ayudará a tener un Aseguramiento de Calidad en el Software (SQA¹), también podrá ser utilizada para medir y plantear una estrategia. Por otra parte servirá de guía para realizar todas las actividades necesarias, y para saber si existen desviaciones en el proyecto. También nos permitirá partir el proyecto en varios segmentos que determinarán la complejidad y las actividades de todos los participantes del proyecto en cada proceso del mismo. De esta manera se podrán prever oportunamente inconvenientes mucho antes de que éstos sucedan.

¹ SQA Software Quality Assurance

El caso de nuestro proyecto no es la excepción pues están involucrados elementos de selección de la plataforma de hardware y software, diseño y desarrollo de un sistema a la medida, etcétera. Los cuales ya han sido resumidos en los capítulos anteriores.

A continuación se llevará a cabo la planeación del proyecto e integraremos las técnicas presentadas en este capítulo. Para obtener los tiempos y costos del proyecto. La herramienta que empleamos para obtener esta información es el MS Project. Diagramas y tablas de tiempo y costos se incluyen en el anexo de esta tesis.

CAPÍTULO 6. ASEGURAMIENTO DE LA CALIDAD Y MANTENIMIENTO DEL SISTEMA

6.1. ASEGURAMIENTO DE CALIDAD DEL SOFTWARE

Definición

La calidad del software se define como la concordancia de los requisitos funcionales y de rendimiento establecidos con los estándares de desarrollo explícitamente documentados y con las características apropiadas que se espera de todo software desarrollado profesionalmente. A continuación presentamos tres puntos importantes del aseguramiento de calidad:

1. Los requisitos del software son la base de las medidas de calidad
2. Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería de software
3. Si el software se ajusta a los requisitos explícitos, pero falla el cumplimiento de los mismos, entonces la calidad del software está en entredicho

Aseguramiento de calidad de software (SQA) es una actividad de protección que se aplica a todo lo largo del proceso de desarrollo de un sistema. La SQA engloba:

1. Métodos y herramientas de análisis, diseño, codificación y prueba
2. Revisiones técnicas formales que se aplican en cada paso del desarrollo del sistema
3. Una estrategia de prueba multiescalada
4. El control de la documentación del software y sus cambios
5. Procedimiento que asegure un ajuste a los estándares de desarrollo del software
6. Mecanismos de medida e información

Factores que determinan la calidad del software

Los factores que afectan la calidad del software se clasifican en dos grandes grupos:

1. Factores que pueden ser medidos directamente (ejemplo: Errores de codificación/ Unidad de tiempo, degradación del tiempo de ejecución / programa)
2. Factores que sólo pueden medirse cualitativamente (Facilidad de uso o mantenimiento)

Estos factores 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.

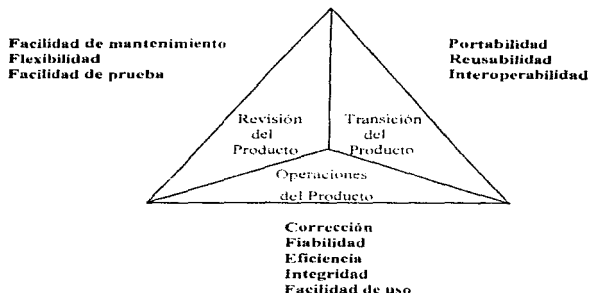


Figura 6. 1 Factores que determinan la calidad del software

Los factores indicados en la figura anterior proporciona las siguientes descripciones:

Factores de operación del producto

- **Corrección:** Grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el cliente.
- **Fiabilidad:** Grado en que se puede esperar que un programa lleve a cabo sus funciones con la precisión requerida.
- **Eficiencia:** Cantidad de recursos de computadora y código requeridos por un programa para llevar a cabo sus funciones.
- **Integridad:** Grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.
- **Facilidad de uso:** Esfuerzo requerido para aprender un programa, trabajar con él, preparar su entrada e interpretar su salida.

Factores de revisión del producto

- **Facilidad de mantenimiento:** Esfuerzo requerido para localizar y arreglar un error en un programa.
- **Flexibilidad:** Esfuerzo requerido para modificar un programa operativo.
- **Facilidad de prueba:** Esfuerzo requerido para asegurar que un programa realice las funciones requeridas.

Factores de transición del producto

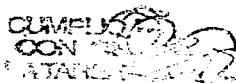
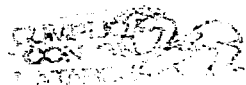
- **Portabilidad:** Esfuerzo requerido para transferir el programa desde un hardware y/o entorno de sistemas de software a otro.
- **Reusabilidad:** Grado en que un programa se puede utilizar en otras aplicaciones.
- **Facilidad de interoperación:** Esfuerzo requerido para acoplar un sistema a otro.

Métricas Del Software

Debido a que es difícil obtener una medida exacta de los factores, se han desarrollado métricas que los afectan. Aún así, éstas son muy subjetivas. Las métricas se pueden mostrar en forma de listas de comprobación, usadas para medir el grado de los atributos del software. En el esquema de graduación se usan las siguientes métricas.

- **Facilidad de auditoría :** Facilidad con la que se puede comprobar la conformidad de los estándares
- **Exactitud :** Precisión de los cálculos y del control.
- **Normalización de las comunicaciones :** Grado en que se usan anchos de banda, protocolos e interfaces.
- **Complejidad:** Grado en que se ha conseguido la total implementación de las funciones requeridas.
- **Concisión:** Lo compacto que es el programa en términos de líneas de código
- **Consistencia:** El uso de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo del software.
- **Estandarización en los datos:** El uso de estructuras de datos y tipos estándar al o largo de todo el programa.
- **Tolerancia de errores:** Daño que se produce cuando el programa encuentra un error.
- **Eficiencia en la ejecución:** Rendimiento en tiempo de ejecución de un programa.
- **Facilidad de expansión:** Grado en que se puede ampliar el diseño arquitectónico de datos o procedimientos.
- **Generalidad:** Amplitud de aplicación potencial de los componentes del programa.
- **Independencia del hardware:** Grado en que el software es independiente del hardware sobre el que opera.
- **Instrumentación:** Grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen.
- **Modularidad:** Independencia funcional de los componentes del programa

- **Facilidad de Operación:** Facilidad de operación de un programa.
- **Seguridad:** Disponibilidad de mecanismos que controlen los programas y los datos.
- **Autodocumentación:** Grado en que el código fuente proporciona documentación significativa.
- **Simplicidad:** Grado en que un programa puede ser entendido sin dificultad.
- **Independencia del sistema de software:** Grado en que el programa es independiente de características no estándar del lenguaje de programación, de las características del sistema operativo y otras restricciones de entorno.
- **Facilidad de traza:** La posibilidad de seguir la pista a la representación del diseño.
- **Formación:** Grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.



La relación entre los factores de calidad y las métricas indicadas se puede ver en la siguiente tabla

Facilidad De Auditoría				X			X			
Exactitud		X								
Normalización De Las Comunicaciones										X
Complejidad	X									
Complejidad		X				X	X			
Concisión			X		X	X				
Consistencia	X	X			X	X				
Estandarización En Los Datos										X
Tolerancia De Errores		X								
Eficiencia En La Ejecución			X							
Facilidad De Expansión						X				
Generalidad						X		X	X	X
Independencia Del HW								X	X	
Instrumentación			X	X		X				
Modularidad		X			X	X	X	X	X	X
Facilidad De Operación			X							X
Seguridad				X						
Autodocumentación					X	X	X	X	X	
Simplicidad		X			X	X	X			
Independencia Del Sistema De SW								X	X	
Facilidad De Trazo	X									
Formación										X

Figura 6.2 Tabla de los factores de calidad y métricas

Debe insistirse en que el peso dado a cada métrica dependerá de los productos particulares, así como de otros aspectos. Los factores y atributos de calidad del software descritos se pueden utilizar para establecer métricas de calidad en cada paso del proceso de ingeniería de software. los más importantes son:

1. La funcionalidad se obtiene mediante la evaluación del conjunto de características y posibilidades del programa, la generalidad de las funciones que se entregan y la seguridad de todo el sistema.
2. La facilidad de uso se calcula considerando los factores humanos, la estética global, la consistencia y la documentación.
3. La fiabilidad se calcula midiendo la frecuencia de fallos y su importancia, la eficacia de los resultados de salida, el tiempo medio entre fallos, la posibilidad de recuperarse a los fallos y la previsibilidad del programa
4. El rendimiento se mide mediante la evaluación de la velocidad de proceso, el tiempo de respuesta, el consumo de recursos, el rendimiento total de procesamiento y la eficiencia.
5. La capacidad de soporte combina la posibilidad de ampliar el programa (extensibilidad), la adaptabilidad y la utilidad, la facilidad de prueba, la compatibilidad, la capacidad de configuración y la sencillez con que se puede instalar un sistema.

El aseguramiento de calidad del software es un sistema de acciones planificado y sistemático que se requiere para asegurarla. La gente que lleva a cabo el SQA debe mirar el software desde el punto de vista del cliente.

Actividades del SQA.

El aseguramiento de calidad comprende una gran variedad de tareas, asociadas a siete actividades principales:

1. Implicación de métodos técnicos.
2. Realización de revisiones técnicas formales.
3. Pruebas de seguridad.
4. Ajuste a los estándares.
5. Control de cambios.
6. Mediciones.
7. Registro y realización de informes.

Una vez que se ha creado una especificación y un diseño, debe ser garantizada su calidad. La actividad central que permite garantizar la calidad es la Revisión Técnica Formal (RTF), la cual es una reunión del personal técnico con el único propósito de descubrir problemas de calidad.

La prueba del software combina una estrategia de múltiples pasos con una serie de métodos de diseño de casos de prueba que ayudan a asegurar una efectiva detección de errores.

Muchos grupos de desarrollo la utilizan como una red de seguridad para garantizar la calidad.

El grado de aplicación de procedimientos estándares varía de empresa a empresa. En muchos casos los estándares los determina el cliente. En otras se impone por sí solos. Si existen estándares formales, se debe establecer una actividad de aseguramiento de calidad que vea que se cumplan. El proceso de control de cambios contribuye directamente en la calidad del software, al formalizar las peticiones de cambio, evaluar y controlar la naturaleza o impacto del cambio.

La medición es una actividad integral. Su objetivo es seguir la calidad del software y evaluar el impacto de los cambios de metodología y procedimiento. Para conseguir esto, se colectan métricas de software, las cuales implican un conjunto de medidas técnicas orientadas a la gestión. El registro de la información y generación de informes para garantizar la calidad son procedimientos para recolección y divulgación de información de los procesos de aseguramiento de calidad.

6.2. REVISIÓN DEL SOFTWARE

Las revisiones se aplican en varios momentos del desarrollo y sirven para detectar errores que puedan ser eliminados. Las revisiones sirven para purificar las actividades de análisis, diseño y codificación.

Una revisión técnica formal es un filtro del proceso de SQA. La Revisión Formal Técnica se define como un medio efectivo para mejorar la calidad de software. Al detectar y eliminar un gran porcentaje de errores, el proceso de revisión reduce substancialmente los problemas en los siguientes pasos de desarrollo y mantenimiento.

REVISIONES TÉCNICAS FORMALES

Los objetivos de las revisiones técnicas formales son:

1. Descubrir errores en la función, la lógica o implementación de cualquier representación del software.
2. Verificar que el software en revisión alcanza sus requisitos.
3. Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.
4. Conseguir un desarrollo uniforme.
5. Hacer proyectos más manejables.

La RTF sirve para promover la seguridad y continuidad, incluye recorridos, inspecciones y otras tareas de revisión técnica de software. Cada RTF se lleva a cabo mediante una reunión que tendrá éxito si es bien planeada, controlada y atendida.

Registro e informe de la revisión

Durante la RTF uno de los integrantes registra dinámicamente todas las mejoras, observaciones y sugerencias técnicas que vayan surgiendo. Al final se revisan cada una de ellas y se genera una lista de sucesos de revisión. Además se genera un informe sumario o minuta en el que se especifica que se revisó, quien lo revisó, qué se descubrió y sus conclusiones. La lista de sucesos sirve para dos propósitos:

1. Identificar áreas problemáticas dentro del producto.
2. Servir como lista de puntos de acción que guíen al productor a hacer las correcciones correspondientes.

Directrices de una revisión.

Se deben establecer de antemano las directrices que sirvan para conducir las revisiones técnicas formales, distribuyéndolas entre los revisores, para ser consensadas y seguidas. A continuación se muestra un conjunto de directrices para las revisiones técnicas formales:

1. Revisar el producto.
2. Fijar una agenda y mantenerla.
3. Limitar el debate.
4. Enunciar áreas de problemas, pero no intentar resolver cualquier problema que se ponga en manifiesto.
5. Tomar notas escritas.
6. Limitar el número de participantes e insistir en la preparación anticipada.
7. Desarrollar una lista de comprobaciones de cada producto a ser revisado.
8. Disponer recursos y una agenda para las RTF.
9. Llevar a cabo un entrenamiento de todos los revisores.
10. Repasar las revisiones anteriores.

Listas de comprobación

Las listas de comprobación pretenden proporcionar un punto de partida para cada revisión. Asigna la función y el rendimiento de los elementos del sistema. Los grupos de ingeniería de software y hardware se centran en las asignaciones de actividades relativas a su especialidad. El aseguramiento de calidad evalúa los requisitos de validación a nivel del sistema y el servicio de campo examina los requisitos para llevar a cabo diagnósticos.

Planificación del Proyecto de Software.

Hace las estimaciones de recursos, costos y tiempos para el desarrollo, llevadas a cabo en la planificación del proyecto, se basan en la asignación de las cargas de trabajo dentro de las

actividades de desarrollo del sistema. La revisión del plan intenta establecer el grado de riesgo.

Análisis de Requisitos

Las revisiones de análisis se centran en el seguimiento, consistencia y corrección de los requisitos del sistema.

Diseño de Software

Se centra en el diseño de datos, procedimientos y arquitectura del sistema. En general, se pueden realizar dos tipos de revisiones. La revisión del diseño preliminar (diseño conceptual) para confirmar la traducción de los requisitos de diseño, centrándose en la arquitectura del software. La segunda revisión se centra en la corrección procedimental de los algoritmos.

Codificación

Aunque la codificación es resultado del diseño procedimental, se pueden introducir errores al traducir el diseño a un lenguaje de programación.

Prueba de Software

Es una actividad de SQA por derecho propio, sin embargo, se puede mejorar la efectividad y complejidad de una prueba validando cualquier plan o procedimiento de prueba preestablecido.

Mantenimiento

Las listas de comprobación que hemos señalado para la revisión del desarrollo del software son válidas para la fase del mantenimiento del software.

6.3 PRUEBAS DE SOFTWARE

La prueba de software es un elemento crítico para el aseguramiento de calidad del software, y representa una revisión final de las especificaciones, diseño y codificación. Incluso debe ser un elemento más del sistema, y su importancia en el costo final está asociado a la creación de pruebas minuciosas y planificadas. El diseño de casos de prueba se centra en un conjunto de técnicas para la creación de matrices que satisfagan los objetivos globales de la prueba.

Objetivos de la prueba

1. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. La prueba tiene éxito si descubre un error que no fue detectado anteriormente. El objetivo es diseñar pruebas que sistemáticamente encuentren diferentes clases de error en menor tiempo y esfuerzo.

La ventaja secundaria de la prueba es mostrar hasta que punto las funciones trabajan de acuerdo con las especificaciones, y como estas alcanzan los requisitos y rendimientos esperados. Además, los datos que se van recogiendo a lo largo de la prueba proporcionan un índice de fiabilidad de software.

El flujo de información en la prueba

Se proporcionan dos clases de entradas al proceso de prueba: La configuración del software que incluye la especificación de requisitos, diseño y código fuente; y una configuración adecuada que incluya un plan y procedimiento de la misma.

A medida que se recopilan y evalúan resultados, empieza a vislumbrarse una medida cuantitativa de calidad y fiabilidad del software. Si se encuentran con regularidad serios errores que requieran modificación al diseño entonces son necesarias pruebas posteriores pues la calidad del software no es buena. Por otro lado, si el funcionamiento es correcto (en apariencia) y los errores que se encuentran son fáciles de corregir, se puede sacar una de dos conclusiones: La calidad y fiabilidad del software son aceptables o las pruebas son inadecuadas para descubrir errores serios.

Diseños de casos de prueba

El diseño de pruebas puede requerir tanto esfuerzo como el propio diseño del producto. Recordando el objetivo de una prueba, se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de tiempo y esfuerzo.

Cualquier producto de ingeniería puede probarse en dos formas: Conociendo la función para la que fue diseñado el producto, se pueden llevar a cabo pruebas dirigidas a detectar que cada función cumpla con los requisitos para la que fue diseñada; conociendo el funcionamiento del producto se pueden hacer pruebas que aseguren que todas las piezas encajan. El primer enfoque de prueba se denomina *"prueba de la caja negra"* y la segunda *"prueba de la caja blanca"*. En seguida describiremos cada uno de los objetivos de las mismas.

La prueba de la caja blanca

Es un método de diseño de casos que usa la estructura de control del diseño procedimental para definir los casos de prueba. Con ellos se garantiza que se ejerciten todos los caminos independientes de cada módulo; también se verifican las decisiones lógicas (verdadera y falsa); se podrán ejecutar los ciclos en sus límites operacionales, además se ejercitarán las estructuras internas para asegurar su validez. Las ventajas de las pruebas de caja blanca son las siguientes:

1. Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Los errores se reproducen cuando implementamos funciones, condiciones o controles fuera de lo normal.
2. A menudo creemos que un camino lógico tiene pocas posibilidades de ejecución cuando de hecho se puede ejecutar en forma regular. El flujo lógico de un programa no es intuitivo, lo que significa que nuestras suposiciones sobre el flujo de control y los datos nos pueden llevar a errores de diseño que sólo se descubren cuando comienza la prueba.
3. Los errores tipográficos son aleatorios. Cuando se traduce un programa a código fuente en cualquier lenguaje, es muy posible que se den errores de escritura.

Para realizar esta prueba debemos realizar las siguientes prácticas:

1. Recorrer por lo menos una vez todos los caminos independientes de cada módulo.
2. Forzar al programa en todas sus vertientes las decisiones lógicas de falso y verdadero.
3. Verificar todos los ciclos iterativos para evitar un ciclo anidado o infinito.
4. Revisar las estructuras internas de los datos para asegurar su validez.

Como ejemplo tomaremos del módulo de llamadas la función de apertura y seguimiento de reportes, para la cual ejercitaremos todos los caminos.

1. Primero introducimos una clave de afiliación y de región que no exista, entonces la función validará que efectivamente no existe, y solicitará al operador los datos del comercio, posteriormente el usuario podrá llenar la solicitud de reporte.
2. Después introducimos el mismo número de afiliación. Esta función debe detectar también si encuentran reportes abiertos, y registrados en la misma fecha, entonces el programa nos deberá mostrar los datos de los mismos, en este momento el operador podrá confirmar si abrirá otro reporte. Si el usuario determina que es una causa de reporte nuevo se captura la solicitud de la llamada, en caso contrario termina el programa.

Como observamos, éstas pruebas centra su atención en la estructura de control. A continuación mostramos la figura que representa la función de apertura y seguimiento a reportes.

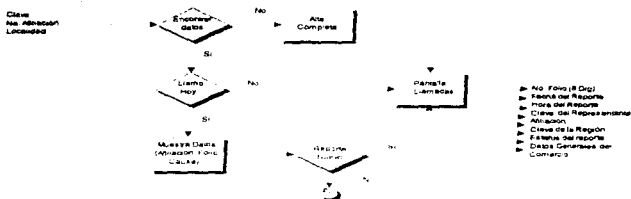


Figura 6.3 Prueba de la caja blanca

Prueba del camino básico

Es una técnica de prueba de la caja blanca. El método permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño y usarla para definir un conjunto básico de caminos de ejecución. Tal como lo vimos en el caso anterior.

Prueba de la caja negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Las pruebas pretenden demostrar que las funciones del software son operativas, que la entrada se acepta adecuadamente (validaciones) y se produce una salida correcta, así como la integridad de la información externa. Una prueba de caja negra examina aspectos del modelo fundamental sin tener en cuenta la estructura lógica interna del software.

Se centran en los requisitos funcionales del software. éstas pruebas no son alternativas a las técnicas de prueba de la caja blanca más bien se trata de un enfoque complementario que intenta descubrir errores que no fueron detectados durante las pruebas de la caja blanca. La prueba de caja negra intenta encontrar errores de las siguientes características:

1. Funciones incorrectas o simples.
2. Errores de interfaz.
3. Errores en estructuras de datos o acceso a bases externas.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

Para realizar esta prueba debemos realizar las siguientes prácticas:

1. Introducir datos de entrada. Se validará que se introduzcan todos los datos del comercio y del reporte con la finalidad de que estos no estén incompletos.
2. Llevar a cabo una prueba de funcionalidad de la pantalla para asegurar que la captura de información sea fácil y rápida.
3. Corroborar que la pantalla se despliegue correctamente aún en diferentes tipos de monitor, de lo contrario se debe tener en cuenta al momento de la definición de características de la Interfase.
4. Verificar que los métodos de búsqueda de información son adecuados y rápidos para las necesidades del usuario.

Con esta prueba podemos detectar , si el sistema es sensible a ciertos valores, además de los volúmenes y niveles de datos que soporta el sistema.

Clase
Tip. Afiliación
Liderado

► No. Foto (B. DGI)
► Fecha del Recurso
► Hora del Recurso
► Clase de Representante
► Afiliación
► Clase de la Región
► Estado del reporte
► Datos Generales del Comercio

Figura 6.4 Prueba de la caja negra

Análisis de valores límite

Por razones no del todo claras, los errores tienden a darse en los límites del campo de entrada. Por eso, se ha desarrollado este análisis como técnica de prueba. En el que en lugar de seleccionar cualquier elemento de una clase, se seleccionan los casos en los bordes de la misma.

Técnicas de gráficas de causa - efecto

En una gran cantidad de situaciones, el intento de traducir una política o procedimiento expresado en un lenguaje natural a un algoritmo basado en software, conduce a la frustración y al error. Las gráficas de causa y efecto son una técnica de diseño que representa las condiciones lógicas y sus correspondientes acciones. La técnica sigue cuatro pasos:

1. Se listan para un módulo las causas (condiciones de entrada) y los efectos (acciones), asignando un identificador a cada uno de ellos.
2. Se desarrolla una gráfica de causa efecto.
3. Se convierte la gráfica en una tabla de decisión.
4. Se convierten las reglas de la tabla de decisión a casos de prueba.

6.4 MANTENIMIENTO DE SOFTWARE

El mantenimiento de software existente puede llevarse hasta el setenta por ciento de todo el esfuerzo gastado por una organización de desarrollo. El porcentaje se incrementa a medida que se produce más software. La naturaleza omnipresente del cambio mediatiza todo el trabajo en software. En cambio es inevitable en la construcción de sistemas basados en computadora, por ello, debemos desarrollar mecanismos de evaluación, control e implementación de modificaciones.

Definición de mantenimiento de software

El mantenimiento de software es mucho más que una "corrección de errores". Este tiene las siguientes formas de manifestarse.

1. **Solicitudes de corrección** La primera actividad de mantenimiento de software es debida a que no es razonable que la primera prueba de software haya descubierto todos los errores latentes de un gran sistema. Este proceso incluye el diagnóstico y la corrección de uno o más errores.
2. **Reportes de problemas de performance** La segunda actividad que contribuye a la definición de mantenimiento se produce por el rápido cambio inherente a cualquier aspecto de la informática.
3. **Solicitudes de maquila** Cuando un paquete de software tiene éxito se reciben recomendaciones de los usuarios, para satisfacer estas peticiones se llevan a cabo acciones de mantenimiento. Esta actividad contabiliza la mayor cantidad de esfuerzo empleado en el mantenimiento de software. Este tipo de mantenimiento también se da cuando se cambia el software para proporcionar una base mejor para futuras mejoras. Otra actividad del mantenimiento perfecto se caracteriza por las técnicas de ingeniería a la inversa y reingeniería.

Características del mantenimiento

Considérese el asunto desde tres puntos de vista diferentes:

1. Las actividades requeridas para cubrir la fase de mantenimiento y el impacto de un enfoque de ingeniería de software sobre la eficacia de tales actividades.
2. Los costos asociados a la fase de mantenimiento.
3. Los problemas que se presentan cuando se lleva a cabo el mantenimiento.

El mantenimiento estructurado

Si sólo se cuenta con el código fuente como elemento de configuración, la actividad de mantenimiento comienza con una dolorosa evaluación del código, a menudo complicada por la pobre documentación interna. Las delicadas características, tales como la estructura del programa, las estructuras globales, las interfaces del sistema, el rendimiento y/o las limitaciones del diseño, son difíciles de descubrir y frecuentemente, mal interpretadas. Lo que hacemos es un mantenimiento no estructurado y pagamos el precio asociado a todo software que no haya sido desarrollado mediante una metodología bien definida. Si existe una completa configuración del software, la tarea de mantenimiento ideal contempla los siguientes pasos:

1. Evaluar la documentación del diseño técnico y funcional.
2. Determinar las características estructurales, de rendimiento e interfaz de software.
3. Estudiar las modificaciones o correcciones y se traza un plan de trabajo.
4. Se modifica diseño y se revisa.
5. Se desarrolla un nuevo código fuente.
6. Se efectuar pruebas de regresión mediante la información contenida.

La secuencia de sucesos anterior constituye el mantenimiento estructurado y forma parte de la metodología de software descrita en el capítulo uno, la cual como sabemos conforma el ciclo de vida del software.

Costos del mantenimiento

Los costos de mantenimiento se han incrementado en los últimos años. Un costo intangible del mantenimiento se encuentra en una oportunidad de desarrollo que se ha de posponer o se pierde, debido a que los recursos disponibles deben estar dedicados a las tareas de mantenimiento. Otros intangibles son:

- Insatisfacción del cliente cuando una petición de reparación o modificación no se puede atender en un tiempo razonable.
- Disminución global del software debido a errores latentes que introducen los cambios de software mantenido.
- Trastornos en otros esfuerzos de desarrollo al tener que poner a la plantilla de desarrollo en tareas de mantenimiento.

El costo final del mantenimiento es una drástica reducción de la productividad cuando se inicia el mantenimiento de viejos programas.

Problemas

La mayoría de los problemas asociados con el mantenimiento de software se debe a las deficiencias de la forma en que el software ha sido definido y desarrollado. La falta de control, disciplina en las actividades de desarrollo casi siempre se traduce en "problemas" para el mantenimiento de software. Entre los problemas clásicos asociados al mantenimiento de software se encuentran los siguientes:

- A menudo es difícil o imposible seguir la evolución del software a través de varias versiones. Los cambios no están debidamente documentados.
- Es difícil o imposible seguir el proceso de construcción del software.
- Es difícil la comprensión de un programa "ajeno" a medida que existen menos elementos de configuración del software, mayor es la dificultad.
- El personaje "ajeno". A menudo no se encuentra la persona que realizó el programa para que pueda explicar lo que se hizo.
- No existe una documentación apropiada o está mal preparada.
- La mayoría del software no fue creado previendo un cambio.
- El mantenimiento del software no se ve como un trabajo atractivo.

Facilidad de mantenimiento

La facilidad de mantenimiento se puede definir cualitativamente como la facilidad de comprender, corregir, adaptar y/o mejorar el software.

La facilidad de mantenimiento depende de muchos factores. Una pobre configuración del software puede tener un impacto negativo incluso cuando se hayan seguido los pasos técnicos mencionados anteriormente. Además de los factores que se pueden asociar a una metodología de desarrollo existen factores relacionados con el entorno de desarrollo, tales como:

- Disponibilidad de una plantilla de software calificada
- Estructura del sistema comprensible
- Facilidad del manejo del sistema
- Usos de lenguajes de programación estandarizados
- Usos de sistemas operativos estándar
- Estructura de la documentación estándar
- Disponibilidad de los casos de prueba
- Facilidades de depuración incorporadas
- Disponibilidad del equipo apropiado para llevar a cabo el mantenimiento

Medidas cuantitativas

La facilidad de mantenimiento como la calidad o la fiabilidad son difíciles de cuantificar. Sin embargo, podemos evaluar la facilidad de mantenimiento indirectamente, considerando que la actividad de mantenimiento que se puede medir. Algunas de las métricas de mantenimiento relacionadas con el esfuerzo del mismo son:

1. Tiempo de reconocimiento del problema.
2. Tiempo de retraso administrativo.
3. Tiempo de recolección de herramientas de mantenimiento.
4. Tiempo de análisis del problema.
5. Tiempo de especificación de cambios.
6. Tiempo activo de corrección.
7. Tiempo de prueba local.
8. Tiempo de prueba global.
9. Tiempo de revisión del mantenimiento.
10. Tiempo total de recuperación.

Cada una de las métricas anteriores puede ser registrada sin mucha dificultad. Tales datos proporcionan un indicador de la eficacia de las nuevas técnicas y herramientas.

Tareas de mantenimiento

Inicialmente se debe conformar una organización de mantenimiento; se deben describir los procedimientos de evaluación y de información, y se debe definir una secuencia estandarizada de sucesos para cada petición de mantenimiento. Además se debe establecer un sistema de registro de información de las actividades y definir los criterios de revisión y evaluación.

Organización de mantenimiento

Aunque no es necesario establecer una organización de mantenimiento formal, una delegación informal de responsabilidades es absolutamente esencial, incluso para pequeños grupos de desarrollo.

Las peticiones se dirigen a un controlador del mantenimiento que remite cada petición a un supervisor del sistema para que la evalúe. El supervisor del sistema es un miembro de la plantilla técnica al que se asigna la responsabilidad de familiarizarse con un pequeño subconjunto de programas producidos. Una vez hecha la evaluación, una autoridad de control de cambios debe determinar las acciones a realizar. La organización sugerida reduce la confusión, y mejora el flujo de las actividades de mantenimiento.

El implementar un mecanismo de control de cambios para aprobar cambios específicos, la organización puede evitar realizar cambios que beneficien a un usuario pero que tenga un impacto negativo sobre el resto de los mismos.

Informes

Todas las peticiones de mantenimiento deben presentarse en forma estándar. Normalmente el equipo de desarrollo genera un formulario de petición de mantenimiento (FPM), orden de servicio (ODS) o Informe de problemas de software, que ha de ser llenado por el usuario que desea la actividad de mantenimiento. Si se cuenta con un error, se debe incluir una descripción de las circunstancias que llevaron al error y sus evidencias (listados, datos de entrada, números de error, etcétera). Para mantenimientos perfectivos o adaptativos, se ha de emitir una breve especificación de cambios.

El FPM es un documento elaborado generalmente por el usuario y se usa como base de planificación de las tareas de mantenimiento. Internamente la organización desarrolla un informe de cambios del software (ICS), indicando:

1. Magnitud del esfuerzo requerido para atender el FPM
2. Naturaleza de las modificaciones
3. Prioridad de la petición
4. Otros datos de la modificación

EL ICS se envía a la autoridad de control de cambios antes de iniciar cualquier planificación.

6.5 FLUJO DE SUCESOS PARA REALIZAR UN MANTENIMIENTO

El flujo de sucesos se divide en cuatro fases que son:

1. Fase de recepción de la petición.
2. Fase de organización de la petición.
3. Fase de ejecución del mantenimiento.
4. Fase de las mejoras al sistema

FASE: RECEPCIÓN

La secuencia de sucesos es producto de una petición de mantenimiento, el primer requisito es determinar el tipo de mantenimiento que se va a llevar a cabo. Para lo cual se debe enmarcar el problema en cualquiera de los siguientes tipos.

1. Mantenimiento Correctivo debido a un error (serio o no tan serio) que escapó de la fase de pruebas.
2. Mejoras en cuestión de rendimiento de la red o del sistema en alguna de sus funciones
3. Una solicitud de maquila.

FASE: ORGANIZACIÓN

Cuando el mantenimiento solicitado es debido a que se detectó un error serio se tendrán que abandonar momentáneamente los controles normales del mantenimiento. El código habrá de ser modificado inmediatamente, sin la correspondiente evaluación de los posibles efectos secundarios y sin una adecuada actualización de la documentación.

Este modo de actuación inmediata para el mantenimiento correctivo será utilizado para situaciones de "crisis" exclusivamente; una vez que se ha resuelto la crisis se deben concluir los controles, y la evaluación para asegurar que los recientes arreglos no generarán problemas que incluso pueden ser más serios del que se presentó. Para el caso de que el error presentado no sea muy serio procederemos a evaluarlo, clasificarlo y colocarlo en la cola de prioridades.

FASE: MEJORA

Cuando el mantenimiento solicitado es de tipo correctivo o de perfeccionamiento las adaptaciones se evalúan y se clasifican asignando prioridades antes de situarlas en una cola de acciones de mantenimiento. cabe hacer mención que no todas las peticiones de mejoras se llevan a cabo. Algunos de los factores por los que se puede denegar una petición son los siguientes: por estrategia del negocio, los recursos disponibles, la línea de productos de

software actuales y futuros, etcétera. Una vez que tenemos establecida la prioridad de cada petición se sitúan en la cola de mantenimiento y por último se planifica el trabajo.

FASE: EJECUCIÓN

Independientemente del tipo de mantenimiento se siguen las mismas tareas técnicas. Estas tareas incluyen la modificación del diseño del software, la revisión, las modificaciones oportunas en el código, las pruebas de unidad, integración, validación y la revisión. De hecho, el mantenimiento de software es realmente una aplicación recursiva de la ingeniería de software. El último suceso del flujo de mantenimiento es una revisión que valide todos los elementos de la configuración del software que asegure la satisfacción del FPM.

Una vez terminada la tarea de mantenimiento, a menudo se lleva a cabo un repaso de la situación con objeto de tener una gran influencia en los futuros esfuerzos de mantenimiento y proporcionar un mecanismo de retroalimentación que es importante para una gestión efectiva de la organización del software. A continuación mostramos un diagrama de flujo el cual muestra los sucesos del mantenimiento.

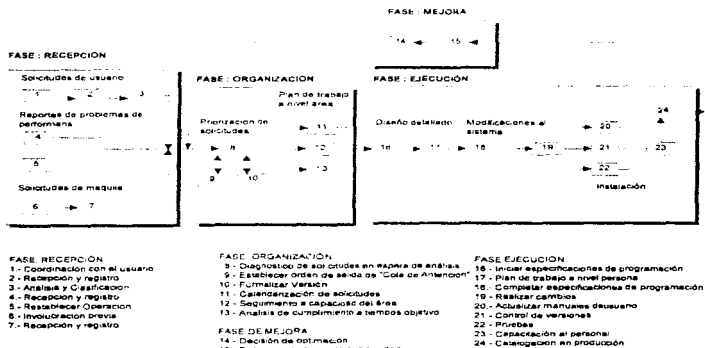


Figura 5 Flujo de Sucesos del Mantenimiento

Registro de información

El primer paso a considerar en este tópico consiste en comprender que datos merece la pena registrar, algunos de ellos son:

1. Identificar el programa
2. Número de sentencias fuente
3. Número de instrucciones en código máquina
4. Lenguaje de programación utilizado
5. Fecha de instalación del programa
6. Número de ejecuciones del programa desde la instalación
7. Número de fallos asociados con el punto 6
8. Nivel e identificación de cambios sobre el programa
9. Número de sentencias fuente añadidas en los cambios al programa
10. Número de sentencias eliminadas en los cambios al programa
11. Número de personas - hora por cambio
12. Fecha de cambio del programas
13. Identificación del ingeniero de software que efectuó los cambios
14. Identificación del FPM
15. Tipo de mantenimiento
16. Fecha de comienzo y final del mantenimiento
17. Número de personas hora acumulado del mantenimiento
18. Beneficios netos asociados con el mantenimiento realizado

Los datos anteriores deben registrarse después de cada esfuerzo de mantenimiento.

Evaluación

A menudo la evaluación de las actividades de mantenimiento de software se complica por la falta de datos registrados. Si este se lleva a cabo se pueden llevar a cabo varias medidas de rendimiento del mantenimiento. Una lista de medidas potenciales son:

1. Número de fallos de procesamiento por ejecución del programa
2. Total de horas hombre por cada categoría de mantenimiento
3. Número medio de cambios por sentencia fuente añadida o eliminada por el mantenimiento
4. Media de personas/hora por lenguaje
5. Tiempo promedio por FPM
6. Porcentaje de peticiones de mantenimiento por tipos

Estas medidas pueden proporcionar una base cuantitativa sobre la cuales se puede tomar decisiones sobre la técnica de desarrollo, la selección de un lenguaje, las previsiones del esfuerzo de mantenimiento, de disposición de recursos y muchos otros aspectos.

6.6 CONTROL DE VERSIONES.

El control de versiones combina procedimientos y herramientas para gestionar las versiones de los objetos creadas durante el proceso de ingeniería de software. El control de versiones también es llamado "Gestión de Configuración". éstas herramientas y procedimientos permiten a un usuario especificar configuraciones alternativas del sistema software mediante la selección de las versiones adecuadas. Esto se puede gestionar asociando atributos a cada versión del software y permitiendo luego especificar una configuración describiendo el conjunto de atributos deseado. Una forma de representar las diferentes versiones de un sistema es mediante una gráfica de evolución, el cual se muestra a continuación:

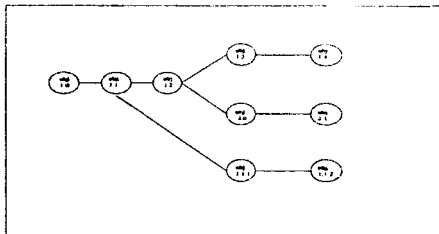


Figura 6.6 Gráfica de evolución

En esta figura mostramos el objeto de configuración 1.0 pasada la revisión se convierte en el objeto 1.1. Algunas correcciones y cambios mínimos producen las versiones 1.1.1 y 1.1.2, que van seguidas de una actualización importante, resultando el objeto 1.2. La evolución del objeto 1.0 sigue hacia el 1.3 y 1.4, pero al mismo tiempo, una gran modificación del objeto produce un nuevo camino de evolución, la versión 2.0. Por lo que una nueva versión se define cuando se realizan cambios significativos en uno o más objetos.

Las versiones nuevas son modificaciones o cambios realizados a algún objeto de aquí surge la importancia de tener un buen control de los cambios a realizar en el software.

6.7 CONTROL DE CAMBIOS.

El control de cambios combina los procedimientos humanos y las herramientas automáticas para proporcionar un mecanismo que regulen los cambios ya que al tenerlos regulados podemos evitar el caos en el sistema. A continuación mostramos un esquema del proceso de control de cambios.

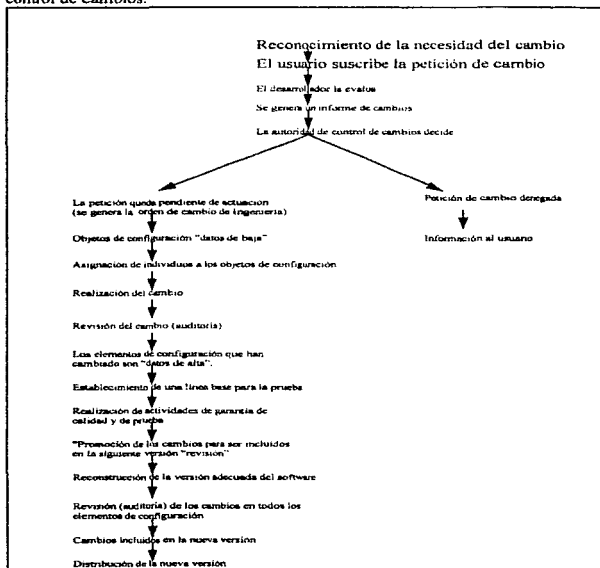


Figura 6.7 Esquema de control de cambios

Al realizar una petición de cambio, se evalúa con la finalidad de calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y los costos estimados. Estos resultados son presentados como *informe de cambios a la autoridad que los regula* - persona o grupo de personas que toman la decisión final del estado de prioridad del cambio. Para cada caso aprobado se genera una orden de cambio de ingeniería en la cual se describe la modificación a realizar, las restricciones deben respetar los criterios de revisión y auditoría. Cuando se efectúa un cambio, los pasos a seguir son los siguientes:

1. El objeto a cambiar es dado de baja en la base de datos o directorio de programas del proyecto.
2. Se realiza la modificación y se aplican las actividades de aseguramiento de calidad de software.
3. El objeto es dado de alta en el directorio de programas.

Los procesos de "alta y baja implementan dos elementos importantes del control de cambios que es el control de acceso y el control de sincronización". El control de acceso gobierna los derechos de los ingenieros de software a acceder y modificar objetos de configuración concretos. El control de sincronización ayuda a asegurar que los cambios en paralelo realizados por diferentes equipos no se sobrescriban.

CONCLUSIONES

La carrera de ingeniero en computación es una profesión que requiere de capacidades y cualidades que son desarrolladas mediante la experiencia, y que día a día se enriquecen con el estudio constante. Todos estos elementos son cubiertos básicamente durante los estudios profesionales, marcando las pautas y lineamientos que serán mejorados al enfrentar nuevos retos y oportunidades a lo largo de la actividad profesional.

Nuestra tesis lejos de querer cubrir todos los aspectos que abarca el perfil del ingeniero en computación, aborda una serie de características que soportan la experiencia, la invaluable asesoría de nuestros maestros, y la documentación de diferentes autores.

Para desarrollar un sistema de información son necesarios varios elementos tales como: técnicas de programación, diseño de bases de datos, conocimiento del hardware y software en el mercado, sistemas operativos, lenguajes de programación, redes y sus topologías, etcétera. Todos estos factores nos permiten enfocar y delimitar el problema.

El siguiente punto por considerar dentro de la tarea de consultor en sistemas lo constituyen las técnicas para asegurar la calidad del producto o servicio por desarrollar. Para ello se llevan a cabo entrevistas, con el objeto de tener un conocimiento profundo del negocio y sus reglas. De ahí que el trabajo del consultor deberá ser no solo automatizarlo, sino que debe incorporar soluciones que hagan más eficiente al negocio con base en el conocimiento de sistemas de las mismas características a escala mundial para tratar de incorporar estas técnicas y procesos en el negocio. Lo anterior, aunado al manejo de una metodología harán que el proyecto sea más eficiente al crearse los mecanismos de control y retroalimentar con el cliente o usuario del producto.

La técnica y el método se ven complementadas a través de las técnicas de administración y control del proyecto. Con ellas nuestras ideas y proyectos podrán ser incorporadas y adquiridas siempre y cuando estén bien sustentadas bajo un planteamiento estratégico y económico claro, el cual nos permita dar seguimiento al proyecto y den certeza al usuario de que se está obteniendo un precio justo, y un producto con la calidad esperada.

Resumiendo nuestra tesis basa el éxito de una consultoría en sistemas en los factores antes mencionados. En la vida real, la capacidad de llevar procesos de integración vertical, y la creación de equipos de trabajo interdisciplinario son de suma importancia. En nuestro caso el proyecto esta regido por los siguientes puntos para el buen logro de un proyecto de ingeniería de sistemas.

1. **Metodología:** contar con ella nos permite definir, delimitar y entender las necesidades del cliente. Al mismo tiempo nos proporciona un conjunto de pasos ordenados que nos permiten garantizar la calidad de los productos y servicios solicitados.
2. **Diseño Técnico y Funcional:** Técnicas para el diseño y selección del modelo de datos, de hardware y software necesarios para implementar la solución del problema en cuestión, con el objetivo final de obtener un modelo preliminar del sistema lo suficientemente robusto para reducir el esfuerzo en el momento de llevarlo a la práctica.
3. **Selección del Hardware y software:** Esto se podrá dar en base a detectar las necesidades de infraestructura y de expansión a corto y mediano plazo para proteger la inversión del cliente, y hacer que sus sistemas puedan adecuarse a cambios en la tecnología, procesos, etcétera.
4. **Plan de Trabajo:** Este debe ser muy claro y debe estar sustentado con técnicas que nos permitan tener un máximo control del proyecto, nos debe permitir conocer nuestros ingresos y egresos, detectar desviaciones y avances con el fin de tomar medidas correctivas en corto y mediano plazo.

Durante el periodo posterior a la instalación del sistema se detectaron fallas de funcionalidad en algunas de sus pantallas, mismas que se corrigieron para que el cliente tuviera un mejor desplazamiento dentro de ellas. Además se corroboró que la consistencia en los datos fuera adecuada y de acuerdo a lo esperado. Después de una semana que este estuvo en producción se estabilizó su operación y actualmente opera sin ningún inconveniente.

Las áreas de oportunidad que se detectan después de un tiempo que el sistema estuvo en producción son:

- Sistema que permita rutear los reportes en base al código postal, dirección, zona, etc.
- Sistema de inventarios para conocer que equipos tiene instalado el usuario en sitio.
- Crear un módulo de soporte al usuario que le permita diagnosticar y resolver problemas técnicos de primer nivel. Esto se conseguiría si se hacen árboles de toma de decisiones para indicarle al usuario como resolver un problema en base a las características del mismo, y las preguntas y respuestas que se generen durante la atención del reporte. El sistema debe ser inteligente, y tendrá la capacidad de reconocer nuevos problemas, incluso dar soporte a los usuarios novatos del sistema.

Son éstos, entre otros, algunas de las posibles mejoras que puede tener el sistema implementado. Este podría ser instalado en otras ciudades y enviar información al centro de operaciones.

El sistema en la actualidad opera con 20 operadores que atienden reportes de primer nivel, un supervisor, el área de logística puede manejar la información ágilmente y reportar a la gerencia media y dirección para la toma de decisiones en corto plazo. Se pretende expandirse en el número de operadores. Por otra parte se cuenta ahora con más proveedores o resolvers de segundo nivel y se puede medir su eficiencia y capacidad (tiempos de respuesta y calidad en el servicio). En cuanto a resultados del sistema implementado se resumen los siguientes beneficios:

- El registro de información y el seguimiento de los reportes se puede hacer en forma más ágil
- Los reportes de primero y segundo nivel pueden ser medidos y clasificados. Se incremento la cantidad de operadoras, cubriendo de esta forma una mayor demanda de clientes
- La infraestructura de hardware y software implementada permiten a la empresa crecer en su número de usuarios a bajo costo
- La empresa puede obtener reportes estratégicos en menos tiempo y ser mas eficientes para la toma de decisiones

Con los cuales se cumplen los objetivos planteados en el inicio del proyecto. El sistema se encuentra operando con éxito en la actualidad.

BIBLIOGRAFÍA

CARLO BATINI/ ESTEFANO CERI / SHAMKANT B. NAVATHE

DISEÑO CONCEPTUAL DE BASES DE DATOS

EDITORIAL ADDISON-WESLEY/DÍAZ DE SANTOS, 1994. P.P.1-193,221-307,309-429

ROGER S. PRESSMAN

INGENIERÍA DE SOFTWARE UN ENFOQUE PRÁCTICO

TERCERA EDICIÓN 1994. P.P 575-741

EDITORIAL Mc GRAW-HILL

INVESTIGACIÓN DE OPERACIONES EN LA CIENCIA ADMINISTRATIVA

G.D.EPPEN/F. J.GOULD

SEGUNDA EDICIÓN 1987. P.P. 353-397

EDITORIAL PRENTICE HALL

ANDREWS S. TANENBAUM

REDES DE ORDENADORES

SEGUNDA EDICIÓN 1991, P.P. 1-22, 166 - 192

EDITORIAL PRENTICE HALL

TECNOLOGÍA DE ENRUTADORES E INTERCONEXIÓN DE REDES

MATERIAL DE RED UNO

SEMINARIO DE ACTUALIZACIÓN, ENERO 1996

REDES DE CÓMPUTO

ING. DANIEL CALLI

SEMINARIO DE ACTUALIZACIÓN, OCTUBRE 1995

CONCEPTOS GENERALES SOBRE REDES LOCALES

DR. MARCELO MEJÍA OLVERA

REVISTA SOLUCIONES AVANZADAS, MAYO 1993, P.P. 5-18

REDES LOCALES TIPO ETHERNET

M.C. JOSÉ A. INCERA DIEGUES

REVISTA SOLUCIONES AVANZADAS, AGOSTO 1993, P.P. 22-26

ETHERNET EN EVOLUCIÓN

DR. MARCELO MEJÍA OLVERA Y M.C. JOSÉ A. INCERA DIEGUES

REVISTA SOLUCIONES AVANZADAS, FEBRERO 1995, P.P. 48-55

REDES LOCALES TIPO ETHERNET

M.C. JOSÉ A. INCERA DIEGUES

REVISTA SOLUCIONES AVANZADAS, MAYO 1993, P.P. 22-26

CARACTERÍSTICAS BÁSICAS CONSIDERAR EN LA SELECCIÓN DE UNA RED LOCAL

ETHERNET, ARCNET O TOKEN RING

ING. ALEJANDRO RAMÍREZ

REVISTA RED P.P. 45-49

REDES DE AREA LOCAL TIPO TOKEN RING

MC RAUL GARCÍA RUIZ

REVISTA SOLUCIONES AVANZADAS, MAYO 1993, P.P. 16-22

ANEXOS

ANEXO 1
Diccionario de Datos

DICCIONARIO DE DATOS

TABLA: LLAMADAS

OBJETIVO: ALMACENAR LOS DATOS DE LAS LLAMADAS DE LOS COMERCIOS

1	FOLIO	N	7	FOLIO DEL REPORTE DE LA LLAMADA
2	AFLIACION	C	7	NO. AFLIACION DE L. COMERCIO
3	CEN. REG	C	2	CLAVE DEL CENTRO REGIONAL
4	COMERCIO	C	35	NOMBRE DEL COMERCIO
5	DIRECCION	C	35	DIRECCION DEL COMERCIO
6	TELEFONO	C	11	TELEFONO DEL COMERCIO
7	CP	C	5	CODIGO POSTAL
8	CVE. REP	C	2	CLAVE DEL REPRESENTANTE
9	CVE. EQP	C	2	CLAVE DEL EQUIPO
10	CVE. RES	C	2	CLAVE DEL RESOLVEDOR
11	SINTOMA	C	35	DESCRIPCION DEL PROBLEMA
12	FECHA. REP	D	8	FECHA DEL REPORTE
13	HORA. REP	C	5	HORA DEL REPORTE
14	REPORTADOX	C	35	PERSONA QUE REPORTA EL PROBLEMA
15	COD. CAUSA	C	2	CODIGO DE CAUSA
16	HORA. ASIG	C	5	HORA DE ASIGNACION
17	FECHA. ASIG	D	8	FECHA DE ASIGNACION
18	FECHA. SOL.	D	8	FECHA DE SOLUCION DEL PROBLEMA
19	HORA. SOL.	C	5	HORA DE SOLUCION
20	SEGUIMIEN	TEXTO	10	OBSERVACIONES DEL REPORTE
21	ESTATUS	C	2	ESTATUS DEL REPORTE

TABLA: PROSPECTOS

OBJETIVO: ALMACENAR LOS DATOS DE LOS COMERCIOS PROSPECTO

1	FOLIO	N	7	FOLIO DEL REPORTE DE L. PROSPECTO
3	CEN. REG	C	2	CLAVE DEL CENTRO REGIONAL
4	COMERCIO	C	35	NOMBRE DEL COMERCIO
5	DIRECCION	C	35	DIRECCION DEL COMERCIO
6	TELEFONO	C	11	TELEFONO DEL COMERCIO
7	CP	C	5	CODIGO POSTAL
8	CVE. REP	C	2	CLAVE DEL REPRESENTANTE
9	FECHA. REP	D	8	FECHA DEL REPORTE
10	HORA. REP	C	5	HORA DEL REPORTE
11	RESPONS	C	35	PERSONA QUE REPORTA EL PROBLEMA
12	CTA. CHQUES	C	13	CODIGO DE CAUSA

DICCIONARIO DE DATOS

TABLA: COMERCIOS

OBJETIVO: ALMACENAR LOS DATOS DE LAS LLAMADAS DE LOS COMERCIOS

1	AFILIACION	C	7	NO. AFILIACION DE L COMERCIO
2	CEN REG	C	2	CLAVE DEL CENTRO REGIONAL
3	COMERCIO	C	35	NOMBRE DEL COMERCIO
4	DIRECCION	C	35	DIRECCION DEL COMERCIO
5	TELEFONO	C	11	TELEFONO DEL COMERCIO
6	CP	C	5	CODIGO POSTAL
7	FAX	C	13	FAX DEL COMERCIO

TABLA: LOCALIDAD

OBJETIVO: CATALOGO DE LOCALIDADES

1	CEN REG	C	2	CLAVE DEL CENTRO REGIONAL
2	NOM LOCAL	C	30	NOMBRE DE LA LOCALIDAD

TABLA: CAUSA

OBJETIVO: CATALOGO DE EQUIPOS

1	CVE EQUIPO	C	2	CLAVE DEL EQUIPO
2	EQUIPO	C	35	DESCRIPCION DEL EQUIPO

TABLA: CAUSAS

OBJETIVO: CATALOGO DE CAUSAS

1	COD CAUSA	C	2	CLAVE DE LA CAUSA
2	CAUSA	C	35	DESCRIPCION DE LA CAUSA

TABLA: RESOLVEDORES

OBJETIVO: CATALOGO DE RESOLVEDORES DE SEGUNDO NIVEL

1	CVE RES	C	2	CLAVE DE L RESOLVEDOR
2	RESOLV	C	35	DESCRIPCION DEL RESOLVEDOR
3	SKY	C	15	SKY TEL DEL RESOLVEDOR

TABLA: REPRESENTEN

OBJETIVO: CATALOGO DE REPRESENTANTES

1	CVE REP	C	2	CLAVE DE L REPRESENTANTE
2	REPRESENTEN	C	35	DESCRIPCION DEL REPRESENTANTE
3	PASSWORD	C	5	PASSWORD DEL USUARIO

ANEXO 2
Directorio de Programas

DIRECTORIO DE PROGRAMAS

REGISTRO	Ecap_cal.prg
CONSULTAS	Ecto_cal.prg
PROSPECTOS	Abc_pros.prg
ENVÍO	Envio.prg
RECEPCIÓN	recepcion.prg
MÁS DE TRES LLAMADAS	N_com3.prg
DISTRIBUCIÓN POR CAUSA	N_caus.prg
OPORTUNIDAD DE ATENCIÓN	N_opres.prg
REPORTES ABIERTOS	N-abiert.prg
PROSPECTOS	N_pro.prg
CAUSAS	Causas.prg
LOCALIDADES	Local.prg
COMERCIOS	Comercio.prg
RESOLVEDORES	Resolv.prg
TIPOS DE EQUIPOS	Tipos.prg
USUARIOS	Represen.prg
BORRAR	Borrar.prg
RESPALDAR	Respaldar.prg
REINDEXAR	indices.prg
EXPORTAR	exportar.prg

ANEXO 3
Revisiones Técnicas
Formales
RTF

REVISIÓN TÉCNICA FORMAL

SISTEMA RESPONSABLE	<i>Electronic Window System</i> <i>Laura de Galvez</i>	FECHA HORA	<i>10-juni-97</i> <i>14:45</i>
----------------------------	---	-------------------	-----------------------------------

OBJETIVO DE LA REVISIÓN

MEJORAS POR REPORTE	()	PRUEBAS UNITARIAS	()
CAMBIOS EN EL DISEÑO	()	PRUEBAS MODULARES	<input checked="" type="checkbox"/>
NUEVA VERSIÓN	()	PRUEBAS INTEGRALES	()

SE ANEXA A ESTA REVISIÓN

SOLICITUD DE CAMBIOS DEL USUARIO	()
LISTAS DE COMPROBACIÓN	<input checked="" type="checkbox"/>
ORDEN DE SERVICIO	()
OTRAS _____	()

RESUMEN DE LOS RESULTADOS DE LA REVISIÓN

<i>Revisión formal de planos con modelos del sistema.</i>

AUTORIZACIONES Y FIRMAS DE LOS PARTICIPANTES EN LA REVISIÓN

<i>Laura de Galvez</i>	<i>[Signature]</i>
<i>[Signature]</i>	

Lista de comprobación

1. Dar Formato la fecha de ocurrencia del sistema
2. Centrar Browser (paginador) de reportes anteriores, ó hacerlo más largo para que se visualice mejor
3. Emparejar los Inputs de : Nombre, dirección y sintoma de la pantalla de registro de llamadas
4. En la pantalla de consultas Ad hoc al introducir una afiliación hay una falla, al parecer los tipos de datos no concuerdan
5. En la opción de envío de archivos: inicializar en blanco el centro regional y el resolvidor, si no hay datos a enviar debe indicarse, agregar un botón de browse o preview.
6. Mostrar el folio del prospecto

1. Agregar número de cuenta y folio al reporte de prospectos
2. Agregar la cantidad total de reportes abiertos en el reporte de Problemas abiertos

1. De todos las pantallas de catálogos quitar icono de lupa y búsqueda de conceptos, excepto en la pantalla de comercios
2. Validar el campo llave contra el valor cero ó blanco.
3. la pantalla de comercios cuando imprime no encuentra la variable Nombre
4. Cambiar todos los letreos que digan ciudad por localidad
5. El módulo de tipos de equipo cancela su operación en el menú después de usarlo , verificar porque hace esto

1. Los respaldos de una base de datos o de todas se debe direccionar a un directorio en particular

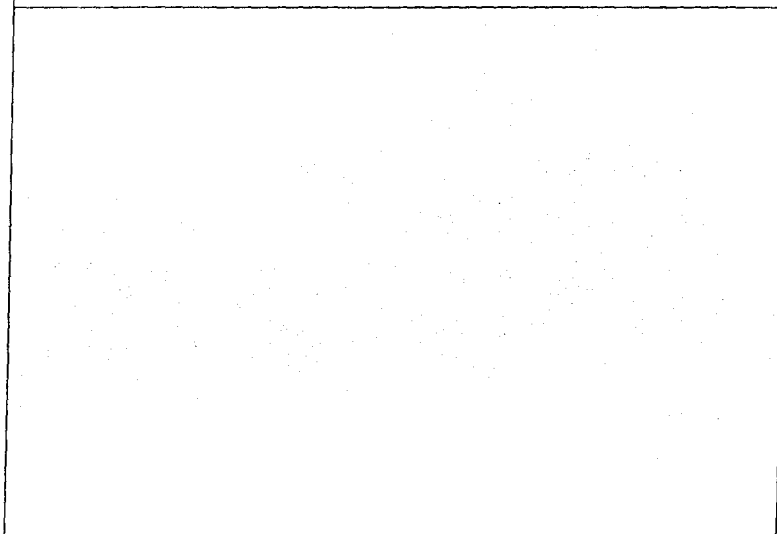
ANEXO 4
Diagramas de Gantt y PERT

Id	Nombre de tarea	Duración	Costo del Servicio	Comienzo	Fin	febrero			marzo		
						2	9	16	23	2	9
1	Ventana Electrónica de Servicio	32,5d	\$ 83.696,00	3/02/97	20/03/97						
2	Análisis preliminar (cap3)	2d	\$ 7.200,00	3/02/97	5/02/97						
3	Revisión de la situación actual	4h	\$ 1.200,00	3/02/97	3/02/97						Líder de proyecto
4	Objetivos y alcances del proyecto	4h	\$ 1.200,00	3/02/97	3/02/97						Líder de proyecto
5	Análisis y Propuestas (Diseño Conceptual)	2d	\$ 4.800,00	4/02/97	5/02/97						Líder de proyecto
6	Revisión con el usuario	0d	\$ 0,00	5/02/97	5/02/97						5/02
7	Desarrollo a la medida (Cap 3 y 4)	16,5d	\$ 48.000,00	6/02/97	28/02/97						
8	Levantamiento de Información Especifica (Diseño Detallado)	9d	\$ 19.200,00	6/02/97	16/02/97						
9	Diseño Funcional (Cap3)	4d	\$ 4.800,00	6/02/97	11/02/97						
10	Definición de Entradas y Salidas	2d	\$ 2.400,00	6/02/97	7/02/97						Analista Programador
11	Definición y Detecación de Políticas y Procedimientos (Regi)	1d	\$ 1.200,00	10/02/97	10/02/97						Analista Programador
12	Diagrama Estructural del Sistema	4h	\$ 600,00	11/02/97	11/02/97						Analista Programador
13	Diagramas de Flujo de Datos (DFD)	4h	\$ 600,00	11/02/97	11/02/97						Analista Programador
14	Definición de Hardware y Software (cap 4)	9d	\$ 9.600,00	6/02/97	16/02/97						
15	Análisis y Diseño de la Infraestructura de Voz	2d	\$ 2.400,00	6/02/97	7/02/97						Consultor Técnico
16	Análisis y Diseño de la Infraestructura de Datos	2d	\$ 2.400,00	10/02/97	11/02/97						Consultor Técnico
17	Análisis de Riesgo y Selección Final de HW y SW (revisión)	1d	\$ 4.800,00	16/02/97	16/02/97						Líder de proyecto;
18	Diseño Técnico (Cap 3)	4d	\$ 4.800,00	12/02/97	17/02/97						
19	Diseño de las Vozes y Pantallas	1d	\$ 1.200,00	12/02/97	12/02/97						Analista Programador
20	Modelo de Datos del Sistema (Diseño Conceptual de BD s)	1d	\$ 1.200,00	13/02/97	13/02/97						Analista Programador
21	Análisis de Datos y Funciones	2d	\$ 2.400,00	14/02/97	17/02/97						Analista Programa
22	Desarrollo, pruebas e instalación del Sistema	7,5d	\$ 28.800,00	19/02/97	26/02/97						
23	Desarrollo	7,5d	\$ 10.800,00	19/02/97	26/02/97						
24	Módulo de Llamadas	5,5d	\$ 4.400,00	19/02/97	26/02/97						

Instalación y Desarrollo del Sistema EWS

6/06/97

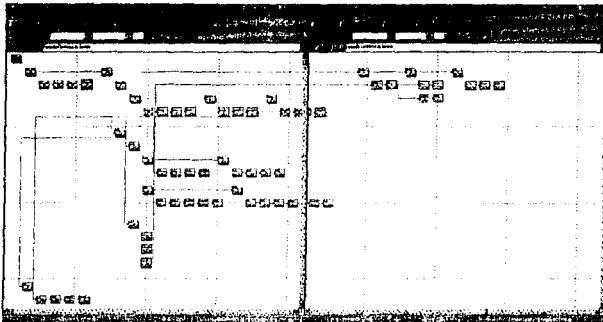
Id	Nombre de tarea	Duración	Costo del Servicio	Comienzo	Fin	febrero			marzo			
						2	9	16	23	2	9	16
26	Registro	2d	\$ 1.600,00	19/02/97	20/02/97							Programador 1
26	Consultas	2d	\$ 1.600,00	21/02/97	24/02/97							Programador
27	Prospectos	4h	\$ 400,00	25/02/97	25/02/97							Programador
28	Envío y Recopradón	1d	\$ 800,00	25/02/97	26/02/97							Programador
29	Módulo de Consultas y Reportes	3d	\$ 2.400,00	19/02/97	21/02/97							
30	Más de Tres Llamadas	4h	\$ 400,00	19/02/97	19/02/97							Programador 2
31	Distribución por Cuzca	1d	\$ 800,00	19/02/97	20/02/97							Programador 2
32	Oportunidades de Atención	4h	\$ 400,00	20/02/97	20/02/97							Programador 2
33	Reportes Abiertos	4h	\$ 400,00	21/02/97	21/02/97							Programador 2
34	Prospectos	4h	\$ 400,00	21/02/97	21/02/97							Programador 2
35	Módulo de Administración	3d	\$ 2.400,00	24/02/97	26/02/97							
36	Cuzca	4h	\$ 400,00	24/02/97	24/02/97							Programador 2
37	Ciudades	4h	\$ 400,00	24/02/97	24/02/97							Programador
38	Comercios	4h	\$ 400,00	25/02/97	25/02/97							Programador
39	Resolvidores	4h	\$ 400,00	25/02/97	25/02/97							Programador
40	Tipos de equipos	4h	\$ 400,00	26/02/97	26/02/97							Programador
41	Usarios	4h	\$ 400,00	26/02/97	26/02/97							Programador
42	Módulo de Sistema	2d	\$ 1.600,00	26/02/97	28/02/97							
43	Borrado	0,5d	\$ 400,00	26/02/97	26/02/97							Programador
44	Exportar	0,5d	\$ 400,00	27/02/97	27/02/97							Programador
45	Reindexar	0,5d	\$ 400,00	27/02/97	27/02/97							Programado
46	Respalidar	0,5d	\$ 400,00	28/02/97	28/02/97							Programado
47	Pruebas Internas	7,5d	\$ 18.000,00	19/02/97	28/02/97							
48	Revisión del Código	7,5d	\$ 0,00	19/02/97	28/02/97							



Ventanilla Electrónica de Servicio
Fecha: 6/05/97

Tarea		Retiempo	
Progreso de tarea		Tarea resumida	
Tarea crítica		Tarea crítica resumida	
Progreso de tarea crítica		Hito resumido	
Línea de base		Resumen de línea de base	
Margen de demora		Línea de base resumida	
Hito		Hito de línea de base resumida	
Hito de línea de base		Progreso resumido	

DIAGRAMA PERT DEL PROYECTO



Método PERT (Gráfica)

Estas pantallas muestran el diagrama de PERT del proyecto y cada una de sus actividades. Por medio de estos diagramas es posible observar la "Ruta Crítica del proyecto". Gracias a este tipo de esquemas es posible observar la forma en que se desarrolla el proyecto en sus diferentes etapas del mismo y las actividades que se llevan a cabo en forma paralela.

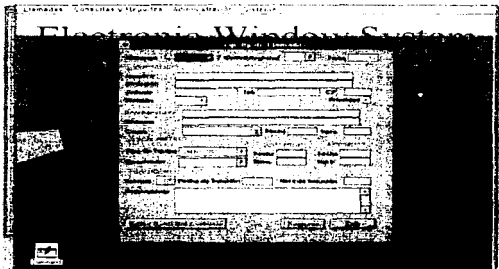
ANEXO 5
Pantallas del Sistema

PANTALLAS

INTRODUCCION

A continuación se muestran algunas de las pantallas más importantes del sistema con una breve descripción de las mismas. Cabe mencionar que aunque estas son las definitivas en el sistema se hicieron en base a un diseño funcional que fue revisado por el usuario final y el cliente.

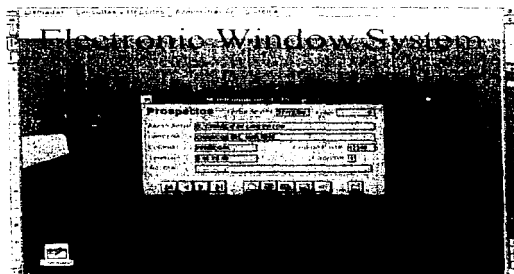
PANTALLA DE REGISTRO DE LLAMADAS



Por medio de esta pantalla se almacenan los datos del reporte y de los comercios que solicitan servicios de Mantenimiento a sus equipos. Como puede observarse se almacenan características del reporte y la fase en la que estos se encuentran: estatus fecha y hora del reporte, observaciones, síntomas, tipo de equipo que reporta, etcétera.

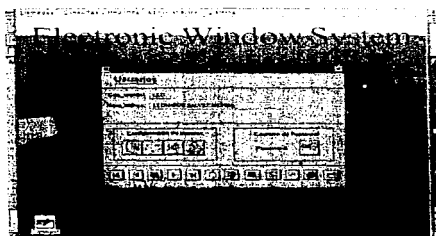
PANTALLAS

PANTALLA DE REGISTRO DE PROSPECTOS



A través de esta pantalla se registran las llamadas de clientes prospectos. Se toman los datos para registrar su información para canalizarlos a ventas para que evalúen la posibilidad de adquirir.

PANTALLA PARA EL REGISTRO DE REPRESENTANTES



A través de esta pantalla se hace el registro de los representante (usuarios del sistema) que habrán de utilizarlo para atender llamadas de los clientes. En esta pantalla se configuran sus accesos a los distintos menús del sistema, además se asignan USER-ID y password de acceso al sistema.

