

UNIVERSIDAD TECNICA DEL NORTE

FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERIA EN SISTEMAS

COMPUTACIONALES

PROYECTO DE TESIS PREVIO A LA OBTENCION DEL

TITULO DE INGENIERO EN SISTEMAS

COMPUTACIONALES

TEMA:

APLICACION DE LA INGENIERIA DE LA
INFORMACION A LA PLANEACION Y DESARROLLO
INFORMATICO

METODOLOGIA APLICADA A UN CASE

RESPONSABLE:

Jesús Octavio González Ortiz

DIRECTOR:

Ing. Rodrigo Naranjo

Ibarra

Mayo del 2001

DEDICATORIA	
AGRADECIMIENTO	
INDICE	
PROBLEMA	
JUSTIFICACION	
OBJETIVOS	
MARCO TEORICO	
HIPOTESIS	
METODOLOGIA	
INTRODUCCION	

CAPITULO I

1. LINEAMIENTOS Y POLITICAS DE PLANEACION Y DESARROLLO INFORMATICO	1
1.1. ORIGEN Y NECESIDAD DE LA PLANEACION	2
1.2. PLANEACION PARA QUE Y PLANEACION DE COMO	3
1.2.1. CUANDO SE DEBE EFECTUAR LA PLANEACION	4
1.2.2. PROBLEMAS QUE SE PRESENTAN EN LA PLANEACION	4
1.2.3. FALLAS EN EL ANALISIS	5
1.2.4. IMPORTANCIA DE LAS ESPECIFICACIONES FUNCIONALES	6
1.3. LA INFORMACION COMO RECURSO	8
1.4. PLANEACION PARA LA ORGANIZACIÓN	
1.4.1. RESPONSABILIDADES DE LA ORGANIZACIÓN	11
1.4.2. RESPONSABILIDADES DE DIRECTIVOS	13
1.4.3. LA PLANEACION Y EL AMBIENTE ADMINISTRATIVO	15
1.4.3.1. ANALISIS DE FLUJO DE TRABAJO	16
1.4.3.2. DISEÑO TECNICO	16
1.4.3.3. DISEÑO SOCIAL	17
1.4.4. ESTUDIO DE PROYECCIONES INFORMATICAS	18
1.5. PLANEACION DE RECURSOS	18

1.5.1. PLANEACION DEL EQUIPO	
1.5.1.1. ESTRATEGIAS DE SELECCION DE EQUIPO	20
1.5.2. PLANEACION DE SOFTWARE	21
1.5.2.1. REQUISITOS DE LAS HERRAMIENTAS	21
1.5.2.2. BENEFICIOS QUE DEBEN BRINDAR LAS HERRAMIENTAS	22
1.5.2.3. PROCESO DE DESARROLLO DEL SOFTWARE	24
1.5.2.4. EVOLUCION DE LAS TECNOLOGIAS DE DESARROLLO	24
1.5.2.5. LA INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADOR	25
1.5.2.6. HERRAMIENTAS CASE EN EL DESARROLLO DEL SOFTWARE	26
1.5.3. PLANEACION DE LA INFORMACIÓN	27
1.5.3.1. JERARQUIZACION DE LA INFORMACIÓN	29
1.5.3.2. DIVISION DE LA INFORMACION POR AREAS	30
1.5.3.3. PLANEACION ESTRATEGICA DE LA INFORMACIÓN	30
1.5.3.4. FUNDAMENTOS DE LAS HERRAMIENTAS CASE BASADAS EN LA INGENIERIA DE LA INFORMACIÓN	31
1.5.3.5. EL DICCIONARIO DE DATOS DE LA HERRAMIENTA CASE	33
1.5.4. PLANEACION PRESUPUESTARIA	34
1.5.5. PLANEACION DE LOS RECURSOS HUMANOS	34
1.5.5.1. DEPARTAMENTO DE INFORMATICA EN INSTITUCIONES	35
1.5.5.2. PERSONAL DE TODA LA ORGANIZACIÓN	35
1.5.6. CONSIDERACIONES ADMINISTRATIVAS Y TECNICAS	35
1.6. LA REINGENIERIA EN LA PLANEACION	
1.6.1. CONCEPTOS	36
1.6.2. PRINCIPIOS DE REINGENIERIA	38
1.6.3. PARAMETROS DE LA REINGENIERIA	38
1.6.4. DIVERSAS METODOLOGIAS DE REINGENIERIA	40
1.6.5. LA REINGENIERIA APLICADA A LA INGENIERIA DE LA INFORMACIÓN	42
1.6.6. REPROCESAMIENTO DE LA INFORMACIÓN	44
1.7. IMPLEMENTACION INFORMATICA	
1.7.1. IMPACTO SOBRE LA ORGANIZACIÓN	44

CAPITULO II

2. METODOLOGIAS DE DESARROLLO DE PLANIFICACIÓN INFORMATICA	
2.1. NECESIDAD DE UNA METODOLOGIA	47
2.2. CARACTERISTICAS Y ESTRUCTURA DE UNA METODOLOGIA	47
2.3. METODOLOGIAS ISP Y BSP	48
2.4. METODOLOGIA DE LA INGENIERIA DE SOFTWARE PARA PLANIFICACIÓN INFORMATICA	50
2.4.1. DESCRIPCION DE SUS PROCEDIMIENTOS	51
2.5. METODOLOGIA DE LA INGENIERIA DE LA INFORMACIÓN	
2.5.1. PRINCIPIOS Y FUNDAMENTOS	52
2.5.2. PROCESOS DE DESARROLLO	53
2.5.3. DIAGRAMAS UTILIZADOS	57
2.6. HERRAMIENTAS CASE	
2.6.1. RELACION DE LA CASE CON OTRAS TECNOLOGIAS	58
2.6.2. CATEGORIAS DE HERRAMIENTAS CASE	60
2.6.3. METODOLOGIAS QUE SOPORTAN LAS HERRAMIENTAS CASE	66
2.6.3.1. IMPLEMENTACION DE UNA METODOLOGIA EN UNA HERRAMIENTA CASE	66
2.6.4. PROCESOS QUE SOPORTA EL SOFTWARE CASE	67
2.6.4.1. DIAGRAMACION AUTOMATICA	68
2.6.4.2. VERIFICACION DE ERRORES	71
2.6.4.3 VERIFICACION DE INTEGRIDAD Y CONSISTENCIA	71
2.6.4.4. VERIFICACION DE DESCOMPOSICION FUNCIONAL	71
2.6.4.5. VERIFICACION DE LA METODOLOGIA	72
2.6.4.6. EL DEPOSITO CASE	72
2.6.5. CAMBIOS EN EL USO DE LAS HERRAMIENTAS CASE	77
2.6.6. HACIA DONDE SE DIRIGE LA CASE	79
2.7. ACOPLAMIENTO DE METODOLOGIAS.	79

CAPITULO III

3. ESTUDIO Y ANALISIS COSTO / BENEFICIO DE PLANEACION Y DESARROLLO INFORMATICO	
3.1. ESTUDIO DE EQUIPO DE PROCESAMIENTO	83
3.1.1. REQUERIMIENTOS DE LA ORGANIZACIÓN	83
3.1.2. CONSIDERACIONES DE EQUIPO EXISTENTE EN LA ORGANIZACIÓN	84
3.1.3. CONSIDERACIONES DE EQUIPO DEL MERCADO	84
3.2. ESTUDIO DE SOFTWARE	84
3.2.1. EVALUACION DE SOFTWARE DEL MERCADO	86
3.2.2. EVALUACION PARA EL DESARROLLO DE SOFTWARE EN LA ORGANIZACIÓN	88
3.2.2.1. TECNOLOGIAS DE DESARROLLO	89
3.2.2.2. LA CASE EN EL ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN	89
3.2.3. CAMBIOS EN EL PROCESO DE GENERACION DEL SOFTWARE	90
3.3. ANALISIS COSTO BENEFICIO	91
3.3.1. COSTO VS. CARACTERISTICAS TECNICAS	91
3.3.2. COSTO VS. PRODUCTIVIDAD	92
3.3.3. COSTO VS. BASE DE LA ADQUISICION	92
3.3.4. COSTO VS. PRACTICA ADMINISTRATIVA	92
3.3.5. COSTO VS. CONSIDERACIONES DEL PERSONAL	93

CAPITULO IV

	97
4. CONTROLES EN LA PLANIFICACION Y DESARROLLO INFORMATICOS	97
4.1. IMPORTANCIA DE CONTROLES ESPECIFICOS	98
4.2. COMITES ORGANIZACIONALES	98
4.2.1. IMPORTANCIA DE COMITES	98

4.2.2. RESPONSABILIDADES	99
4.2.3. SELECCIÓN DE SUS MIEMBROS	
4.2.4. DEPARTAMENTO DE AUDITORIA	100
4.2.5. RELACION ENTRE COMITÉ ORGANIZACIONAL Y DEPARTAMENTO DE AUDITORIA.	101
4.3. CONTROLES EN EL PROCESO DE PLANEACION INFORMATICA	102
4.4. CONTROLES EN LA APLICACION DE LA PLANEACION	
4.5. CONTROLES EN EL PROCESO DE DESARROLLO INFORMATICO	102
4.6. IMPLEMENTACION DE VERIFICACION DE ERRORES EN LA HERRAMIENTA CASE	103
4.7. ANALISIS DE LA LEGISLACION ECUATORIANA	103
4.8. ANALISIS DE LA LEGISLACION INTERNACIONAL GENERALMENTE ACEPTADA	

CAPITULO V

5. IMPLEMENTACION DE UNA HERRAMIENTA CASE PARA LAS FASES DE ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION BASADO EN LA METODOLOGIA DE LA INGENIERIA DE LA INFORMACIÓN	114
5.1. OBJETIVOS DE LA APLICACION CASE	115
5.2. CATEGORIA DE LA HERRAMIENTA CASE	
5.3. PROCESOS QUE SOPORTA LA APLICACION CASE	
5.4. COMPONENTES Y DESCRIPCION DEL SISTEMA CASE A IMPLEMENTAR	116
5.4.1. DICCIONARIO DE DATOS	117
5.4.2. BANCO DE TRABAJO	118
5.4.3. DIAGRAMACION AUTOMATICA	
5.5. PROCESOS DE VERIFICACION	118
5.6. DESARROLLO Y CODIFICACION DE LOS COMPONENTES DEL SISTEMA CASE	

CAPITULO VI	125
6. VERIFICACION DE HIPOTESIS Y CONCLUSIONES	
CAPITULO VII	128
7. RECOMENDACIONES	
	129
ANEXOS	172
BIBLIOGRAFIA	

DEDICATORIA

A mis padres, por ser quienes me dieron la vida y dedicaron sus mejores años para entregarme lo que ellos no tuvieron.

A mi esposa, por su entrega y esfuerzo en concebir una vida mejor.

A mi hijo, por su inocencia y fortaleza que nos transmite, dentro de su propio mundo.

AGRADECIMIENTO

A la Universidad Técnica del Norte, por permitir que la Ciencia y Técnica estén al servicio del Pueblo.

A mis compañeros de estudio, por la entrega sin restricciones de una amistad que nos une en nuestro diario vivir y saber que siempre podemos extendernos la mano.

A todos los Docentes Universitarios por haberme transmitido sus conocimientos en beneficio del engrandecimiento de nuestro país.

A mi Director de Tesis por la ayuda y guía brindada. Más que un Director un Amigo.

APLICACION DE LA INGENIERIA DE LA INFORMACION A LA PLANEACION Y DESARROLLO INFORMATICO

METODOLOGIA APLICADA A UN CASE

PROBLEMA:

El cambio tecnológico es de gran relevancia para la estrategia organizacional por algunas razones. En muchos sectores la adecuada gestión de la tecnología es uno de los factores claves para conseguir ventajas competitivas. Las innovaciones tecnológicas pueden contribuir a cambiar la estructura de un sector o incluso a la creación de otros.

Producto de todo este cambio tecnológico a dado como resultado que la sociedad actual está envuelta en una revolución de la información. Para hacer frente a esta transformación, la dirección de las organizaciones necesitan adecuar y estructurar desde una perspectiva estratégica un marco conceptual y analítico de cómo utilizar su información y la del entorno en beneficios para sí mismos.

No es desconocido que toda organización genera diariamente gran cantidad de información, la misma que se requiere para tomar decisiones o encaminar acciones tendientes a mejorar el rendimiento de dicha organización. Esta acumulación de información se vuelve paulatinamente un gran obstáculo y que puede dar como resultado contratiempos en trabajos, debido a que muchas de las veces, dicha información esta mal recopilada o procesada y por lo tanto es no confiable.

Ante este inminente problema y expansión de la información, una organización se ve, entonces, en la necesidad de automatizar los procesos más fundamentales, para lo cual es imprescindible saber cuál es el flujo y cultura de la información. Podemos

determinar que al automatizar los procesos de la organización, la estructura y funciones de los Sistemas de Información deben ser coherentes con la estrategia de la misma en cada momento; es decir, deben estar acordes con los objetivos, planes y metas de la organización. Por lo tanto, la planificación y el diseño de un Sistema de Información deben tener acceso a los objetivos estratégicos de la organización.

En este sentido, el empleo de una tecnología que ayude a vencer este obstáculo debe ser una prioridad de toda organización. Hay que considerar que una tecnología CASE “no siempre tiene éxito”, principalmente porque en las organizaciones que la tecnología fracasó, “no supieron reconocer la complejidad de sus problemas”. Hay que plantearse y estructurar en cualquier organización el concepto de proceso de negocio para lograr mejoras sustanciales en la manera de trabajar y obtener beneficios¹ en las organizaciones con la ayuda de las tecnologías y herramientas de desarrollo.

Debido a que el concepto de organización, no sólo corresponde a la parte física (edificios, bienes muebles, etc), o sólo a la parte de Recursos Humanos, sino que se debe relacionar íntimamente a la naturaleza, objetivos, metas; se debe definir un adecuado proceso de planeación de manejo de información, la cuál debe incluir todas las necesidades de una organización tales como:

- ↻ Formación adecuada de su estructura organizacional.
- ↻ Implementar estructuras centralizadas y/o descentralizadas.
- ↻ Analizar la formación sobre herramientas o sobre metodologías que el personal de desarrollo posee.
- ↻ Implementar estándares y procedimientos de gestión de proyectos.
- ↻ Determinar parámetros para medir la productividad y la calidad.

Definir necesidades técnicas como:

¹ Los beneficios pueden ser de tipo lucrativo (económicos) o mejoras en la prestación de servicios.

- ® Automatización del proceso para luego desarrollar el software.
- ® Utilización de una metodología disciplinada para organizar el trabajo.
- ® Emplear una plataforma hardware, para el desarrollo y mantenimiento del software.
- ® Desarrollar estándares y procedimientos técnicos para garantizar la calidad de la planeación.
- ® Analizar la actitud del personal en relación a la planeación y posterior desarrollo informático.

De todas estas interrogantes, una manera formal de plantear el problema es:

“La realización de una eficaz automatización, que permita mantener un mayor control con apego a los principios de integración y unificación de sistemas de información, necesita de la implementación de procedimientos de planeación y desarrollo informático que estén orientados y diseñados de acuerdo a los requerimientos de información y datos que necesita una organización”.

JUSTIFICACION:

La humanidad se halla en medio de una explosión de la información, un fenómeno tan amplio que está cambiando el aspecto de toda nuestra sociedad. Entender la expansión informativa ayudará a situar en perspectiva la trascendencia de la información administrativa para analizar y explorar el área informática.

Durante los años 80 el investigador John Naisbitt, en su libro, Megatrends: Ten directions transforming our lives, amplió el significado de la explosión informativa. La más dramática de las tendencias de Naisbitt es el reconocimiento de que la sociedad ha sido impelida de una sociedad industrial a una sociedad de la información.

Los avances en la tecnología computacional y de la información de la última década han cambiado la situación que enfrenta la directiva de una organización: “cómo implantar las computadoras y cómo usarlas eficientemente”. Así mismo, dentro de organizaciones que ya posean infraestructura informática se debe analizar algunas consideraciones técnicas de suma importancia como: equipo existente y su estado, probable caída en desuso tecnológico, adquisición y/o reemplazo de equipo y herramientas de software.

A la hora de tomar decisiones sobre estos problemas, cada organización pública o privada, es completamente autónoma para saber que adquirir o no, para qué y con qué fin. Por esta razón, se debe tener muy en cuenta, que cualquier acción a seguir estará guiada en base a objetivos generales de la organización, metas, cambios y proyecciones de crecimiento interrelacionados y razonables con completa identificación a la realidad organizativa y de su entorno de influencia.

De la misma forma, al decidir sobre como implantar sistemas de información muchas de las veces no se toma en cuenta un sinnúmero de interrogativas como por ejemplo: pensamiento del personal, cuál es la mejor metodología que se aplique a la organización, flujo de trabajo, correlación organizacional de tareas y actividades, valor del proceso, entre otras. Así mismo por parte de toda la organización, siempre se está con la expectativa: “a qué hora se usará el nuevo sistema automatizado”, creando así una especie de presión innecesaria hacia el personal de sistemas y junto con una mala práctica y tentación de desarrollar cuanto antes posible el bosquejo físico del nuevo sistema, no se estudia y comprende completa y detalladamente todos los requerimientos lógicos y de información del sistema.

Dados todos estos motivos, se asevera que la planificación y no la improvisación, es un requisito importante para el desarrollo e implementación de recursos informáticos y para proyectar su futuro en función de las necesidades del sector productivo y sociedad.

Desde otro punto de vista, la planificación sin su respectivo control, tendría un discernimiento algo tambaleante, de tal suerte que, sin un seguimiento específico y dedicado, los objetivos de tal planeación pudieran desviarse o no cumplirse.

De la mala ejecución o aplicación de la planeación informática en la organización, ésta corre el riesgo de presentar, luego de algún avance significativo en la ejecución, errores o inexactitudes que sean muy notorios y ocasionen una atención considerable de los medios masivos, al cual tiene influencia, así como pérdidas financieras.

Para que no se produzcan estos efectos negativos, se tiene la necesidad de analizar y ejecutar procedimientos de controles tanto generales como aplicativos que funcionan desde el momento mismo de la planeación, ejecución y con los procesos de verificación y replanteamiento inherentes a la Ingeniería de la Información. También se debe tener en cuenta controles que vigilen los resultados obtenidos.

La creación de una herramienta CASE que sea un soporte para las fases de planeación y desarrollo informático, se convierte en una poderosa ayuda a la automatización.

Dentro de la tecnología CASE hay aspectos del software, del hardware, de la metodología y de la gestión.

Su desarrollo estará dirigido hacia las fases de análisis y diseño de sistemas de información debido a que, si bien es cierto existen algunas herramientas CASE en el mercado, estas no están al alcance de todas las instituciones, por lo que se presenta como una buena alternativa de desarrollo de software. Esta herramienta tendrá un Banco de Trabajo (Workbench), que soportará la metodología de la Ingeniería de la Información.

El porqué usar la Metodología de la Ingeniería de la Información, se debe a que tiene otro tipo de enfoque de desarrollo a alto nivel que comienza con la Planificación Estratégica de la Información hasta llegar a la construcción del sistema. Todo el análisis está basado alrededor de la información que generan las diferentes áreas de la organización y se aplica más específicamente hacia el desarrollo de Sistemas de Información. De esta forma utilizando la misma metodología aplicarlo en la Planeación y Desarrollo Informático.

OBJETIVOS:

Generales:

- Aplicar la Ingeniería de la Información a la Planeación y Desarrollo Informático.
- Desarrollar procedimientos de planeación y desarrollo informático para mantener un mayor control, integración y unificación en el proceso de construcción de Sistemas de Información.

Específicos:

- Determinar la relación existente entre las herramientas CASE y la Planeación y Desarrollo Informático.
- Proponer procedimientos, métodos y técnicas de gestión que ayuden a resolver los requerimiento de Planeación y Desarrollo Informático aplicados al control de la Información.
- Desarrollar una herramienta CASE que aplique técnicas de la Ingeniería de la Información para sistematizar las fases de análisis y diseño de Sistemas de Información.
- Evaluar los beneficios de la Metodología de la Ingeniería de la Información y su aplicabilidad al desarrollo de Sistemas de Información.

MARCO TEORICO:

Nuestro recurso más grande es la información, es decir datos que tiene significado y que transfieren conocimientos utilizables. No sólo es esencial en la vida moderna, sino también en la administración y ejecución de la mayoría de actividades de toda organización. Esta información se ha incrementado con mucha rapidez y, junto con ella la necesidad de más y mejor administración de ella. En consecuencia la información es un “agente catalizador de la moderna administración”.

Las funciones propias de la administración empresarial son la planeación, organización, actuación y control. En la diversidad de todas las organizaciones cada una sufre su propia y personal crisis informática, por lo que no hay una solución universal, sin embargo, la clave del éxito de un proyecto de desarrollo informático, es la elección correcta de una metodología que ayude y refuerce a las funciones de la administración.

Es importante recordar que no todas las operaciones manuales son menos eficaces y más costosas que los medios automáticos, no todos los problemas de información se solucionan con sólo instalar equipos con tecnología de punta y que no todas las organizaciones tienen necesidades de información que justifiquen sistemas informáticos muy sofisticados.

La determinación de que información y a quién debe ir es esencial. Los directivos necesitan saber que tipos de equipos de procesamiento existen en el mercado, sus costos y su contribución para el logro del objetivo organizacional.

Un verdadero reto para las organizaciones, en general, radica no en adquirir la tecnología de los sistemas de información, sino en saber administrarla y desarrollarla para su utilización productiva.

De todo este análisis, concluimos que “el resultado de una buena y efectiva planeación determinará en gran medida el éxito o fracaso del desarrollo informático”.

De esta planificación se desprenderá cuestionamientos como: qué se desea y puede lograr, cómo lograrlo, cómo actuar en caso de posibles variaciones del medio, adaptaciones del personal, impacto organizacional.

Puntos muy importantes de la planeación, constituyen la adaptación de estrategias frente a la utilización de metodologías de desarrollo.

Es imprescindible que se adopte una metodología única para toda la organización, así como también sea la más adecuada a su realidad. De la misma forma se debe normar el uso de herramientas de automatización y el desarrollar estándares y controles de procesamiento.

Estos estándares son importantes, debido a que, por lo general los proyectos quedan fuera de control o desvían su objetivo final, por no haberlos impuesto a tiempo. Por su parte la metodología nos servirá para que todo el desarrollo informático esté enmarcado dentro de un sólo contexto y de esta forma cuantos participan en un proyecto mantengan un sistema unificado e integral, participativo, realista y progresivo. En cuanto a las herramientas hay que aclarar que estas no proporcionan la metodología, sólo automatizan la asistencia a esta.

Una de estas herramientas, son los CASE, que son imprescindibles debido a que su depósito CASE (mecanismo para almacenar y organizar toda la información), incluye “información sobre el problema que se va a resolver, procesos que están siendo utilizados, modelos de datos, modelos de procesos, prototipos, historia y recursos del proyecto y el contexto organizativo”.

La tecnología CASE es una combinación de herramientas de software y de metodologías, propone una nueva formulación del concepto de ciclo de vida de Software. Un CASE a cambiado la forma de construir sistemas al poseer un ambiente de desarrollo interactivo, programación visual proporcionado por interfaces gráficas y automatización de tareas de desarrollo y mantenimiento del software.

HIPOTESIS:

- * Aplicando correctamente los métodos de la Ingeniería de la Información a los procesos de planificación y desarrollo informático se obtendrán mejores técnicas en la construcción de Sistemas de Información.
- * Usando una herramienta CASE se obtendrá un mejor resultado en el Análisis y Diseño de Sistemas de Información.

METODOLOGIA:

El análisis de este problema estará basado en investigaciones documentales aplicando para el efecto: técnicas de fichaje, lectura científica, consultas en correo electrónico. Se basará también en experiencias adquiridas durante el ejercicio de trabajos institucionales y/o en citas a consultoras del área.

Se aplicarán también encuestas, entrevistas con diferentes instituciones públicas y/o privadas, para tener un conocimiento de hacia donde van las organizaciones y que procedimientos, metodología y técnicas están usando en materia de Planeación y Desarrollo Informático.

Toda esta investigación servirá de base para analizar la situación actual, de cómo actúan, proceden y toman decisiones, ciertas empresas o instituciones, que en materia de planificación y desarrollo informático estas realizan.

Para el diseño y propuesta de la Planeación, se utilizarán enfoques de Ingeniería de la Información, la cual considerará áreas estratégicas existentes en toda organización, para establecer un mayor control hacia la información y su flujo existente en una organización. Al realizar la planificación de esta manera, se tendrá una perspectiva amplia, que exigirá la participación activa de la o las personas que estén en los mandos directivos.

Debido a que se toma como base y principal recurso a la información, ésta será analizada en concordancia con los objetivos y fines de la organización, para de esta forma alcanzar una ventaja en la integración del personal de la organización ya que son parte de ella y la planeación también lo es.

El software CASE se desarrollará sobre herramientas visuales para lo cual se escogerá entre VISUAL BASIC, VISUAL C u otra herramienta que preste mejores posibilidades, librerías y bibliotecas de desarrollo.

La herramienta CASE se fundamentará en técnicas de la Ingeniería de la Información, al realizar la Planeación en base a esta misma metodología, se determinará que relación existe al realizar la Planeación dirigida hacia la utilización de una herramienta para análisis y diseño.

INTRODUCCION

Desarrollar un proceso de Planeación y Desarrollo Informático, merece especial cuidado debido a todas las implicaciones que de este se derivan.

El acceso a recursos tecnológicos; computadoras, sistemas automatizados, etc., a producido un cambio en la forma de vida y de trabajo de todas las organizaciones. El cambio social radica en que hoy se necesita una mejor cualificación de nuestros conocimientos y mayores exigencias en las ciencias computacionales para acceder a puestos de trabajo que tengan el carácter de operativos administrativos. Todas estas exigencias a dado campo para desarrollar otras áreas del saber humano.

La implantación de estas tecnologías, conlleva a estudiar aspectos del comportamiento humano y organizacional. Cómo nos afecte su utilización y adaptación, dependerá del éxito, fracaso o mediocridad de nuestras instituciones.

Es por esto que, se desarrolla tópicos del Planeación de Recursos en el Capítulo I. Estos recursos ya no solamente son la mano de obra, el capital y las herramientas. También se trata los recursos de la información, las tecnologías, la organización. Así como también, la reingeniería para ser aplicada sobre la base estructural ya existente.

En el Capítulo II, se analiza algunas metodologías de desarrollo que involucran no sólo al análisis de los problemas o áreas por resolver; sino que, se analiza desde la misión misma de la institución.

Como paso previo para el desarrollo mismo de la planeación, se debe considerar cual va a ser el beneficio en relación con los costos que esta tendrá. Es por esto, que el Capítulo III, da enfoque generales de qué parámetros deberían regir un análisis de este tipo.

Todo proceso humano tiene implicaciones susceptibles de errores. Es por esto, que en el Capítulo IV se propone referencias de cómo poder establecer mecanismos de control para el proceso de planeación y desarrollo informático.

Al ser este un análisis de la aplicación de la Metodología de la Ingeniería de la Información formulada por James Martin, dentro de la estructura de los capítulos mencionados se hace referencia y enfoques desde dicha metodología.

En el Capítulo V, se plantea un software CASE que ayudará a las fases de análisis y diseño de sistemas de información; y finalmente proponer los resultados en las conclusiones de este trabajo, así como también dar las respectivas recomendaciones.

Creo que este trabajo de investigación, por ser tratado desde una panorámica amplia como lo es, la misión estratégica de cada institución; su aplicabilidad depende en gran medida desde las decisiones que los directivos o autoridades responsables de llevar la conducción de una organización así lo crean conveniente. Sin embargo deseo resaltar el carácter de que una propuesta de planeación, no sólo debe ser elaborada por Planificadores Expertos o Certificados. Las planificaciones no tienen mucho éxito o el éxito deseado, sin la concurrencia y aceptación de todas personas que directamente se verán involucradas, de una u otra manera, en este proceso. No con esto quiero decir, que todo debe hacerse siempre y cuando todas las personas se pongan de acuerdo; tal vez algo que queda en la imaginación.

Deseo resaltar las palabras de Lucas Achig cuando manifiesta: “ el planeamiento universitario, en lugar de convertirse en el motor de desarrollo y cambio institucional se transformó en mediador y procesador de la coyuntura, entrampándose en el presente en lugar de construir el futuro”; y prosigue diciendo:”...Este planeamiento tiene problemas en su ejercicio dinamizador por cuanto, aún la toma de decisiones en la vida universitaria, está llena de mucho ‘emocionalismo’”.

A pesar de que realiza un análisis desde la perspectiva universitaria, creo que sus palabras bien pueden aplicarse al contexto nacional, cuando desde el mismo Estado Ecuatoriano han manejado al país de acuerdo a conveniencias de quienes ostentan el poder.

La incorporación de sistemas informáticos y recursos computacionales dentro de las organizaciones, es un punto clave, para poder meditar hacia dónde queremos llegar con estos sistemas y cuál debe ser el comportamiento en adelante de cada una de nuestras instituciones.

Finalmente pido disculpas por cualquier error involuntario que se encuentre presente en este documento.

El Autor

CAPITULO I

1. LINEAMIENTOS Y POLITICAS DE PLANEACION Y DESARROLLO INFORMATICO

Una política es: una guía para las decisiones administrativas, una ley administrativa interna que rige las decisiones.

El concepto de planeación va ligado al desarrollo de las empresas; analicemos el desarrollo de estas:

- a. La empresa como máquina: “Las organizaciones eran consideradas como máquinas cuya función era servir a sus creadores, proporcionándoles una retribución sobre su inversión en dinero y tiempo”².
- b. La empresa como organismo: “Después de la Primera Guerra Mundial, emergió el nuevo concepto de empresa considerándola como organismo. Al cual se le atribuía vida y propósitos propios que se suponían eran la supervivencia y el crecimiento”³.
- c. La empresa como organización: “Una organización es: 1) un sistema con algún propósito, el cual 2) es parte de uno o más sistemas con algún propósito, y 3) en el cual alguna de sus partes tienen sus propios propósitos”⁴.

La automatización e informatización se introduce y difunde a partir de la segunda era de las empresas, el cual aumentó significativamente el contenido técnico de muchos trabajos.

Además, como veremos más adelante, la información es tratada de acuerdo a cada era de las empresas de diferente manera, llegando a ser considerada en los actuales

² Ackoff Russell. *Planificación de la Empresa del Futuro*. Pg. 41

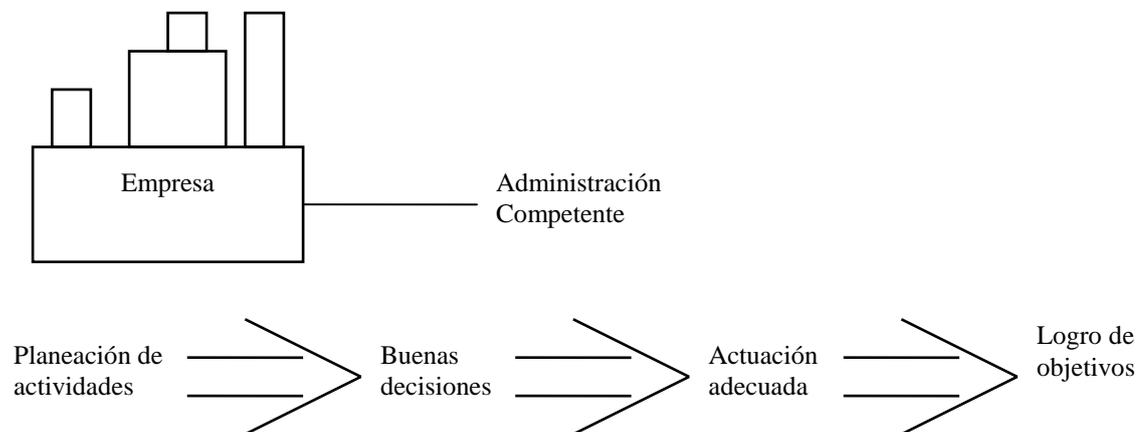
³ Ackoff Russell. *Planificación de la Empresa del Futuro*. Pg 43

⁴ Ackoff Russell. *Planificación de la Empresa del Futuro*. Pg 46

momentos como un recurso preponderante para la administración de una organización.

1.1. ORIGEN Y NECESIDAD DE LA PLANEACION

La planeación estratégica formal con sus características modernas fue introducida por primera vez en algunas empresas comerciales a mediados de 1950; que las denominaron sistemas de planeación a largo plazo.⁵



Planeación y Organización de Empresas, Gómez Ceja Guillermo 8va. Edición 1996, México PG 6

FIGURA 1 Necesidad de la Planeación

Toda empresa y/o institución, para que pueda sobresalir y sostenerse, necesita que su administración sea competente. En la Figura 1, podemos apreciar que la planeación de actividades, es decir, que las previsiones de trabajo que se realiza deben conducirnos a tomar las mejores decisiones y éstas unidas a la actuación adecuada de todos los miembros de una organización, que apoye las decisiones, permitirá lograr los objetivos que se han planteado conseguirlos.

⁵ Steiner George. *Planeación Estratégica*. Prefacio

La necesidad de planear se deriva del hecho de que toda organización opera en un medio que experimenta constantes cambios, dentro de un futuro incierto y desconocido, en el cual la tasa de desarrollo tecnológico es acelerada, las expectativas ambientales se han engrandecido, las necesidades humanas son mayores, lo que podríamos resumir como necesidades administrativas, ambientales y humanas.

1.2. PLANEACION PARA QUE Y PLANEACION DE COMO

El propósito de la planeación es determinar lo que debe hacerse en un determinado período de tiempo, con recursos actuales asignados para el efecto y futuros recursos que deberán conseguirse para lograr el objetivo deseado. Además, no debe eliminar o menospreciar ningún tipo de riesgo y prevenir y superar las crisis que en el transcurso de las operaciones se presenten. Intenta asegurar el uso efectivo de los recursos disponibles.

Ningún esquema de planeación es un recetario, ni mucho menos un decálogo que debe seguirse al pie de la letra; sin embargo podemos mencionar que como base para desarrollar la planeación en una organización se debe considerar que en una organización existen e interactúan: máquinas, ambientes físicos, área de influencia (sector o comunidad sobre la cual tiene incidencia la presencia de la organización), dinero, sistemas, relaciones de trabajo y personas; por lo que resumiendo podríamos decir que es la conjunción de tecnología, dinero, entorno y personas.

Por lo que, se debería tener en cuenta que una planeación debe contemplar:

- a. Fines de la Organización.
- b. Diseño de la Organización.
- c. Relaciones y líneas de autoridad.
- d. Manuales de Funciones y de Procedimientos.
- e. Ambiente Interpersonal.

- f. Estabilidad emocional y preparación intelectual de los miembros de la organización.
- g. Sistemas de trabajo presentes en la organización.
- h. Nivel de influencia de la organización sobre la comunidad.
- i. Visión de futuro de la organización.
- j. Propuestas de parte de los miembros de la organización sobre optimización de procedimientos.
- k. Presupuestos financieros.

1.2.1. CUANDO SE DEBE EFECTUAR LA PLANEACION

Toda organización debe tener una metodología de planificación; debido a que, sin ésta, su administración sería de acuerdo a las improvisaciones. Es por este motivo que no existe tiempo alguno en el cual una entidad debe empezar a realizar su planeación. Es deber de la directiva quien decidirá cuando empezar; sin embargo, debido al continuo estado cambiante del mundo no puede dejar de prescindir de tan importante herramienta de ayuda en la administración.

Dependerá en gran medida, de la metodología a usar, la que determinará el tiempo de duración en la ejecución de la planeación.

1.2.2. PROBLEMAS QUE SE PRESENTAN EN LA PLANEACION

La introducción de cualquier metodología de planeación, da lugar a conflictos, resistencias, oposición, inseguridad, insatisfacción, etc.

Esto se debe, en gran medida, a la no participación e información a todos los miembros de la organización y, tal vez, a las acciones tareístas, inmatiatistas que en muchas de las organizaciones se tiene como una política de facto.

Dentro de los principales problemas que generan prejuicios, podemos identificar:

- a. Alteraciones en las relaciones interpersonales.
- b. Cambio en los flujos de la información.
- c. Sentimiento de conflictos de autoridad.
- d. Nuevas exigencias para todos los empleados de la organización.
- e. Estado de inseguridad.
- f. Temor a fracasar.
- g. Intereses grupales por sobresalir y/o prevalecer en la estructura orgánica.

La mayor parte de los problemas de una planeación se debe principalmente a que aún las organizaciones no conciben a sus miembros como parte activa de la misma, sino más bien, como simples operarios del sistema. Como se expuso anteriormente, toda institución hoy en día, debe ser tratada como un organismo y asumir la teoría general de sistemas.

1.2.3. FALLAS EN EL ANALISIS

Las fallas se puede establecer a partir de la identificación de riesgos que la organización puede sufrir.

Para identificar riesgos se puede crear una lista de comprobación de elementos de riesgo. La lista de comprobación se puede utilizar para identificar riesgos y se enfoca en un subconjunto de riesgos conocidos y predecibles en las siguientes subcategorías genéricas:

- Tamaño del producto: riesgos asociados con el tamaño general del software a construir o a modificar.
- Impacto en la organización: riesgos asociados con las limitaciones impuestas por la gestión, los empleados o por el entorno.

- Características del usuario: riesgos asociados con la sofisticación del usuario y la habilidad del desarrollador para comunicarse con él en los momentos oportunos del diseño.
- Definición del proceso: riesgos asociados con el grado de definición del proceso del software y su seguimiento por la organización de desarrollo.
- Entorno de desarrollo: riesgos asociados con la disponibilidad y calidad de las herramientas que se van a emplear en la construcción del producto.
- Tecnología a construir: riesgos asociados con la complejidad del sistema a construir y la tecnología que contiene el sistema.
- Tamaño y experiencia de los desarrolladores del software: riesgos asociados con la experiencia técnica y de proyectos de quienes van a realizar el trabajo.

1.2.4. IMPORTANCIA DE LAS ESPECIFICACIONES FUNCIONALES

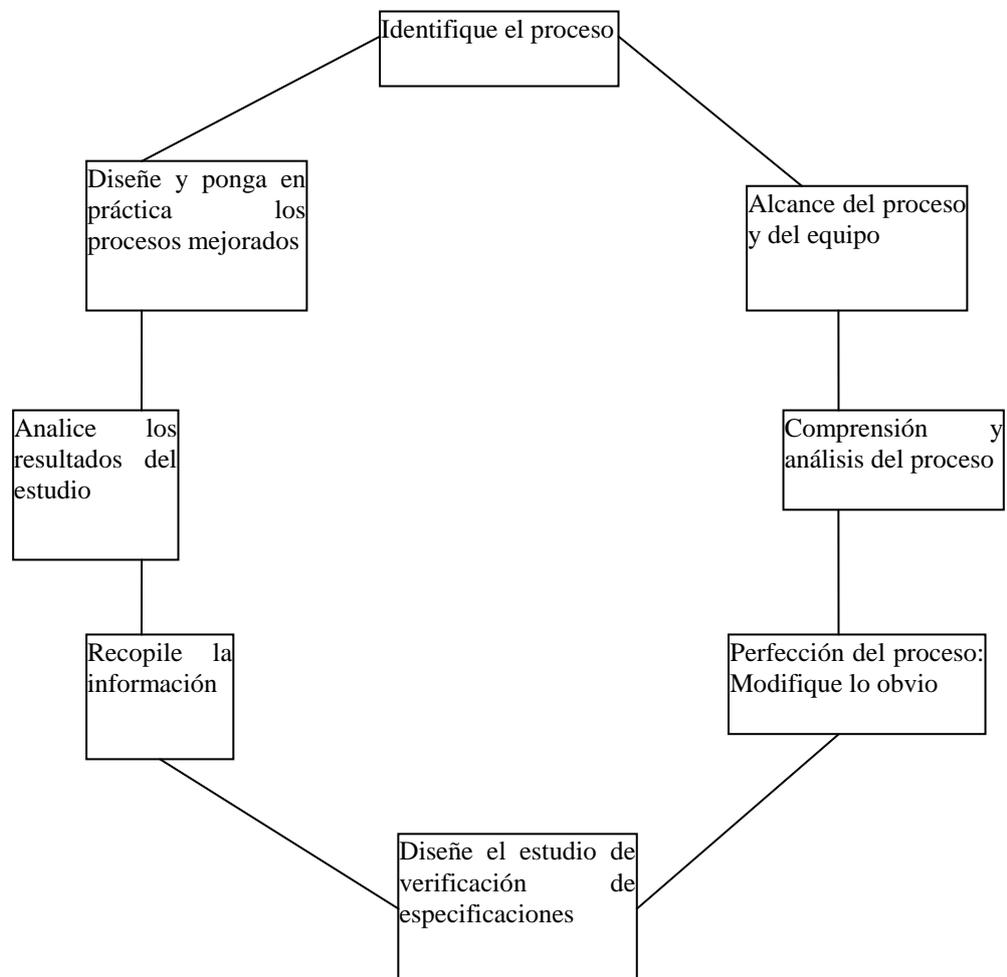
Las especificaciones funcionales tienen su base en los principios del análisis de sistemas:

- Debe representarse y entenderse el dominio de información de un problema.
- Deben definirse las funciones que va a realizar el software.
- Debe representarse el comportamiento del software.
- Debe dividirse los modelos que representan información, función y comportamiento de manera que se descubran los detalles por capas.
- El proceso de análisis debería ir desde la información esencial hasta el detalle de la implementación.⁶

⁶ Pressman Roger. *Ingeniería De Software Un Enfoque Práctico*. Pg 188

Utilizando y desarrollando estos principios se puede examinar detalladamente el origen de cada proceso, tarea y de cada dato que se necesite como entrada y/o salida del sistema.

Estas se utilizan para organizar los requisitos del usuario. Varios usuarios pueden tener varias especificaciones que concuerden con otros usuarios; en estos casos forman grupos de especificaciones funcionales. Para poder asegurarse que las especificaciones funcionales están completas y además identificar redundancias, las funciones deben estar referenciadas en cada entidad.



La esencia de la Reingeniería de Procesos. Peppard. PG 168
 FIGURA 2 Verificación de Especificaciones

Las especificaciones funcionales, dentro de todo el proceso de desarrollo informático, representan con objetividad a cada proceso del entorno global de una organización. Es así, que en la Figura 2, podemos apreciar que se propone en primera instancia la identificación de un proceso a desarrollar. Luego de lo cual. Es importante determinar el alcance del proceso, para tener en cuenta cual será el ámbito de acción de dicho proceso; juntamente con esto la comprensión y análisis del mismo nos permitirá, modelar el proceso y poseer una base sobre la cual se trabajará. La actividad de perfección, nos permite depurar los pasos que pueden estar repetidos en un mismo proceso o que pueden ser asumidos (embebidos) en otros; sin que estas reducciones de pasos, afecten al conjunto del proceso. Una vez revisado el modelo, las especificaciones funcionales, deberán proporcionar mayor y mejor información sobre cual será por ejemplo: el comportamiento del software, el dominio de la información, modelos que usa, etc.. Los pasos de recopilación de información y análisis de resultados, deberán ser producto del nuevo modelo a desarrollar y tendientes a comprobar si los cambios que se estima realizar son los más adecuados. Finalmente, como debe ser obvio, se diseñará e implementará todo el modelo realizado.

1.3. LA INFORMACION COMO RECURSO

En las últimas décadas se ha dado una revolución en la manera en que las instituciones tratan a la información y a los sistemas de información. En la actualidad, las instituciones emplean la información y los sistemas de información como herramientas; los cuales se han constituido en los Sistemas Estratégicos de Información.

Los sistemas estratégicos de información cambian las metas, operaciones, productos o relaciones con el entorno de las instituciones para ayudarlas. Cambian a la institución así como a los servicios y procedimientos internos, llevándola a nuevos patrones de comportamiento. Las instituciones requieren de un cambio en

sus operaciones internas para sacar ventaja de las nuevas tecnologías de los sistemas de información.

Estas (las instituciones) actualmente consideran a la información como un recurso, a la manera del capital, la mano de obra y herramientas.

Período	Concepción de la Información	Sistemas de información	Finalidad
1950-1960	Requerimiento burocrático necesario y perverso. Un dragón de papel	Los sistemas de información correspondientes a este período se denominaban máquinas electrónicas de contabilidad	Velocidad en la contabilidad y en el procesamiento de documentos
1960-1970	Apoyo a propósitos generales	Sistemas de información para la administración de información a ser presentada	Velocidad a los requerimientos generales de fábrica
1970-1980	Control administrativo	Sistemas de soporte a mejorar y adaptar las decisiones del usuario	Sistema de soporte para la toma de decisiones gerenciales
1985-2000	Recurso estratégico Ventaja competitiva Arma estratégica	Sistemas estratégicos	Promover la sobrevivencia y prosperidad de la institución

TABLA 1 Conceptos Cambiantes De Los Sistemas De Información:

Anteriormente, la información se consideraba a menudo como un mal necesario asociado con la burocracia para el diseño, la fabricación y la distribución de un producto o servicio. La información era un "dragón de papel" que potencialmente

podía estrangular a la organización y evitar que realizara su verdadera función. Los sistemas de información de los noventa se enfocaban a la reducción del costo del procesamiento rutinario de documentos, en especial en la contabilidad.

En los sesenta, las instituciones empezaron a ver la información de forma diferente, reconociendo que ésta podía ser usada en general para dar soporte a la administración. Los sistemas de información de los sesenta y los setenta en general se denominaban Sistemas de Información para la Administración (MIS) y se pensaba que eran una especie de fábrica que producía informes sobre la producción semanal, información mensual de tipo financiero, de inventarios, cuentas por cobrar, cuentas por pagar y otras.

En los setenta y a principios de los ochenta, la información y todos los sistemas que la recolectaban, almacenaban y la procesaban, fueron vistos como proveedores de control administrativo, muy afinado, de propósitos especiales y sobre medida para la institución. Los sistemas de información que surgieron durante este período recibieron el nombre de Sistemas de Soporte de Decisiones (SSD) y Sistemas de Soporte Gerencial (SSG). Su finalidad era mejorar y acelerar el proceso de toma de decisiones de ciertos ejecutivos y administradores en una vasta gama de problemas.

A mediados de los ochenta, la información fue vista como un recurso estratégico, como una fuente potencial de ventajas competitivas. Estos cambios de conceptualización sobre la información reflejan avances en la planeación estratégica.

Los sistemas estratégicos de información deben diferenciarse de los sistemas al nivel estratégico para los directivos que se enfocan en problemas de toma de decisiones a largo plazo. Los sistemas estratégicos de información pueden ser usados por todos los niveles de la institución. Estos fundamentalmente modifican las metas, productos, servicios y relaciones internas o externas de la empresa.

Estos sistemas modifican profundamente la manera como una institución lleva a cabo sus operaciones.

1.4. PLANEACION PARA LA ORGANIZACIÓN

1.4.1. RESPONSABILIDADES DE LA ORGANIZACIÓN

Una organización es la estructuración técnica de las relaciones que debe existir entre las funciones, niveles y actividades de los elementos materiales y humanos de un organismo social, con el fin de lograr su máxima eficiencia dentro de los planes y objetivos.⁷

La organización se refiere a estructurar cómo deben ser las funciones, jerarquías y actividades. Por idéntica razón, siempre se refiere a funciones, niveles o actividades que están por estructurarse.

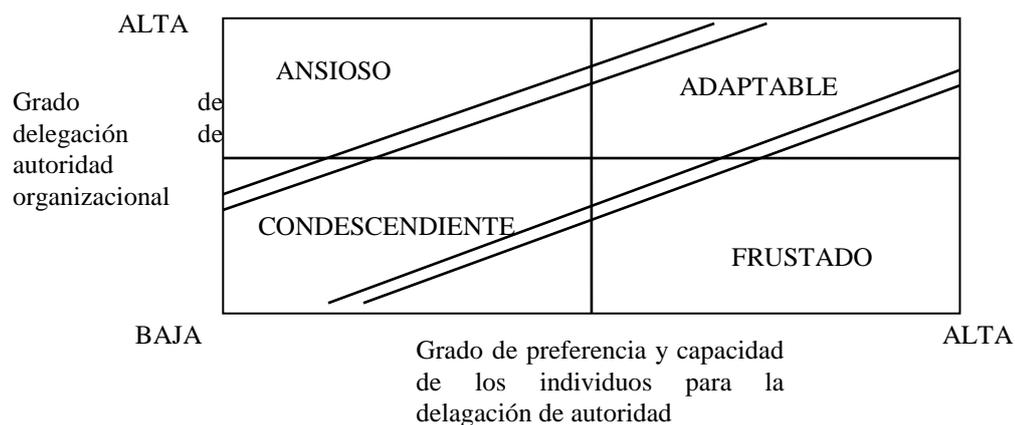
Una estructura de organización establece los medios o conductos a través de los cuales se ejerza autoridad sobre cada una de las unidades de organización. Estos medios son canales definidos de supervisión y pueden considerarse como las líneas formales de autoridad, por las cuales se mueven las instrucciones y las órdenes, se transmiten las comunicaciones formales y los informes sobre operaciones y llegan a todos los niveles de la organización.

Cuando una institución crece, su estructura crece vertical y horizontalmente. Esto es de suma importancia para el dirigente, porque tal crecimiento influye en el

⁷ Gómez Guillermo. *Planeación y Organización de Empresas*. Pg. 191

número de unidades de organización que tienen que ser coordinadas, afecta la comunicación entre ellas, da origen a relaciones muy complicadas y, además influye en los costos.

El crecimiento vertical de la estructura de una organización es consecuencia de la delegación de autoridad hacia los niveles inferiores. Son características del crecimiento vertical de organización las relaciones de superior a subalterno, es decir, que aquél delega autoridad a éste, quien a su vez delega autoridad en otro y así sucesivamente, hasta formar una línea que va de la cima al fondo de la estructura orgánica. La delegación de autoridad, se genera muchas veces debido a los sistemas administrativos y procedimientos sobre los que debe trabajar una organización. La figura 3 muestra diversos estados de ánimo del personal de acuerdo al grado de autoridad que se le haya asignado. Es preferible, que en toda organización exista estados condescendientes y adaptables, debido a que estas características permiten una mejor labor; dependerá de la circunstancia y de los procesos para determinar que grado de autoridad deberá tener los miembros de la organización.



La esencia de la Reingeniería de Procesos. Peppard. PG 112

FIGURA 3 Matriz de delegación de autoridad según Clutterbuck, Clark y Armistead

La organización tiene tres objetivos, que pueden ser intercalados o ser independientes entre sí: el crecimiento, la estabilidad y la interacción. El último se refiere a las organizaciones que existen principalmente para proveer un medio para la asociación de sus miembros con otros.

Las responsabilidades de organización, han sido observadas por un gran número de personas y se desarrolló un campo de pensamiento e investigación llamado teoría de sistemas, dedicado a la tarea de descubrir factores universales de organización.

El propósito de una teoría general de sistemas es crear una conciencia de los factores universales de organización o, en otras palabras, una ciencia universal, usando elementos de organización comunes, encontrados en todos los sistemas como punto de partida.⁸

La teoría moderna de organización se encuentra dentro de la periferia de la teoría general de sistemas. Tanto la teoría general de sistemas como la teoría moderna de organización estudian:

1. Las partes (individuos) en masa y los movimientos de los individuos hacia el sistema y fuera del mismo.
2. La interacción de individuos con el ambiente que se encuentra dentro del sistema.
3. Las interacciones entre los individuos dentro del sistema.
4. Los problemas del crecimiento general y estabilidad de los sistemas.

La teoría moderna de organización y la teoría general de sistemas son similares en cuanto a que ambas consideran la organización como un total integral. Difieren, sin embargo, en término de sus generalidades. La teoría general de sistemas es

⁸ Gómez Guillermo. *Planeación y Organización de Empresas*. Pg. 210

concerniente a todos los niveles del sistema, mientras la teoría moderna de organización se enfoca principalmente sobre la organización humana.

1.4.2. RESPONSABILIDADES DE DIRECTIVOS

La autoridad es el derecho de actuar o de exigir acción a otros, dentro de un área prescrita. El concepto de autoridad está asociado con la facultad de tomar decisiones y de ver que éstas se cumplan. El aspecto de obediencia a la autoridad se la debería obtener por medio de persuasión y solicitudes. En una minoría se emplearía la coerción o la fuerza.

La autoridad tiene limitaciones definidas. Debe usarse de acuerdo con los esfuerzos para lograr los objetivos aceptados de la unidad organizacional. No la utiliza un gerente de oficina según su capricho o según lo puedan sugerir sus deseos. El uso de la autoridad también está influido por las personas con quienes va a aplicarse. La exigencia de ciertas acciones a otros debe estar dentro de su capacidad para desempeñar ciertas tareas.

La relación establecida por la autoridad es vertical u horizontal. Las relaciones de autoridad vertical son las que se encuentran en distintos niveles organizacionales y se refieren a la asociación superior-subordinado. Las relaciones de autoridad horizontal se refieren a las unidades organizacionales en un nivel de la organización y corresponden a la asociación gerente a gerente del mismo nivel organizacional.

Por último, la autoridad es dinámica. Su formato cambia de acuerdo a condiciones y requisitos específicos de la persona o del grupo. No siempre se aplica en la misma forma o en el mismo grado.

Cuando se le da autoridad a un miembro de la administración o la asume para el desempeño de un trabajo específico, se crea la obligación de ejecutar el trabajo. La

aceptación de esta obligación se conoce como responsabilidad, la que puede ser definida como sigue: “La responsabilidad es la obligación para el cumplimiento de un deber y del cual uno responde en la asignación de una tarea señalada”.⁹

Por lo general la responsabilidad toma la forma de una lista de obligaciones. Entre esta lista podríamos mencionar:

- ✍ Realizar un continuo estudio del exceso de la capacidad de procesamiento de la oficina.
- ✍ Revisar cuales son las áreas de trabajo de oficina que requieren la mayoría de su tiempo.
- ✍ Investigar los últimos desarrollos en máquinas de oficina que pudieran usarse en su unidad organizacional.
- ✍ Revisar los procedimientos administrativos.
- ✍ Revisar reportes y desechar documentos inútiles, etc.

Tanto la autoridad como la responsabilidad deben ser definidas y conocidas por todos los interesados. La definición de la autoridad y de la responsabilidad ayuda a lograr la coordinación necesaria entre los varios esfuerzos que intervienen.

La autoridad de cualquier gerente debe ser coigual a la responsabilidad asignada y viceversa. La asociación de la autoridad y la responsabilidad es estrecha y existe una donde existe la otra.

1.4.3. LA PLANEACION Y EL AMBIENTE ADMINISTRATIVO

Como se ha mencionado anteriormente, la planeación trae consigo una serie de resistencias, temores, etc. Por lo general, el ambiente administrativo es importante porque está relacionado a la forma en que las personas sienten hacia las prácticas

⁹ Terry George. *Administración y Control de Oficinas*. Pg. 469

administrativas que se aplican en las organizaciones. El ambiente influye en todo el proceso del comportamiento que conduce a la satisfacción en el trabajo. Es indispensable que exista una comunicación mucho más fluida desde arriba hacia abajo y viceversa, de esta manera se mantendrá siempre en conocimiento sobre las decisiones que a futuro se implementarán.

En la toma de decisiones, quien las toma, siempre corre el riesgo de resistencia. En este aspecto, podríamos valernos del uso de los comités que se analizarán mas adelante.

1.4.3.1. ANALISIS DE FLUJO DE TRABAJO

Martin describe el análisis del área de negocio de la siguiente manera:

“El análisis de las áreas de negocio (AAN) establece un marco detallado para construir una empresa basada en la información. Estudia un área de negocio cada vez y la analiza con detalle. Utiliza diagramas y matrices para modelar y registrar los datos y actividades de la empresa, y para dar una clara visión de las complicadas y sutiles maneras en que se relacionan los diferentes aspectos de la información de la empresa”.¹⁰

Para establecer este marco detallado, el analista debe empezar a investigar aspectos de contexto libre, relacionado al proceso que se quiere diseñar. Esto conlleva a un entendimiento básico del problema y del proceso; luego, seguir con un conjunto de interrogantes que permitan obtener un conocimiento más detallado del proceso y además obtener posibles soluciones.

En este análisis el flujo de la información representa cómo cambian los datos y el control a medida que se mueven dentro de un sistema. A lo largo de todo el proceso

¹⁰ Pressman Roger. *Ingeniería de Software Un Enfoque Práctico*. Pg. 166

de cambio, se pueden ir introduciendo información adicional que ayudan a obtener el resultado deseado.

1.4.3.2. DISEÑO TECNICO

Para el diseño técnico, el analista debe evaluar los principios técnicos del sistema. Debe realizar una valoración de las tecnologías que se requieren para desarrollar el sistema, métodos, procesos, etc., además del riesgo que implicará el optar por una metodología de desarrollo.

Se debe realizar un estudio de función, rendimiento, restricciones, tecnologías y alternativas que puedan afectar a la consecución de un sistema aceptable.

1.4.3.3. DISEÑO SOCIAL

Los organigramas y relaciones existentes en las organizaciones no pueden cambiarse fácilmente. Sin embargo, un proyecto de software, al involucrar personal de toda la organización, obliga; hasta cierto punto, a cambiar los esquemas de procesamiento. Además, toda implementación informática exige cambios e innovaciones inmediatas. Por ejemplo, la redefinición de flujos de información, establecimiento de comunicación y relaciones horizontales y verticales más reducidas, etc.

Los equipos de trabajo de desarrollo de software, deberán tener cierto grado de autoridad e independencia, cuanto así lo deseen los directivos de la organización; siempre enmarcados en los objetivos y plazos impuestos institucionalmente.

Estos equipos se pueden establecer sobre la base de:

- ✓ La dificultad del problema que hay que resolver.
- ✓ El tamaño del programa.

- ✓ El tiempo requerido para la entrega del software y la rigidez de la fecha de entrega.
- ✓ La presión, urgencia y/o importancia del área por resolver.
- ✓ El grado en que el problema puede ser automatizado; etc.

En el contexto de toda la organización o del área involucrada en el desarrollo del software, debe existir un alto grado de comunicación entre el equipo de desarrollo y el personal que conoce del proceso. Por lo general, esta comunicación algunas veces no es de lo más cordial, debido al temor que nace en las personas por creer que serán desplazadas de sus puestos de trabajo.

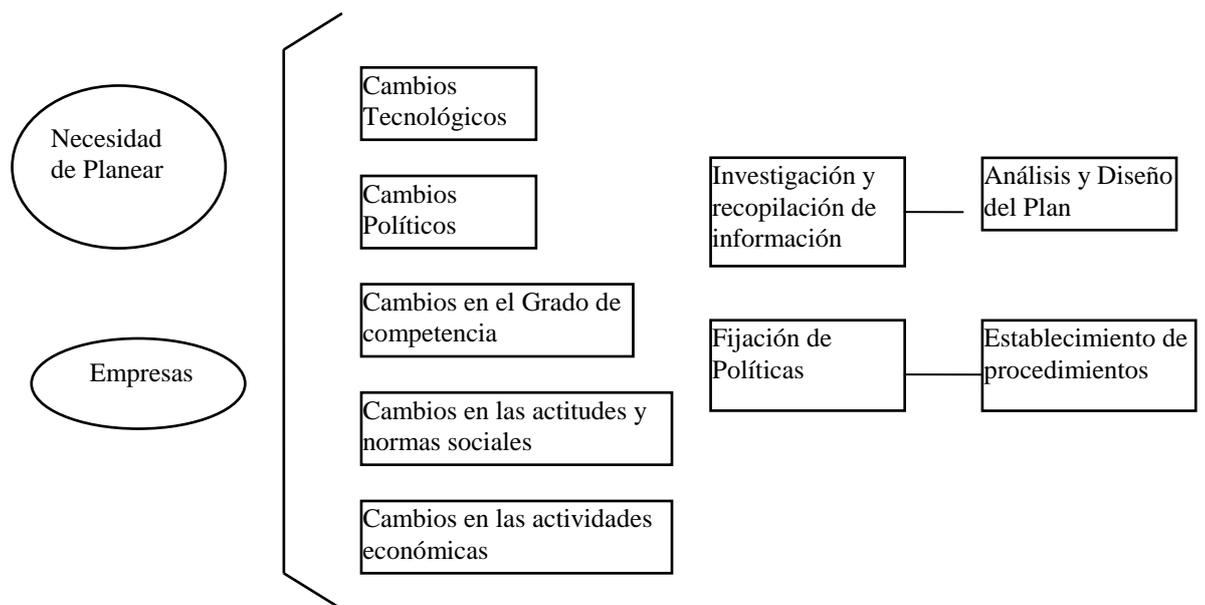
1.4.4. ESTUDIO DE PROYECCIONES INFORMATICAS

Tanto del análisis del flujo de trabajo, como de los diseños técnico y social, se puede tener ya una visión mucho más clara de cual debe ser el camino a seguir en la planeación y desarrollo informático.

Es así que, las proyecciones deben estar basadas, no solo en la misión estratégica de la organización, sino también en la misión que cada área institucional tiene definida para aportar a la misión organizacional.

1.5. PLANEACION DE RECURSOS

Una de las tareas de la planificación del desarrollo informático es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de software, dentro de los recursos tenemos a las Herramientas (hardware y Software), que son la base que proporciona la infraestructura de soporte al esfuerzo de desarrollo, luego tenemos a los componentes reutilizables, también el recurso primario, las personas (el recurso humano); además, como un elemento integrador al presupuesto económico.



Planeación y Organización de Empresas, Gómez Ceja Guillermo 8va. Edición 1996, México PG. 8

FIGURA 4 Planeación Estratégica

Al hablar de Planeación de Recursos, esto tiene que ver con todo el esquema de la teoría general de sistemas. Debido a esto, podemos apreciar en la Figura 4, que la

Planeación Estratégica debe incorporar las consideraciones tecnológicas, políticas, competencia ó el modo de desenvolverse otras organizaciones del mismo tipo o área, el aspecto social-económico tanto de los miembros de la organización como del entorno, etc.. Todas estas condicionantes de una Planeación deben estar sustentadas por una adecuada investigación del medio donde se va a desarrollar, diseñar un plan que conduzcan a establecer políticas internas de cada organización de acuerdo a como sus miembros ven al entorno y por último el establecer procedimientos que deban seguir dentro de la organización.

1.5.1. PLANEACION DEL EQUIPO

1.5.1.1. ESTRATEGIAS DE SELECCION DE EQUIPO

Se puede elegir de entre una amplia gama de medios manuales, una mecanización completa o muy elevada, o establecer combinaciones de medios manuales y mecánicos. Es importante recordar que no todas las operaciones manuales son menos eficaces y más costosas que los medios mecánicos, y que no todos los problemas de información se solucionan con sólo instalar máquinas sofisticadas, y que no todas las empresas tienen necesidades de información que justifique sistemas de información muy mecanizados.¹¹

Sin embargo, en la actualidad, por lo general es necesario cierto grado de mecanización y/o automatización.

¹¹ Terry George. *Administración y Control de Oficinas*. Pg 197

La selección de máquinas requiere tener conocimiento de que existan en el mercado y elegir las mejores unidades de acuerdo con las necesidades particulares fijadas por la planeación del sistema.

Para poder decidir sobre la utilización de las máquinas, se necesita saber al menos: los tipos de máquinas y equipo existentes, las características de cada una de ellas, el costo inicial, el costo del mantenimiento.

Para obtener este conocimiento se puede apoyar en: consultar con los representantes de los fabricantes de estas máquinas, asistir a los cursos, seminarios y/o conferencias ofrecidos por ellos, informarse en la literatura existente sobre el tema y en pláticas con las instituciones que tengan dichas máquinas en operación, acudir a consultores expertos en el tema. Pero siempre será esencial la participación del personal de la institución, ya que les proporcionará percepciones prácticas de lo que sucede y por qué. Además, se familiarizarán con el procesamiento propuesto así como con los antecedentes necesarios para la instalación y operación exitosas.

Es práctica común crear un comité para propósitos del estudio y recomendaciones respecto a cuál máquina deberá usarse así como las razones de porqué.

Entre las principales tareas del comité está la de aclarar cuál es el trabajo de oficina que en realidad va a hacerse. La determinación de qué información y a quién debe ir es esencial.

1.5.2. PLANEACION DE SOFTWARE

Al planear el software se debe tomar en cuenta: software ya desarrollado, software ya desarrollado y que necesite adaptaciones, software por desarrollar.

En el primer caso, se necesita tener la certeza de que el software ya desarrollado cumpla con todas las especificaciones funcionales de la organización sobre la cual se va implementará. Para el segundo, se debe medir cuales son las implicaciones que este tendrá al modificar parte del software ya desarrollado. Además, es importante saber si las modificaciones las realizarán la propia organización considerando el costo y la disponibilidad del código fuente o la capacidad de modularidad que tenga el sistema adquirido; o dichas modificaciones las realizará la compañía proveedora. En el tercer caso, la decisión de desarrollar software se la puede realizar bajo subcontratación o desarrollo directo con personal de la propia organización.

1.5.2.1. REQUISITOS DE LAS HERRAMIENTAS

En las herramientas; sean estas de base, particulares y/o de aplicación, se debe considerar, el costo de los programas, el uso de los equipos, tiempo, personal y operación. Los beneficios que justifiquen el desarrollo o adquisición de un sistema pueden ser el ahorro en los costos de operación, la reducción del tiempo de proceso de un sistema, mayor exactitud, mejor servicio, una mejoría en los procedimientos de control, mayor confiabilidad y seguridad, además de que el cliente en realidad haga un uso adecuado del sistema. Las formas de utilización de los sistemas, el costo y los beneficios que reportará el sistema, el efecto que producirá en quienes lo usarán y el efecto que éstos tendrán sobre el sistema y la congruencia de los diferentes sistemas.

1.5.2.2. BENEFICIOS QUE DEBEN BRINDAR LAS HERRAMIENTAS

Las herramientas son esenciales para el análisis de sistemas. Ellas mejoran la forma en que ocurre el desarrollo y tienen influencia sobre la calidad del resultado final.

Las herramientas extienden en tres formas la capacidad del análisis de sistemas: proporcionan el potencial para mejorar la productividad del análisis, facilitan el

desarrollo de procesos más eficaces y mejoran la calidad del sistema. En otras palabras, tanto el proceso de desarrollo de sistemas como el producto que se obtiene con él, pueden mejorarse con el uso de herramientas apropiadas.

Con las herramientas correctas, el equipo de desarrollo tiene el potencial de ser más productivo; se pueden completar las mismas actividades de desarrollo en un tiempo menor que el que se necesita cuando no se utilizan las herramientas. Aumentan la productividad al disminuir la cantidad de tiempo necesaria para documentar, analizar y desarrollar sistemas de información.

La disponibilidad de herramientas para el flujo de datos, estimula al analista a poner mayor hincapié, antes de iniciar el desarrollo del sistema, sobre la determinación de los requerimientos de sistemas.

Identificar los requerimientos del usuario, trasladarlos en una forma comprensible y comunicarlos a todas las partes interesadas, puede ser un proceso de desarrollo más eficiente antes de iniciar con rapidez la codificación de programas.

Cuando las herramientas mejoran los procesos, por lo general también ocurre lo mismo con los resultados. Los usuarios de los sistemas de información desean calidad en el sistema con un tiempo razonable.

Hace algún tiempo no había muchas herramientas; por tanto, no era posible el desarrollo de prototipos de aplicación ni tampoco el análisis estructurado. La invención de los lenguajes de cuarta generación y de diagramas de flujo de datos, dos herramientas esenciales para realizar respectivamente estas tareas, cambiaron en las organizaciones los procedimientos para analizar sistemas.

Las herramientas de análisis asistido por computadora mejoran la velocidad y disminuyen el tiempo necesario para completar la tarea de desarrollo. La automatización también se hace cargo de algunas tareas que son pesadas. El

desarrollo de diagramas de flujo de datos, parte esencial del método de análisis estructurado, es una tarea que puede consumir mucho tiempo. El dibujo de diagramas puede convertirse en algo tedioso y quizá sea necesario revisar varias veces los diagramas. Las herramientas automatizadas para flujo de datos, hacen posible dejar al software de la computadora el proceso de dibujo. La consistencia que pueden ofrecer los procedimientos es una excelente razón para ampliar el conjunto de herramientas asistidas por computadora para el desarrollo de sistemas.

De manera similar, la generación de código por computadora es una tarea que realizan mejor las computadoras que las propias personas, ya que las reglas de generación se pueden aplicar en forma consistente y exacta.

Los proyectos de desarrollo de sistemas de información dependen de la captura y análisis de los detalles que describen una situación real, los requerimientos de una aplicación y las especificaciones de diseño. Estos datos quizá pertenezcan a una determinada aplicación o a todos los sistemas utilizados en una organización.

Una ventaja que distingue a muchos sistemas automatizados es la captura, almacenamiento, procesamiento y recuperación de los detalles de un sistema. Una vez en forma procesable por la computadora, los detalles del sistema pueden utilizarse para muchas finalidades.

1.5.2.3. PROCESO DE DESARROLLO DEL SOFTWARE

El proceso del software ha sido el foco de atención de la última década. Un proceso de software define el enfoque que se toma cuando el software es tratado por la ingeniería, pero la tecnología del software también acompaña a las tecnologías que acompañan el proceso (métodos, técnicas y herramientas automatizadas).

El proceso del software debe tener muy en cuenta:

- ↳ Gestión de recursos del proyecto
- ↳ Aspectos técnicos de la calidad del software
- ↳ Necesidad de la satisfacción del cliente
- ↳ Entrega oportuna del producto
- ↳ Importancia de mediciones y métricas
- ↳ Control de calidad de software

1.5.2.4. EVOLUCION DE LAS TECNOLOGIAS DE DESARROLLO

La tecnología CASE no puede considerarse realmente como una nueva generación de la tecnología de software, más bien es una combinación e integración de técnicas y herramientas de la tercera, cuarta y quinta generaciones.

Las tecnologías de desarrollo, han ido evolucionando conforme a las necesidades humanas y avance de las computadoras. Una tecnología de desarrollo de software no puede estar aislada de la tecnología de hardware, debido a que ambas son mutuamente complementarias.

Así, podemos manifestar que la primera generación estaba orientada a las máquinas, a través de los lenguajes de máquina. En la segunda generación, se orienta a las necesidades de cada usuario en combinación con las capacidades de las máquinas mediante los lenguajes ensambladores. Al ser esta una limitante en el desarrollo del software, que en gran medida dio origen a “la crisis del software”, el desarrollo se orienta en la tercera generación al usuario y da avances en el análisis de datos; en esta generación se hace uso de los lenguajes de programación de alto nivel. Para la cuarta generación, los lenguajes de aplicación establecen a la información y datos como un recurso más, dentro de las operaciones de la organización, estableciendo a éste como un recurso estratégico y que su tratamiento y procesamiento debe estar vinculado íntimamente con la estrategia global de la institución. Al haberse

desarrollado ya toda una vasta gama de teorías de tratamiento de la información, se hacen muy populares las aplicaciones abiertas, en donde a parte de los requerimientos del usuario, del análisis de datos y otros, se establecen sistemas que puedan ser aplicados a una serie de actividades similares.

Para la quinta generación, aún en desarrollo, se hace uso de conceptos de inteligencia artificial, redes neuronales artificiales, lenguaje natural. Algunas aplicaciones poseen ya ciertos elementos inteligentes.

1.5.2.5. LA INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADOR

Las herramientas de diseño para automatizar varias de las tareas del ciclo de vida se centran en el análisis del software y en las tareas de diseño. Con ayuda de estas herramientas, los profesionales del desarrollo pueden crear sistemas de software interactivos. Esta nueva tecnología es la llamada Ingeniería de Software Asistida por Ordenador.

Definida de una forma simple, la ingeniería de software asistida por ordenador es la automatización del desarrollo del software, comúnmente referida como tecnología CASE.

1.5.2.6. HERRAMIENTAS CASE EN EL DESARROLLO DEL SOFTWARE

Según el Dr. Eduardo Rivera Porto el término "ingeniería del software" fue expuesto por primera vez en el 1968 en una conferencia de la OTAN. En esa conferencia se reveló la existencia de una "crisis de software". Eso reconocía la problemática de desarrollo de software y se aspiraba entonces a proponer y encontrar soluciones.

En aquel momento muchos de los expertos identificaron incorrectamente, que la producción del código era la razón primordial de dichos problemas. Debido a la

falta de éxito, a finales de los setenta comenzó un cambio en el modo de atacar los problemas de especificaciones, diseño y administración. Este período introdujo una variedad de técnicas que enfocaban las fases iniciales del ciclo de vida del desarrollo del software.

La idea básica del CASE es la de apoyar cada fase del ciclo de vida con un conjunto de herramientas que ahorren tiempo y trabajo. Algunas herramientas CASE se concentran en apoyar fases iniciales del ciclo de vida. Estas proveen asistencia automatizada de manera de dibujo automatizado de diagramas, pintar pantallas y correcciones. Otras enfocan las fases de implementación del ciclo de vida. Estas incluyen codificación automatizada y generadores de prueba. En algunos casos, estas herramientas son utilizadas en conjunto de lenguajes de tercera y cuarta generación.

Algunas actividades típicas dentro de las fases son: estimado de recursos necesarios, dibujo del documento de especificaciones, codificación, pruebas de integración y la redacción del manual para los usuarios.

A principios de los setenta se introdujo la técnica de gráficas tales como Hippo, Nasic Schneiderman y flujogramas estructurados. La llegada de diseño de flujo gramas y análisis estructurado trajo el concepto de diccionarios de datos, el cual contenía detalles de cada tipo de datos y otro tipo de información pertinente al sistema. Eventualmente todas esta herramientas gráficas fueron integradas con bases de datos de diccionarios para producir diseños y herramientas de desarrollo muy robustas.

La historia de herramientas CASE para el desarrollo de software se puede resumir de la siguiente forma:

Principio de los 80's

- ⌘ Documentación Automatizada.
- ⌘ Diagramas Computarizados.

	⌚ Herramientas de Análisis y Diseño.
Mitad de los 80's	➤ Análisis Automático de Diseño y Verificación. ➤ Banco Automático de Información del Sistema.
Finales de los 80's	⌚ Generación Automática de Código. ⌚ Automatización de Diseño de Enlace.
Principio de los 90's	➤ Dispositivo de Metodología Inteligente. ➤ Interfases Reusables como Metodología de Desarrollo.
Tendencias Futuras del Case	⌚ La utilización de la tecnología multimedia. ⌚ La incorporación de técnicas de inteligencia artificial. ⌚ Sistemas de realidad virtual.

Una de las primeras herramientas fue el sistema operativo UNIX. Este proporcionó un tipo de cimiento sólido en el cual otras herramientas podían empotrarse y ofrecer así facilidades útiles para el manejo de programas.

1.5.3. PLANEACION DE LA INFORMACIÓN

Nuestro recurso más grande es la información. No sólo es esencial en la vida moderna, sino también en la administración y ejecución de la mayoría de las actividades de toda empresa. Es universal porque existe en todas partes.

Puesto que el papel suele ser el medio más utilizado para transmitir la información, el término "papeleo" es de uso común para identificar los esfuerzos de información, aún cuando estén implicados otros medios, tales como microfilmes, cintas, discos, etc.

La información se refiere a datos que tienen significado: palabras, cifras o símbolos; y que transfieren conocimientos utilizables.¹²

¹² Terry George. *Administración y Control de Oficinas*. Pg. 19

Los datos representan la materia prima que, por medio del procesamiento, se convierte en información. El procesamiento, tal como la lectura, la escritura, el cálculo, la selección, coloca los datos en forma de relaciones que revelan ciertos significados pertinentes al evento al cual se aplican. Así mismo debe observarse que no todos los datos representan información y que, desde el punto de vista del usuario, cierta información tiene más valor que otra.

La información se ha incrementado con mucha rapidez y, junto con este auge, la necesidad de más y mejor administración de ella. Debido al crecimiento y complejidad de nuestra sociedad y del entorno internacional y de las empresas e instituciones.

El problema de utilizar la información con efectividad se centra principalmente alrededor de asuntos del tipo y del formato en que deba llegar la información a quien la necesita.

La base para una decisión correcta es una información correcta, precisa y oportuna. El utilizar una mala información para la toma de decisiones puede conducir a grandes dificultades. Pero el proporcionar una información adecuada no constituye un fin en sí mismo, más bien es un dispositivo por medio del cual los encargados de las decisiones reciben información que les ayuda a resolver sus problemas.

Por otra parte, el proporcionar más información de la necesaria, se deriva de la creencia de que una mayor cantidad de información es conveniente y es bueno que el gerente la conozca. Inevitablemente esto da como resultado el proporcionar información que no es esencial.

1.5.3.1. JERARQUIZACION DE LA INFORMACIÓN

La información es heterogénea. Se emplean diversos tipos para diversos usos y se debe considerar cada uno de ellas con sus propias características.

- └ Interna y externa: refiriéndose el calificativo a la fuente de la información generada.
- └ Repetitiva y no repetitiva: La primera se genera continuamente; la última incluye estudios especiales.
- └ Compulsoria y operacional: La información compulsoria es la que requiere el gobierno y otras agencias externas; la información operacional es empleada por los directivos para planear y llevar a cabo sus operaciones.
- └ Activa y pasiva: La activa sirve para quien la recibe a emprender una determinada acción; y la pasiva, que no implica acción posterior, tal como las notificaciones.
- └ Pasada y la futura: La planeación requiere información y proyección del futuro con base en la información del pasado.

Así tenemos que mientras mayor sea el número de decisiones administrativas en los niveles superiores mayor será la importancia de la información externa y futura. La información no repetitiva es de poca utilidad para la automatización. En contraste, la información repetitiva y activa ofrece por lo general buenas posibilidades para automatización.

La información es dispersa; existe en todos los departamentos de una empresa. Es dispersa porque la necesitan las personas de toda la organización así como por ciertas otras organizaciones externas

1.5.3.2. DIVISION DE LA INFORMACION POR AREAS

Además de las corrientes verticales intra departamentales, la estructura organizativa cualquiera que sea, produce corrientes de información horizontales y oblicuas extra departamentales.

Los flujos de información entre los grupos de una organización son necesarios para su eficiente gestión, siempre y cuando tales corrientes no distorsionen el propio organigrama.

En ocasiones, las organizaciones crean espontáneamente canales alternativos de información, sin los cuales las funciones no podrían ejercerse con eficacia; estos canales alternativos se producen porque hay pequeños o grandes fallos en la estructura y en el organigrama que los representa.

Otras veces, la aparición de flujos de información no previstos obedece a afinidades personales o simple comodidad. Estos flujos de información son indeseables y producen graves perturbaciones en la organización.

1.5.3.3. PLANEACION ESTRATEGICA DE LA INFORMACIÓN

“El primer paso de la ingeniería de la información es la Planificación de la Estrategia de la Información. Los objetivos generales del PEI son (1) definir los objetivos y metas del negocio que sean estratégicos, (2) aislar los factores de éxito críticos que permitirán al negocio esos objetivos y metas, (3) analizar el impacto de la tecnología y automatización en las metas y objetivos y (4) analizar la información existente para determinar su papel en la consecución de las metas y objetivos”.¹³

La información debe ser considerada como información operacional, la cual usamos para desarrollar nuestro trabajo y obtener los resultados deseados. Además, existe la información administrativa que se genera inevitablemente, debido a la estructura organizacional y relaciones propias interdepartamental o seccionales.

¹³ Pressman Roger. *Ingeniería del Software Un Enfoque Práctico*. Pg. 163

Estos dos tipos de información deben ser tratados por igual. La operacional puede ser la más fácil de identificar, debido a que es la materia prima con la cual los procesos obtendrán sus resultados. La información administrativa, puede ser la que genera más gastos organizacionales, debido a que es generada de acuerdo a las necesidades de los propios empleados. No está contemplado dentro del flujo de la información y de procesos, pero es indispensable debido a los canales de comunicación.

1.5.3.4. FUNDAMENTOS DE LAS HERRAMIENTAS CASE BASADAS EN LA INGENIERIA DE LA INFORMACIÓN

En las metodologías de desarrollo de sistemas de información centrada en la información, un modelo de dato lógico que represente la información utilizada a través de una organización, es el punto de arranque de todo el desarrollo del sistema. El proceso empieza con un análisis de alto nivel de una organización, sus objetivos comerciales y las necesidades de la información estratégica. Una visión global de las necesidades de información se representa en un modelo que muestra todas las entidades de datos básicos de la organización y sus relaciones entre sí. Después, basándose en este modelo se construyen individualmente los sistemas de información de la organización. Así los procedimientos se derivan de los datos.

La ingeniería de la información es tanto un enfoque comercial de ingeniería como una formulación para la construcción de sistemas de software. Es una formulación comercial porque comienza con una “planificación estratégica de la organización”. Es una formulación de ingeniería porque proporciona un procedimiento paso a paso para construir sistemas de información.¹⁴

¹⁴ McClure Carma. *Case La Automatización del Software*. Pg. 119

La ingeniería de la información es una disciplina más reciente que la ingeniería de software; tiene un enfoque más amplio que la ingeniería de software, comenzando por un nivel más alto: con la planificación estratégica. Sin embargo, en la fase de diseño del programa, su formulación del desarrollo del programa es básicamente la misma que en la ingeniería de software.

La ingeniería de la información tiene un campo de acción más estrecho que la ingeniería de software. Se emplea aquella para construir sistemas de información, mientras que la ingeniería de software se emplea para desarrollar toda clase de sistemas, tanto de tiempo real como comerciales.

La ingeniería de la información es un enfoque orientado a la información para el desarrollo de software y por tanto difiere de los orientados a los datos, como las de Jackson y de Warnier-Orr. La ingeniería de la información ha sido diseñada para desarrollar sistemas de bases de datos y puede trabajar con estructuras de datos no jerarquizadas. El modelado lógico de datos y la normalización son pasos requeridos en la ingeniería de la información. La ingeniería de la información construye sistemas de información integrados, porque están contruidos sobre el mismo modelo lógico de datos.

Además el énfasis de la ingeniería de la información está en los datos desde el punto de vista de los requerimientos organizacionales y no en los requerimientos del sistema de software. Todo el enfoque del desarrollo de la ingeniería de la información se centra en la visión lógica de cómo el sistema utiliza los datos.

Las premisas básicas de la ingeniería de la información son:

- ✓ Se necesita una estrategia general de desarrollo del sistema centrada en una planificación de los objetivos estratégicos de la organización para construir sistemas que satisfagan mejor las necesidades de la corporación.

- ✓ Sistemas de información que puedan integrarse mejor si los datos a compartir se controlan centralmente por ser parte de un mismo modelo lógico de datos.
- ✓ La representación lógica de datos es relativamente estable, en tanto que los procedimientos que utilizan los datos cambian frecuentemente. Por tanto, el modelo lógico de datos, que refleja lo que es la organización, no como trabaja, deberá ser la base del desarrollo del sistema.

1.5.3.5. EL DICCIONARIO DE DATOS DE LA HERRAMIENTA CASE

El Diccionario de datos es la descripción lógica de los datos para el usuario. Reúne la información sobre los datos almacenados en una base de datos (descripción, significado, estructura, consideraciones de seguridad y uso de aplicaciones, etc.).

Es una referencia de datos acerca de los datos recopilados para guiarse durante el análisis y el diseño; recopila, coordina y confirma lo que un término específico significa para la gente de la organización.

Un diccionario de datos base, puede estar compuesto por:

-  El nombre y el sinónimo (alias) del dato.
-  Descripción del contenido del dato.
-  Los datos elementales que se relacionan con el término.
-  El rango permitido del dato.
-  La longitud disponible en caracteres.
-  Información adicional

Además de esta información se puede incluir más elementos para que este sea más robusto. Sin embargo, no se debe caer en la redundancia ni en la inconsistencia, todo

dependerá del grado de especificidad y particularidad que el equipo de desarrollo desee dar al diccionario.

1.5.4. PLANEACION PRESUPUESTARIA

El presupuesto es un plan de acción financiera para un programa que cubre un periodo definido. J. L. Dohr dice: "el presupuesto constituye esencialmente un instrumento de control financiero y su preparación requiere minuciosas estimaciones y cálculos sobre supuestas ventas, producción total, costos de distribución, de administración y financieros".¹⁵

El propósito fundamental de un presupuesto es coordinar las actividades de las diversas unidades administrativas que intervienen en un proyecto, ayudar a que la administración ejerza el control sobre las diferentes partes del programa y descubrir cuál es el camino más productivo por el que puedan encauzarse los esfuerzos del programa.

1.5.5. PLANEACION DE LOS RECURSOS HUMANOS

La cantidad de personas, requeridas para el desarrollo de un proyecto de software, solo puede ser determinada después de hacer una estimación del esfuerzo de desarrollo (por ejemplo personas mes o personas años) y seleccionar la posición dentro de la organización y la especialidad que desempeñara cada profesional.

1.5.5.1. DEPARTAMENTO DE INFORMATICA EN INSTITUCIONES

La informática hoy, está sumida en la gestión integral de la organización, y por eso las normas y estándares propiamente informáticos deben estar sometidos a las normas generales de la misma. En consecuencia, las organizaciones informáticas forman parte de lo que se ha denominado el "management" o gestión de la empresa.

¹⁵ Gómez Guillermo. *Planeación y Organización de Empresas*. Pg. 34

Cabe aclarar que la Informática no gestiona propiamente la empresa, ayuda a la toma de decisiones, pero no decide por sí misma.

1.5.5.2. PERSONAL DE TODA LA ORGANIZACIÓN

Los recursos humanos son un punto clave dentro de una organización, es por eso que, se necesita conocer los tipos de procesos a llevarse a cabo para la contratación de personal en la empresa, incluyendo el reclutamiento, selección, motivación, etc.; debido a que estos manejarán la información, sistemas, flujos, procesos para los cuales se les haya contratado.

Otro factor es que, al ser usuarios finales de sistemas de información, deben ser cada vez más técnicamente instruidos y autosuficientes, al final esto produce que ellos mismos generen sus propias herramientas que ayudan a la operación diaria de una organización.

1.5.6. CONSIDERACIONES ADMINISTRATIVAS Y TECNICAS

Sin la computadora, no tendríamos muchos de nuestros adelantos y comodidades actuales.

La fuerza motriz detrás del progreso de las computadoras tiene un solo propósito: Proporcionarnos la forma de aumentar nuestra productividad. Las computadoras logran esto en muchas formas, pero de especial interés aquí es que las computadoras proporcionan rápidamente la información necesaria para facilitar las operaciones de las organizaciones. Proporcionan la información procesada y refinada para facilitar la toma de decisiones y así servir como uno de los principales adjuntos administrativos. Las computadoras fomentan la planeación, organización, ejecución y control en forma cuidadosa, incrementan las funciones del usuario, no las sustituyen.

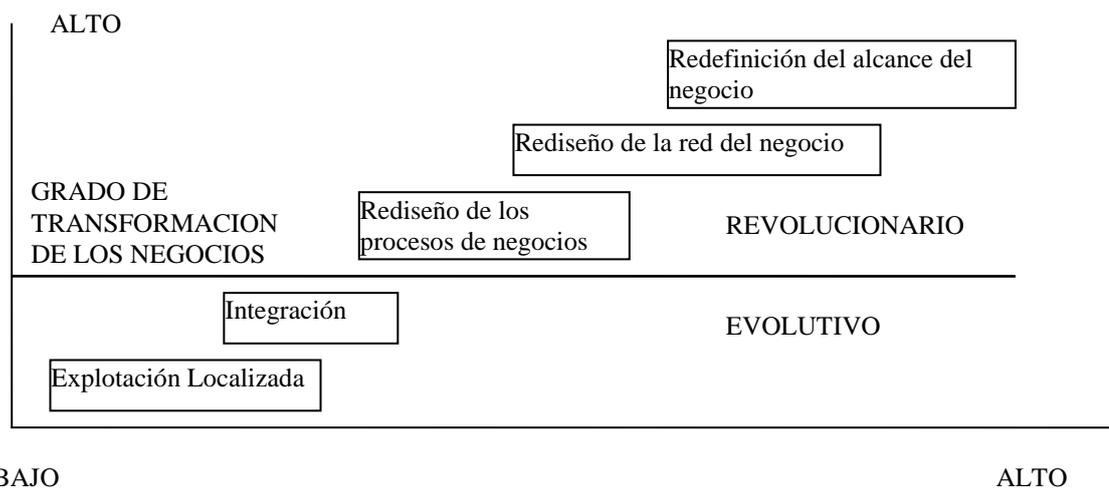
Es erróneo pensar que al colocar una computadora moderna en medio de políticas administrativas ineficaces, prácticas antiguas y una mala organización, se obtengan grandes logros. Utilizadas con propiedad, aumentan el poder e influencia de la mente humana, sin minimizar su importancia incrementan las capacidades humanas.

1.6. LA REINGENIERIA EN LA PLANEACION

1.6.1. CONCEPTOS

“Es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras espectaculares en medidas críticas y contemporáneas de rendimiento, tales como costos, calidad, servicio y rapidez”.

“RPE (reingeniería en los procesos de la empresa) es una filosofía de mejora. Busca lograr mejoras graduales en el rendimiento, rediseñando los procesos mediante los cuales una organización opera, maximizando el contenido de valor agregado y minimizando cualquier otra cosa. Este método se puede aplicar a nivel de procesos individuales o a toda la organización.”¹⁶



La reingeniería no sólo es automatizar procesos existentes, sino presentar nuevos procesos que rompan con los actuales, logrando mejorar la forma de hacer las cosas. La Figura 5, nos muestra como las Tecnologías de la Información dieron un aporte para la reestructuración de los negocios (organizaciones). Estas se pueden dividir en dos pasos: el evolutivo que toda organización pasará esta etapa al iniciar sus actividades en el cual debe realizar un análisis del sector de influencia al que desea llegar y buscar un marco de integración con el entorno y luego al incorporar las tecnologías de la información, pasará al ciclo revolucionario, en el cual, se plantearán los rediseños de la organización a fin de lograr una mejor estructuración interna.

La reingeniería en sus inicios se aplicó como sinónimo de otros términos como downsizing, compactness, streamlining, empowerment. Estos significan empequeñecimiento (reducción de la cantidad de empleados de la empresa reduciendo secciones innecesarias), compactación (eliminación de jefes y directores intermedios para acelerar la ejecución de proyectos), agilización (modificación de puestos y organigramas), fortalecimiento (que significa dotar de confianza y decisión a los empleados de la empresa).

1.6.2. PRINCIPIOS DE REINGENIERIA

¹⁶ Peppard y Rowland. *La Esencia de la Reingeniería en los Procesos de*

La reingeniería de procesos tiene aspectos similares al proceso de la ingeniería de la información. La reingeniería de procesos, al igual que la ingeniería de la información, debe producirse de forma descendente, comenzando por la identificación de los objetivos principales de la organización y culminando con especificaciones detalladas de las tareas.

Hammer sugiere siete principios de la reingeniería:¹⁷

- 3 Organizarse en torno a los resultados.
- 3 Hacer que quienes utilicen la salida del proceso lleven a cabo el proceso.
- 3 Incorporar el procesamiento de la información al trabajo.
- 3 Dotar los recursos geográficamente dispersos como si estuviesen descentralizados.
- 3 Enlazar actividades paralelas.
- 3 Poner el punto de decisión en el lugar en el que se efectúa el trabajo.
- 3 Capturar datos una sola vez, en el lugar donde se producen.

1.6.3. PARAMETROS DE LA REINGENIERIA

Es un proceso que incluye tres aspectos:

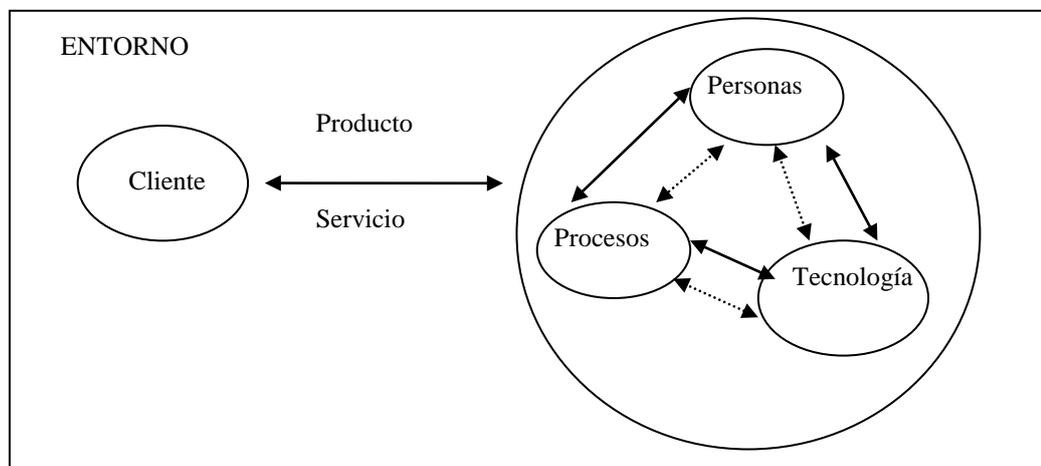
- ↳ Definir objetivos y dividir proyectos en unidades manejables, es decir olvidarse de las metas grandiosas y concentrarse en avances pequeños pero permanentes, que al ser metas alcanzables pueden ser cuantificadas con facilidad.
- ↳ Reformar a la administración intermedia para convencerlos de la necesidad del cambio. Los acuerdos adoptados entre la dirección de la

Negocios. Pg. 20

¹⁷ Roger Pressman. *Ingeniería De Software Un Enfoque Práctico*. Pg. 507

empresa y los empleados "de abajo" fracasan muchas veces porque pasan por encima de los directores, jefes y subjefes, es decir toda aquella plana que arbitra y que realmente dirige lo que se pensó "arriba".

- ↳ Adoptar las tecnologías de información como una herramienta inherente a la empresa. La tecnología de las redes de información en la empresa, permite acelerar la comunicación y disminuye el tiempo usado en trámites y gestiones, permitiendo crear tiempo para concentrarse en la estrategia.



La esencia de la Reingeniería de Procesos. Peppard. PG 45

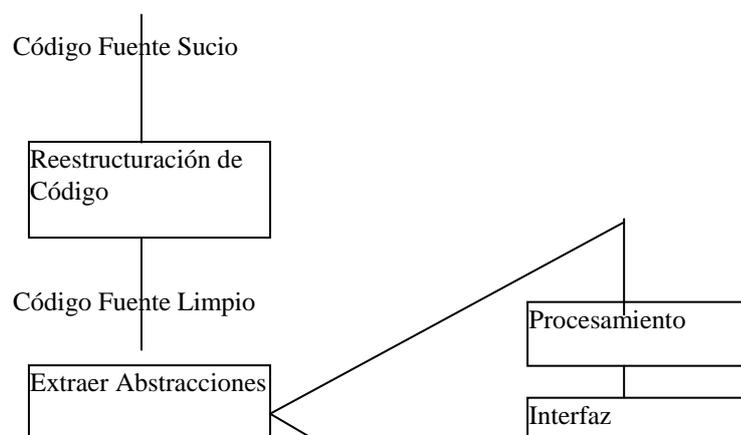
Figura 6 Procesos, personas, tecnología

La Figura 6, nos muestra que toda organización esta conformada por tres aspectos fundamentales: los procesos, las personas, la tecnología. Esta base que forma una organización, estará siempre vinculada con el entorno externo; es decir, siempre se formará para poder satisfacer las necesidades del sector externo, ya sea mediante la fabricación de productos o la entrega de servicios. Vemos que esta interacción dentro de una unidad, determinará las relaciones con el entorno.

1.6.4. DIVERSAS METODOLOGIAS DE REINGENIERIA

La reingeniería puede aplicarse de varias maneras:

- 3 Ingeniería inversa: Proceso de recuperación del diseño de las organizaciones ya existentes.
- 3 Ingeniería progresiva: Utiliza información del sistema ya existente y lo mejora. Vuelve a implementar la funcionalidad del sistema existente y añade más funciones.



La ingeniería inversa, extrae especificaciones de procesos reales, los comprende y redefine su funcionalidad. La Figura 7, nos muestra la forma de la ingeniería inversa aplicada al software. En esta el código sufre una reestructuración para obtener código limpio. El proceso de extraer abstracciones, es considerado como el núcleo de la ingeniería inversa. En esta, se evalúa el código anterior y se extrae la forma de procesamiento, además de la interfaz de usuario y las estructuras de datos que son utilizadas; para a partir de esto, empezar a realizar una redefinición y simplificación del software.

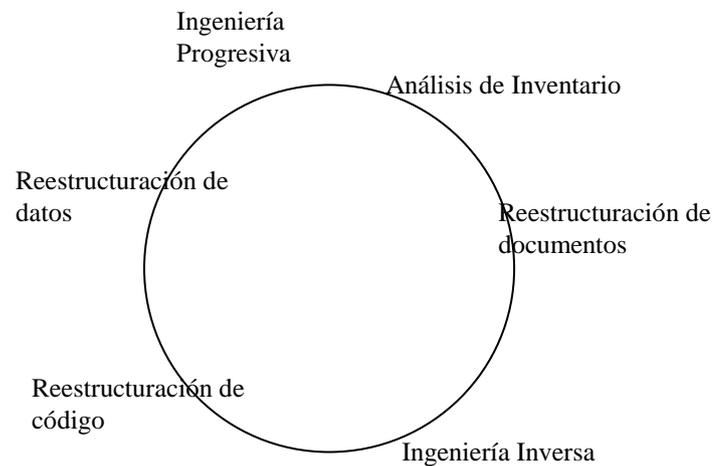
1.6.5. LA REINGENIERIA APLICADA A LA INGENIERIA DE LA INFORMACIÓN

La reingeniería requiere tiempo; consumen cantidades significativas de dinero y absorbe recursos que de otro modo podrían dedicarse a preocupaciones más inmediatas. Por todas estas razones, la reingeniería no se lleva a cabo en unos pocos meses, ni siquiera en unos pocos años. Esta es la razón por la cual toda organización necesita una estrategia pragmática para la reingeniería del software.

Es muy frecuente relacionar la reingeniería de los procesos de negocios con los servicios de información, sin embargo, es necesario aclarar que no se trata de una actividad de computación. Algunas técnicas de reingeniería proceden de la experiencia en el desarrollo de los sistemas de información, mientras que otras surgen de la ingeniería industrial y otras ciencias administrativas. En este sentido, muchos proyectos exitosos calificados como de reingeniería han sido proyectos de sistemas de información en los cuales se ha aplicado, en cierta medida, la reingeniería a los procesos de negocios. Los nexos entre reingeniería y servicios de información pueden llevar a la conclusión errónea de que posicionamiento y reingeniería son metodologías de la tecnología de información cuando, en realidad, son actividades de negocios. Es importante separar los conceptos de computación y de reingeniería; los proyectos de reingeniería deberán ser responsabilidad de los altos mandos y niveles medios y no del departamento de servicios de información de una institución.

Si bien la reingeniería no es un asunto de la tecnología de información, la empresa misma depende en gran parte de las computadoras. Por tanto, la aplicación de esa tecnología con el fin de mejorar la operación, se considerará por lo general en los proyectos de reingeniería, dado que, en la práctica, este tipo de proyectos examina procesos de negocios en los cuales es frecuente descubrir nuevas y mejores aplicaciones para la informática y la tecnología. Además, la reingeniería en particular puede relacionar, en forma directa, el uso de la tecnología con los procesos de negocios. Por ello, es lógico que la tecnología de la información se emplee para contribuir al esfuerzo de reingeniería, más si se tiene en cuenta que este novedoso enfoque hace énfasis en el uso de modelos automatizados de los

procesos de la empresa y en las herramientas automatizadas que apoyan el posicionamiento.



Ingeniería del Software, Pressman PG 511
Figura 8 Reingeniería del Software

El proceso de reingeniería mostrado en la Figura 8, es un proceso cíclico, que en ocasiones, algunos de estos pasos volverán a ser utilizados; además no es un esquema rígido, sino que dependiendo de las condiciones los procesos pueden anteponerse a otros. Previamente, es importante el análisis del inventario, es decir, el qué se posee actualmente y cómo se están haciendo las cosas. La revisión y reestructuración de documentos, nos conducirá a la aplicación de la ingeniería inversa; es decir, a la comprensión y abstracción de los procesos. Con todo esto, la sección de reestructuración son pasos subsecuentes y necesarios para la redefinición de los procesos analizados.

La tecnología de la información es un factor en todos los niveles del modelo de cambio, así el apoyo de la tecnología actual y el diseño de una arquitectura total de la información corporativa se encuentran en el nivel de posicionamiento; los requerimientos para los nuevos sistemas de computación se desarrollan en el de

reingeniería; la compra y la programación de nuevos sistemas en el de infraestructura, y su implementación y aplicación en el de operaciones.

1.6.6. REPROCESAMIENTO DE LA INFORMACION

Una vez que se hayan recolectado las especificaciones de la información necesaria para el diseño del sistema, resulta por demás conveniente, el que, en base a técnicas de reingeniería se realice un estudio y análisis de las mismas. Para lo cual podemos valernos de la Ingeniería de Reversa, de la Ingeniería Directa o Progresiva.

1.7. IMPLEMENTACION INFORMATICA

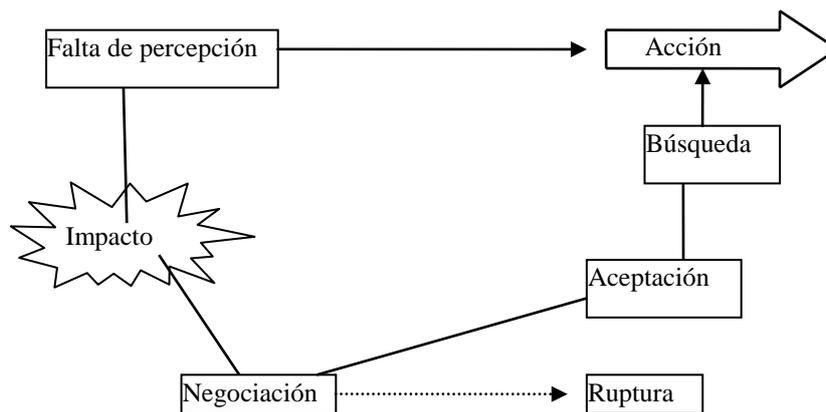
1.7.1. IMPACTO SOBRE LA ORGANIZACIÓN

Los sistemas computacionales tienen implicaciones sociales, en especial en el área de los requisitos para el puesto, de la habilidad del empleado individual y del empleo. Se eliminan muchas de las tareas de oficina manuales y monótonas con el sistema de la computadora que ejecute el laborioso y lento trabajo de oficina. En el aspecto social, es conveniente y es un beneficio para los trabajadores. Muchos creen que han entrado a lo que podría llamarse una Segunda Revolución Industrial, que está sustituyendo a los seres humanos con máquinas en la ejecución de las laboriosas faenas mentales, en la misma forma en que la primera Revolución Industrial sustituyó con máquinas la mayor parte de las fatigosas tareas físicas.

Las opiniones difieren con respecto a los cambios en los requisitos de habilidad como resultado del uso de una computadora. Se enfatiza el entrenamiento y la necesidad de eficiencia en habilidades específicas por el uso de la automatización de la información.

Desde un amplio punto de vista sociológico, el cambio estimulado por las computadoras enfatiza las modificaciones del empleo, las que pueden considerarse que ofrecen mayores o menores oportunidades.

La forma de hacer el trabajo cambiará del manual al mental y del trabajo mental a tareas más desafiantes. La creencia de que se dispondrá de menos oportunidades de trabajo debido a la automatización de la oficina, señala una fuerza de temor dominante. Se han creado nuevas demandas, los empleados han sido desplazados, pero no necesariamente reemplazados. Se puede realizar como práctica administrativa la reubicación de empleados, entrenamiento y capacitación de otros.



La esencia de la Reingeniería de Procesos. Peppard. PG 206

Figura 9 Reacciones al cambio

El ámbito reorganizativo, al realizar implementaciones informáticas, lleva consigo misma, la reestructuración de los procesos, la exigencia del trabajo y del empleado y el impacto sobre toda la estructura que ésta ocasionará. La Figura 9, nos permite ver cómo al personal de la organización, mientras no exista una necesidad de mejorar su ambiente de trabajo no pasarán a evaluar teóricamente el impacto que las modificaciones tendrán. Dentro de este proceso, podrán existir

rupturas, descontentos, etc., hasta que se realice un análisis del impacto y la consecuente aceptación del nuevo estado de trabajo.

Al introducir computadoras en una organización estas cambian la forma de actuar de los empleados. Pero al mismo tiempo debemos considerar algunas prácticas comunes para una mejor utilización de las mismas:

1. Considerar a la computadora como un sistema para procesamiento de datos, no como una simple máquina.
2. Aprender todo lo que sea posible sobre los diversos usos de las computadoras.
3. Capacitar al personal sobre las nuevas terminologías a usar en el tratamiento de la información, automatización y computadoras.
4. No considerar nunca a una computadora como panacea de todos los males de la información corriente.
5. Al planear el uso de la computadora, tómense en cuenta las probables necesidades para el futuro.
6. Con entrenamiento suficiente, utilícese el personal actual para la operación de la computadora, entre otros

CAPITULO II

2. METODOLOGIAS DE DESARROLLO DE PLANIFICACIÓN INFORMATICA

2.1. NECESIDAD DE UNA METODOLOGIA

Al comenzar un sistema de planeación, lo primero que debe hacerse es escoger una metodología que guíe todo el proceso de planeación, desarrollo, implementación y control.

Las organizaciones que consiguen mayor beneficio con el uso de la tecnología CASE utilizan un conjunto totalmente integrado de herramientas CASE que automatizan y soportan el proceso integrado del ciclo de vida del software. La selección de la metodología debe ser el primer paso de una implantación CASE.

2.2. CARACTERISTICAS Y ESTRUCTURA DE UNA METODOLOGIA

Una metodología es la parte operativa del proceso del conocimiento. Es la manera de organizar el proceso de investigación a través de técnicas, herramientas o estrategias adecuadas. Establece operaciones concretas basadas sobre las orientaciones generales del método.

La estructura de una metodología para sistemas de información debe constar al menos de:

- 3 Modelo de Planificación del proyecto.
- 3 Notación definida (simbología a utilizar).

- 3 Esquema de flujo de información.
- 3 Jerarquización de datos.
- 3 Esquema de diseño de datos.
- 3 Esquema de diseño arquitectónico.
- 3 Reglas que acompañen a los diagramas.

2.3. METODOLOGIAS ISP Y BSP

Estas metodologías, constituyen métodos formales para los gerentes, con la finalidad de brindar apoyo en el desarrollo de sistemas de información.

METODO ISP (PLAN DE SISTEMAS DE INFORMACION): La teoría del modelo ISP es una manifestación de la filosofía Entidad Relación que usa una semántica formal.

Es una metodología en 6 fases:

1. Identificar la unidad organizacional.
2. Crear una estrategia de modelo de negocio.
3. Crear un modelo de entidad.
4. Crear un modelo funcional de trabajo.
5. Medio ambiente y arquitectura de sistema de información.
6. Plan de acción y medio ambiente del sistema.

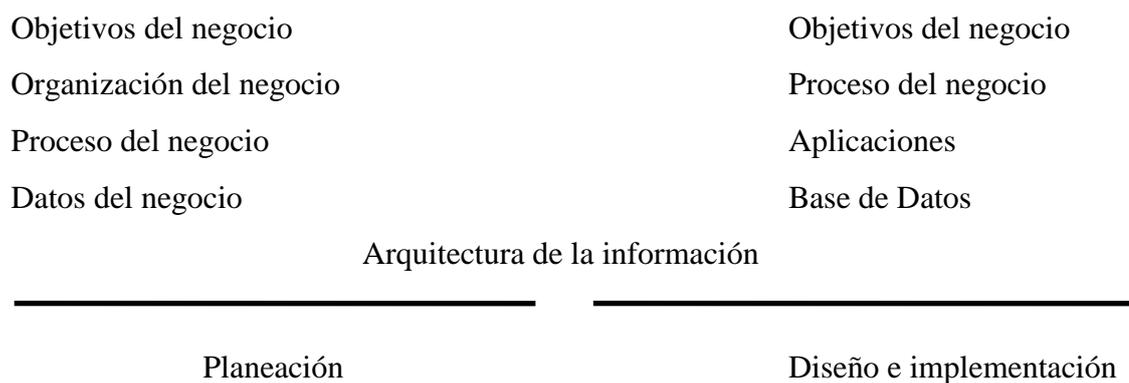
Las dos primeras fases, tienen por objetivo definir las metas de la organización; para lo cual se basa en una revisión de las operaciones, políticas y el plan de la empresa. La revisión de documentación incluye entre otras, organigramas y descripciones de puestos, manuales de procedimientos; todo esto, con el fin de identificar la unidad organizacional y crear una estrategia del negocio (organización). Además, tiene por objeto concatenar las metas de la organización con las metas de los sistemas de información.

La creación de modelos, debe definir las entidades y funciones de trabajo que se van a desarrollar. Finalmente con esto, se debe definir el entorno y acción del sistema de información; es decir, cual es el alcance que tendrá y sobre que arquitectura se la implementará. Además, se evalúa que impacto tendrán los sistemas sobre la organización.

METODO BSP (BUSINESS SYSTEMS PLANNING) Sistema de planificación para negocios: Establece un plan de sistema informáticos que satisfaga sus necesidades a corto y largo plazo. El flujo de operaciones es desde niveles superiores hacia abajo:

1. Acuerdo a alto nivel.
2. Preparación para el estudio.
3. Definir procesos.
4. Definir clases de datos.
5. Soporte de proceso de datos actual.

La arquitectura de la información sobre este esquema sería:



Esta metodología, desarrollada por IBM, se centra en la identificación de los datos necesarios para poner en marcha una organización.¹⁸

¹⁸ Senn James. *Análisis y Diseño de Sistemas de Información*. Pg. 78

Es esta metodología, los datos son vistos como un recurso muy valioso. Se había mencionado que el flujo de operaciones era desde arriba hacia abajo; de tal forma que, como primer paso se debe obtener el compromiso del máximo órgano directriz de una organización. Estos deben ser quienes definen los procesos importantes de la actividad de la misma y luego definen clases de datos que representarán entidades de interés. Como parte del análisis, se definen matrices para capturar detalles de datos.

2.4. METODOLOGIA DE LA INGENIERIA DE SOFTWARE PARA PLANIFICACIÓN INFORMATICA

En el año 1985 Boehm¹⁹ estableció siete principios básicos en la ingeniería del software para asegurar el éxito en el desarrollo de un producto software. Estos principios han sido generalmente aceptados:²⁰

1. Dirección por medio del uso de un plan por fases del ciclo de vida.
2. Realización de validaciones continuas.
3. Mantenimiento de control del producto estricto.
4. Uso de técnicas modernas de programación.
5. Mantenimiento de una contabilidad clara de la situación del proyecto.
6. Usar menos y mejor personal.
7. Mantener un compromiso de mejora del proceso.

Los principios que a continuación se presentan es la base de la filosofía estructurada, sin embargo se extendieron al análisis, diseño y a la planificación de la información.

1. Abstracción: Representación del problema de una forma general y simplificada.
2. Formalización: Seguir un procedimiento metódico y riguroso.
3. División: Dividir el problema en varios subproblemas interdependientes.
4. Ordenación jerárquica: Organizar los componentes en una estructura jerárquica.

¹⁹ Cuevas Agustín Gonzalo. *Ingeniería del Software: Práctica de la programación*. Pg. 39

5. Ocultación: Ocultar la información innecesaria.
6. Colocación: Colocar físicamente próximos los datos relacionados lógicamente.
7. Integridad conceptual: Seguir una filosofía y una arquitectura de desarrollo consistente.
8. Totalidad: Comprobar que no falta nada.
9. Independencia lógica: Concentrarse en funciones lógicas que hay que realizar, independientemente de la implantación física.²¹

Aunque las técnicas de creación del software cambian, los principios básicos permanecen constantes.



Ingeniería del Software, Pressman PG 18

Figura 10 Capas de Ingeniería de Software

En la Figura 10, nos muestra un enfoque de la Ingeniería de Software, que se fundamenta en capas. Propone que, la base es el enfoque de calidad que deben seguir un proceso de ingeniería. Los procesos, permiten un desarrollo eficiente de la ingeniería de software, los métodos nos indica cómo construir técnicamente un software y las herramientas brindan el soporte automático o no para los métodos y procesos.

2.4.1. DESCRIPCION DE SUS PROCEDIMIENTOS

²⁰ Cuevas Agustín Gonzalo, *Ingeniería del Software, Práctica de la Programación*. Pg. 39

²¹ McClure Carma. *Case La Automatización Del Software*. Pg. 328

El desarrollador de software, debe incorporar una estrategia de desarrollo denominado algunas veces modelo de proceso o paradigma de ingeniería de software. Cada modelo o estrategia se selecciona de acuerdo a la naturaleza del proyecto. Estos modelos se caracterizan por ayudar al control y coordinación de un proyecto.

Modelo Lineal Secuencial: Llamado ciclo de vida básico o el tradicional ciclo de vida de un sistema que contiene el análisis, diseño, codificación, pruebas y mantenimiento. Las fases de análisis y diseño pueden valerse de la Ingeniería de la Información porque esta recoge los requisitos en el nivel estratégico y de área de la organización.

Modelo de Construcción de Prototipos: Representación rápida de aspectos que serán visibles para el usuario.

Modelo de Desarrollo Rápido de Aplicaciones: Es un modelo de desarrollo de software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto.

Modelo Incremental: Combina aspectos del modelo lineal secuencial con la construcción de prototipos.

También se puede optar por los modelos: espiral, ensamblaje de componentes, desarrollo concurrente.

2.5. METODOLOGIA DE LA INGENIERIA DE LA INFORMACIÓN

2.5.1. PRINCIPIOS Y FUNDAMENTOS

La metodología de la ingeniería de la información de James Martin es una formulación descendente por pasos para la construcción de sistemas de información. Los primeros pasos de esta metodología se centran en la información y el modelo de la empresa a alto nivel. El objetivo es gestionar el desarrollo del sistema y la interacción a través del control de los datos. Los datos compartidos en

el sistema de información se definen en un modelo lógico y se controlan centralmente. Así, la metodología de Martin revisa todo el sistema en el intento de identificar cómo se utiliza y se comparte la información. Las funciones de alto nivel de la empresa se definen junto con la información del negocio. En los pasos posteriores, se definen con más detalles los datos y las funciones. En el último paso, se definen e implantan los datos y lógica del programa.

Las estructuras del programa se construyen en el nivel más alto de un modelo de la organización que establece una infraestructura común de información que aúna los sistemas de información utilizados en toda la organización. Así pues, los procedimientos se derivan de los datos. El diseño lógico está separado del diseño físico y lo precede.

2.5.2. PROCESOS DE DESARROLLO

La metodología de la ingeniería de la información divide el proceso de desarrollo del software en cuatro fases básicas

Estadio 1

Planificación estratégica de la Información

- 3 Diagrama de Descomposición
- 3 Diagrama de Entidad / relación
- 3 Matriz de planificación de la Información

Estadio 2

Análisis Área del Negocio

- 3 Modelo de Datos
- 3 Diagrama de flujo de Datos / dependencia
- 3 Diagrama de Descomposición

3 Matriz Entidad / proceso

Estadio 3

Diseño del Sistema

- 3 Diagrama de Estructura de datos
- 3 Diseños Pantallas / informes
- 3 Diagrama de Descomposición
- 3 Diagrama de Acción

Estadio 4

Construcción del Sistema

Estadio 1: Planificación de la estrategia de la información:

Este estadio comienza con la creación de un plan estratégico de los sistemas del negocio en el que se definen los objetivos del negocio para los próximos cinco a diez años. Junto con el plan estratégico se construyen los modelos de alto nivel y datos de la organización. El modelo de la empresa define las funciones básicas (por ejemplo, CONTABILIDAD) y la estructura de la organización. Un diagrama estructurado en árbol jerárquico, llamado diagrama de descomposición, es el que se emplea para definir las funciones del negocio y las estructuras de la organización. Para definir los tipos de datos de la organización se emplea un diagrama de entidad / relación.

Una entidad es cualquier información de la empresa que se pueda almacenar (como empleados, departamentos, clientes, facturas, etc.). Las entidades se descubren entrevistando los usuarios del sistema de información. Las visiones de los usuarios se analizan y luego se mezclan y clasifican para crear un modelo

compuesto. Además, la ingeniería inversa de los sistemas existentes puede emplearse para revelar el modelo de datos existente, el cual puede utilizarse como punto de partida para crear el modelo necesario para soportar la dirección estratégica de la empresa.

Finalmente, se determina en este estado el orden en el cual se van a desarrollar los sistemas de información que satisfagan los objetivos de la organización. El usuario tiene una importante función activa en este estadio y en todo el proceso de desarrollo de los sistemas de ingeniería de la información.

Diagrama Entidad Relación con Easy CASE

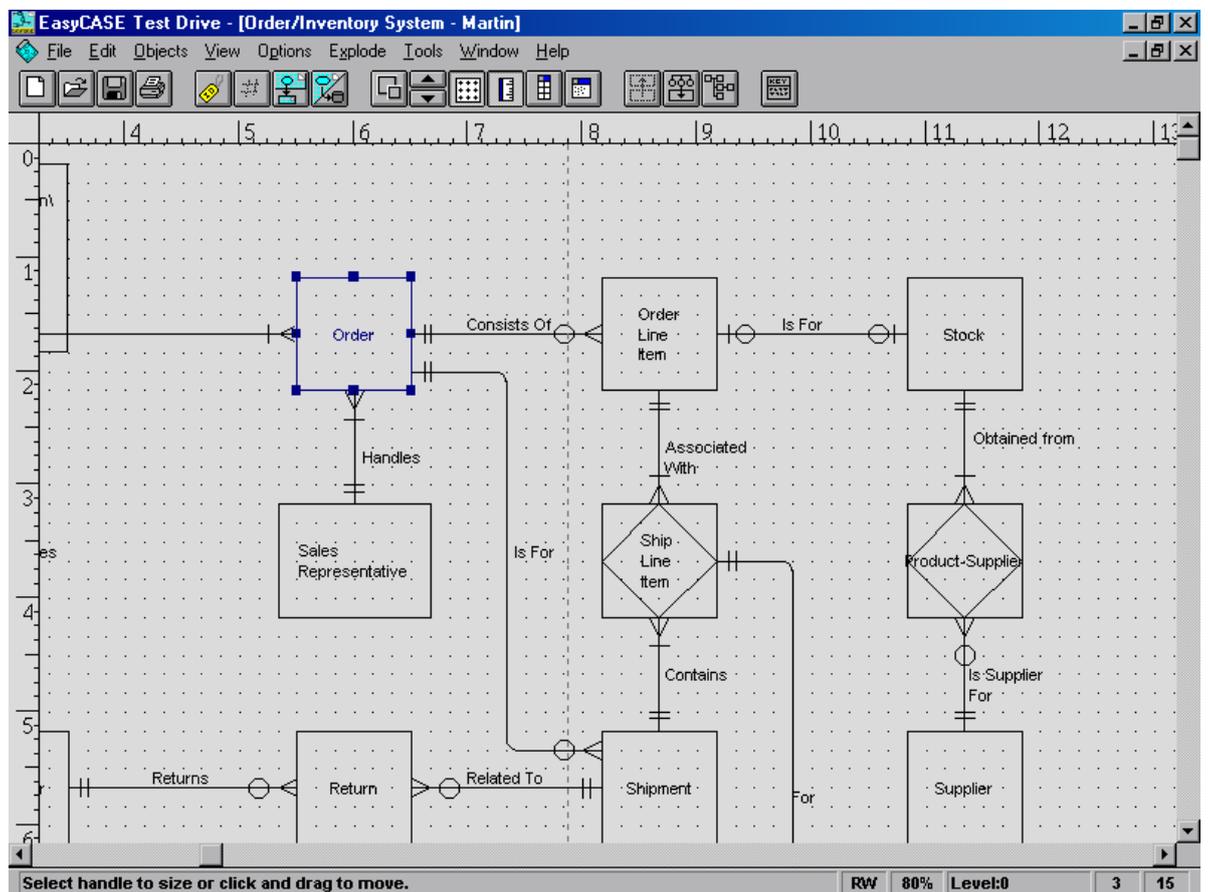


Figura 11

La Figura 11, muestra un ejemplo de Diagrama Entidad Relación mediante Easy CASE.

Estadio 2: El análisis del área del negocio:

Las funciones del negocio definidas en el estadio 1 se usan para dividir la organización en áreas lógicas del negocio (por ejemplo, COMPRAS). Después se definen y construyen los sistemas de información necesarios para soportar un área del negocio en particular.

Este estadio se concentra en la definición de los datos y procesos necesarios para satisfacer los objetivos de la empresa dentro de un área en particular del negocio. Una parte del diagrama entidad / relación desarrollada en el estadio 1 que concierne a un área en particular del negocio se extrae y desarrolla en un modelo de datos completamente normalizado. Se definen completamente las entidades, sus atributos y sus relaciones recíprocas.

El diagrama de descomposición del estadio 1 que representa las funciones del negocio, se descompone en procesos por cada una de las áreas del negocio. El proceso puede representarse en diagramas de dependencia (un tipo de diagrama de flujo de datos), en diagramas de flujo de datos o en diagramas de descomposición. Se emplea una matriz de entidad / proceso para relacionar los datos con los procesos en los que intervienen.

Estadio 3: El diseño del sistema:

El estadio 3 se ocupa de las consideraciones de diseño del sistema lógico. Los procesos del negocio definidos en el estadio 2 conforman las bases para el desarrollo del sistema en este estadio. Se diseñan los procedimientos necesarios para realizar los procesos y las estructuras lógicas en sus datos. Se utiliza la descomposición funcional descendente para diseñar los procesos. En este estadio, la metodología de la ingeniería de la información se hace similar a la metodología

tradicional de la ingeniería de software para el desarrollo de programas. El sistema diseñado se representa con diagramas de estructura de datos, con diseños y con diagramas de acción y descomposición. Los diagramas de acción muestran detalladamente la lógica del procedimiento y los accesos a las bases de datos.

El prototipo de la interfaz de usuario (en pantallas, diálogo o gráficos) se emplea para descubrir los requerimientos del usuario. Es necesaria la implicación del usuario para ayudar a construir y comprobar los modelos de prototipos.

Estadio 4: La construcción:

Mientras que el estadio 3 trataba el diseño del sistema lógico, el estadio 4 se ocupa del diseño del sistema físico y de la implantación del programa. Durante el estadio 4, toda la información del sistema se entrega como una base de datos lógica, una base de datos física y un código documentado de trabajo. Las herramientas CASE, como los lenguajes de cuarta generación, herramientas de soporte de decisiones y las generaciones de códigos, se emplean en el estadio 4.

2.5.3. DIAGRAMAS UTILIZADOS

Los tipos de diagramas necesarios para soportar una formulación de ingeniería de la información para desarrollar un sistema son:

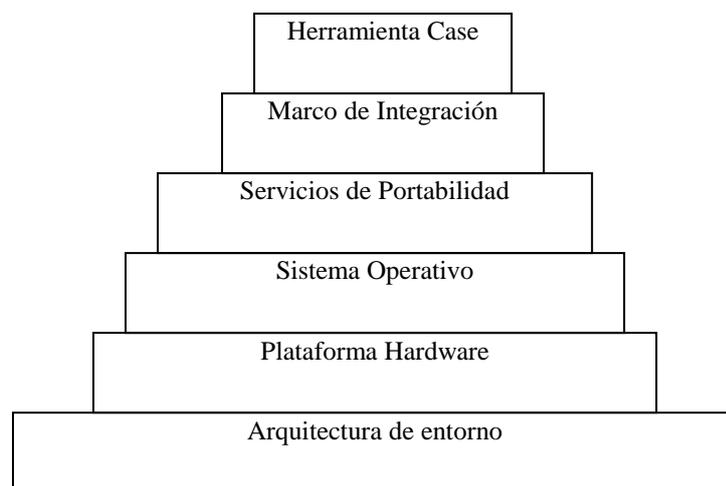
- + Diagrama de entidad / relación: muestra los tipos de entidades y sus relaciones.
- + Diagrama de la estructura de los datos: muestra los tipos de entidades, sus atributos y sus relaciones.
- + Diagrama de descomposición jerárquica estructurada en árbol: muestra el modelo corporativo de alto nivel y la estructura arquitectónica

jerárquica de las actividades del negocio, de los procesos, de las aplicaciones, de los procedimientos y de los programas.

- + Diagramas de flujo de datos y de dependencia: muestran el flujo de los datos entre procesos.
- + Diseños de pantalla y de informes: muestran el diseño de la interfaz del usuario.
- + Diagramas de acción: muestran detalladamente la lógica de los programas.

2.6. HERRAMIENTAS CASE

2.6.1. RELACION DE LA CASE CON OTRAS TECNOLOGIAS



Ingeniería del Software, Pressman PG 542

Figura 12 Bloques de Construcción Case

Las Herramientas Case, pueden ser; una herramienta puntual que abarque una actividad específica o puede ser un conjunto integrado que parte desde la arquitectura del entorno. En la Figura 12, podemos ver, que las Herramientas Case están en la parte final de los bloques de construcción. Esto se debe, a que las Case se deben soportar sobre una arquitectura de hardware y software. La arquitectura de entorno y plataforma hardware, nos da la base sobre la cual el software trabajará; el sistema operativo y los servicios de portabilidad son el nexo entre los distintos usuarios y software que se encuentren interactuando, además de permitirles migrar hacia diferentes plataformas. Finalmente, el marco de integración permite la comunicación entre las diversas herramientas Case presente y que muestra al usuario las mismas especificaciones de un proyecto.

Las principales líneas de evolución hacia las que parecen encaminarse las herramientas CASE son:

CASE para sistemas bajo arquitectura cliente/servidor: No hay que confundir el hecho de que una herramienta CASE funcione en un entorno de arquitectura cliente/servidor, con, que el sistema desarrollado mediante una herramienta CASE vaya a funcionar bajo dicha arquitectura.

En la actualidad ya hay ejemplos de los dos casos, herramientas CASE que funcionan bajo un entorno cliente/servidor, en red y con un repositorio centralizado en un servidor y herramientas CASE que generan aplicaciones que funcionan en un entorno cliente/servidor, en las cuales se puede indicar dónde deben residir los componentes de la aplicación en tiempo de ejecución, liberando al programador de aspectos referidos a los protocolos de comunicaciones, seguridad, interfases gráficas de usuario, etc.

La línea de evolución, en este caso, vendrá marcada por versiones mejoradas de la herramienta, que faciliten cada vez más la distribución de los elementos de una aplicación entre los diferentes clientes y servidores y una mayor liberalización del

programador, de todos los aspectos que no sean propios de la aplicación (protocolos de red, seguridad, etc.).

CASE multiplataforma: Estas herramientas soportan las combinaciones dominantes de diferentes plataformas físicas, sistemas operativos, interfases gráficas de usuario, sistemas de gestión de bases de datos, lenguajes de programación y protocolos de red. En este sentido el futuro podrá ser de apertura creciente a nuevas plataformas y portabilidad más generalizada.

CASE para ingeniería inversa y directa: Ya existen algunas herramientas de este tipo. Su evolución marcará notables mejoras en la obtención de los diseños a partir del código ya existente (ingeniería inversa) y la regeneración del mismo, una vez optimizado el diseño (ingeniería directa).

CASE para trabajo en grupo (groupware): Estas herramientas se centran en el proceso de desarrollo más que en el producto a desarrollar, facilitando la integración de diferentes grupos humanos, pertenecientes incluso a empresas diferentes, trabajando conjuntamente en un gran proyecto.

Estas deberían incorporar las facilidades clásicas de ofimática: correo electrónico, calendarios en línea, planificación de actividades, preparación de documentos, actas de reuniones, etc.

CASE para desarrollo de sistemas orientados a objetos: En la actualidad existen algunas herramientas que cubren alguna de las fases del ciclo de vida de desarrollo de aplicaciones orientadas a objetos (Interfase de usuario, análisis, diseño, programación, etc.). El objetivo futuro podría ser cubrir el ciclo de vida completo.

Una atención especial merece las herramientas CASE adaptables, algunas de las cuales permiten que sea el propio usuario quien defina su metodología y los

símbolos de las notaciones a utilizar. Estas herramientas se denominan "meta-CASE".

2.6.2. CATEGORIAS DE HERRAMIENTAS CASE

No existe una única clasificación de herramientas CASE y algunos autores identifican de diferente manera a cada herramienta. Haciendo un agrupamiento y coincidiendo entre los autores podemos clasificarlas de acuerdo a:

- © Las plataformas que soportan.
- © Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- © La arquitectura de las aplicaciones que producen.
- © Su funcionalidad.

Las herramientas CASE, en función de las fases del ciclo de vida abarcadas, se pueden agrupar de la forma siguiente:

Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.

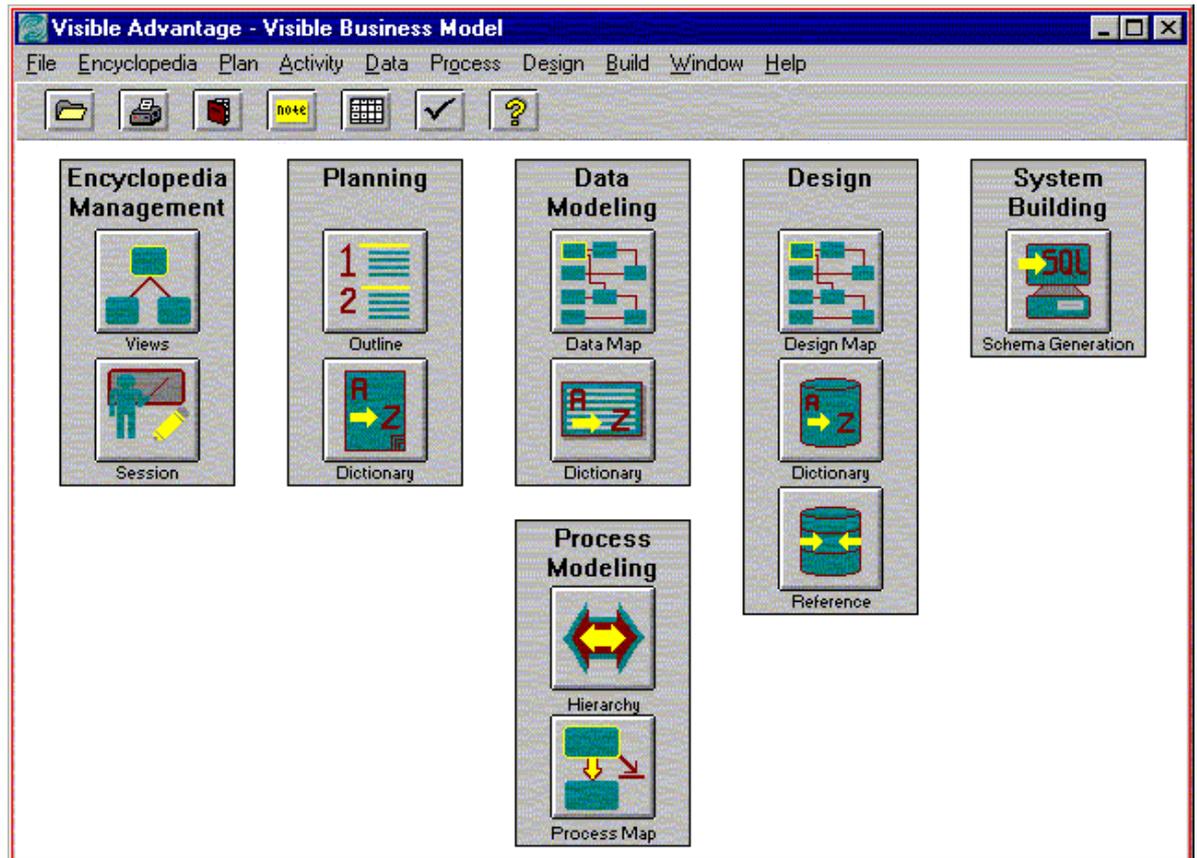


Figura 13 Modelo de Negocios con Visible Advantege

La Figura 13, muestra una herramienta integrada.

Herramientas que comprenden algunas fases del ciclo de vida de desarrollo de software:

Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior) o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior) o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación.

Juegos de herramientas o toolkits, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Las herramientas I-CASE se basan en una metodología. Tienen un repositorio y aportan técnicas estructuradas para todas las fases del ciclo de vida. Estas son las características que les confieren su mayor ventaja: una mejora de la calidad de los desarrollos.

Utilizando la funcionalidad como criterio principal, su clasificación sería:

Herramientas de planificación de sistemas de gestión.

Sirven para modelizar los requisitos de información estratégica de una organización. Proporcionan un "metamodelo" del cual se pueden obtener sistemas de información específicos. Su objetivo principal es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.

Herramientas de análisis y diseño.

Permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Se tienen:

- © Herramientas de análisis y diseño.
- © Herramientas de creación de prototipos y de simulación.
- © Herramientas para el diseño y desarrollo de interfases.
- © Máquinas de análisis y diseño.

Herramientas de programación.

Se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Ejemplos de estas herramientas son:

- © Herramientas de codificación convencionales.
- © Herramientas de codificación de cuarta generación.
- © Herramientas de programación orientadas a objetos.

Herramientas de integración y prueba:

Sirven de ayuda a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:

- © Herramientas de análisis estático.
- © Herramientas de codificación de cuarta generación.
- © Herramientas de programación orientadas a objetos.

Herramientas de gestión de prototipos.

Los prototipos son utilizados ampliamente en el desarrollo de aplicaciones, para la evaluación de especificaciones de un sistema de información, o para un mejor entendimiento de cómo los requisitos de un sistema de información se ajustan a los objetivos perseguidos.

Herramientas de mantenimiento:

La categoría de herramientas de mantenimiento se puede subdividir en:

- © Herramientas de ingeniería inversa.
- © Herramientas de reestructuración y análisis de código.
- © Herramientas de reingeniería.

Herramientas de gestión de proyectos.

La mayoría de las herramientas CASE de gestión de proyectos, se centran en un elemento específico de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de

esfuerzo, coste y duración, hacer un seguimiento continuo del proyecto, estimar la productividad y la calidad, etc. Se incluyen dentro de las herramientas de control de proyectos las siguientes:

- © Herramientas de planificación de proyectos.
- © Herramientas de seguimiento de requisitos.
- © Herramientas de gestión y medida.

Herramientas de soporte.

Se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se mencionan a continuación:

- © Herramientas de documentación.
- © Herramientas para software de sistemas.
- © Herramientas de control de calidad.
- © Herramientas de bases de datos.

Diferenciando las funciones CASE tenemos cinco grupos:

Repositorio.

Funcionan en torno a un repositorio central, siendo éste el núcleo fundamental que contiene todas las definiciones de objeto y sus relaciones. Los objetos pueden ser especificaciones del sistema en forma de diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas, etc. El repositorio es un concepto más amplio que el de diccionario de datos y soporta a los demás grupos de funciones. No es fácil encontrar en el mercado productos Case con funcionalidades estrictamente a las de repositorio, ya que, a pesar de su innegable importancia, tienen un carácter auxiliar de los demás grupos de funciones. Cualquier sistema Case

poseerá un repositorio propio o bien, trabajará sobre un repositorio suministrado por otro.

Re-ingeniería.

Los sistemas Case permiten establecer una relación estrecha y fuertemente formalizable entre los productos generados a lo largo de distintas fases del ciclo de vida, permitiendo actuar en el sentido especificaciones-código (ingeniería directa) y también en el contrario (ingeniería inversa). Ello facilita la realización de modificaciones en la fase más adecuada en cada caso y su traslado a las demás.

Soporte del ciclo de vida.

El ciclo de vida de una aplicación o de un sistema de información se compone de varias etapas, que van desde la planificación de su desarrollo hasta su implantación, mantenimiento y actualización. Aunque el número de fases puede ser variable en función del nivel de detalle que se adopte, pueden de modo simplificado, identificarse las siguientes:

- © Planeamiento.
- © Análisis y Diseño.
- © Implantación (programación y pruebas).
- © Mantenimiento y actualización.

Soporte de proyecto.

Este tipo de funciones hace referencia al soporte de actividades que se producen durante el desarrollo, derivadas fundamentalmente del trabajo en grupos, tales como facilidades de comunicación, soporte a la creación, modificación e intercambio de documentación, herramientas personales, controles de seguridad, etc. Los sistemas Case pueden conceder a estas cuestiones una importancia variable por lo cual el soporte de proyecto constituye un factor de diferenciación.

Mejora continua de calidad.

Aunque frecuentemente se asocia a los sistemas Case con la mejora de la productividad en el desarrollo de aplicaciones, debe tenerse en cuenta que una de las principales ventajas estriba también, en la mejora de la calidad de los desarrollos realizados.

2.6.3. METODOLOGIAS QUE SOPORTAN LAS HERRAMIENTAS CASE

Las herramientas CASE, soportan todas la metodologías existentes. Estas pueden ser:

1. Desarrollo estructurado de sistemas
2. Desarrollo estructurado de Learmonth & Burchett
3. Diseño estructurado de Yourdon
4. Análisis estructurado de DeMarco
5. Diseño estructurado de Orr
6. Diseño estructurado de Jackson
7. Análisis estructurado de Gane-Searson
8. Desarrollo de prototipos
9. Ingeniería de la Información de Martin
10. Desarrollo SQL, etc.

2.6.3.1. IMPLEMENTACION DE UNA METODOLOGIA EN UNA HERRAMIENTA CASE

Para implementar una metodología, se requiere del compañero de metodología, que es una herramienta que estructura el proceso de desarrollo de software de acuerdo a las fases y reglas de la metodología escogida para la herramienta. Este compañero de metodología no permite ir al siguiente paso hasta que la tarea que se esté realizando este completa de acuerdo a las fases.

2.6.4. PROCESOS QUE SOPORTA EL SOFTWARE CASE

Repositorio

Base de datos central de una herramienta CASE. El repositorio amplía el concepto de diccionario de datos para incluir toda la información que se va generando a lo largo del ciclo de vida del sistema, como por ejemplo: componentes de análisis y diseño (diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas), estructuras de programas, algoritmos, etc. En algunas referencias se le denomina Diccionario de Recursos de Información.

Apoyándose en la existencia del repositorio se efectúan comprobaciones de integridad y consistencia:

- 3 Que no existan datos no definidos.
- 3 Que no existan datos autodefinidos (datos que se emplean en una definición pero que no han sido definidos previamente).
- 3 Que todos los alias (referencias a un mismo dato empleando nombres distintos) sean correctos y estén actualizados.

Las características más importantes de un repositorio son:

- ↳ Tipo de información. Que contiene alguna metodología concreta, datos, gráficos, procesos, informes, modelos o reglas.
- ↳ Tipo de controles. Si incorpora algún módulo de gestión de cambios, de mantenimiento de versiones, de acceso por clave, de redundancia de la información.
- ↳ Tipo de actualización. Si los cambios en los elementos de análisis o diseño se ven reflejados en el repositorio en tiempo real o mediante un

proceso por lotes (batch). Esto será importante en función a la necesidad de que los cambios sean visibles por todos los usuarios, en el acto.

- ↳ Reutilización de módulos para otros diseños. El repositorio es la clave para identificar, localizar y extraer código para su reutilización.
- ↳ Posibilidad de exportación e importación para extraer información del repositorio y tratarla con otra herramienta (formateo de documentos, mejora de presentación) o incorporar al repositorio, información generada por otros medios.
- ↳ Interfases automáticas con otros repositorios o bases de datos externos.

2.6.4.1. DIAGRAMACION AUTOMATICA

Algunos de los diagramas y modelos utilizados con mayor frecuencia son:

- Diagrama de flujo de datos.
- Modelo entidad - relación.
- Historia de la vida de las entidades.
- Diagrama Estructura de datos.
- Diagrama Estructura de cuadros.
- Técnicas matriciales.

Algunas características referentes a los diagramas son:

- ┌ Número máximo de niveles para poder soportar diseños complejos.
- ┌ Número máximo de objetos que se pueden incluir para no encontrarse limitado en el diseño de grandes aplicaciones.
- ┌ Número de diagramas distintos en pantalla o al mismo tiempo en diferentes ventanas.
- ┌ Dibujos en formato libre con la finalidad de añadir comentarios, dibujos, información adicional para aclarar algún punto concreto del diseño.

- ┌ Actualización del repositorio por cambios en los diagramas. Siempre resulta más fácil modificar de forma gráfica un diseño y que los cambios queden reflejados en el repositorio.
- ┌ Control sobre el tamaño, fuente y emplazamiento de los textos en el diagrama.
- ┌ Comparaciones entre gráficos de distintas versiones. De esta forma será más fácil identificar qué diferencias existen entre las versiones.
- ┌ Inclusión de pseudocódigo que servirá de base a los programadores para completar el desarrollo de la aplicación.
- ┌ Posibilidad de deshacer el último cambio facilitando que un error no conlleve perder el trabajo realizado.



Figura 14 ENTIDADES EN DBEASY

La Figura 14, muestra un ejemplo de él contenido de entidades en DbEasy.

Prototipado

El objetivo principal de esta herramienta es poder mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá la aplicación una vez desarrollada. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la fase de diseño.

La herramienta será más útil, cuanto más rápidamente permita la construcción del prototipo y por tanto antes, se consiga la implicación del usuario final en el diseño de la aplicación. Asimismo, es importante poder aprovechar como base el prototipo para la construcción del resto de la aplicación.

2.6.4.2. VERIFICACION DE ERRORES

En la verificación de errores tenemos a los errores de tipo y de sintaxis, las cuales se basan en las reglas de los diagramas de flujos de datos. Así, cada representación de proceso debe tener por lo menos un flujo de datos de entrada y por lo menos uno de salida. Dos datos de almacenamiento no pueden conectarse directamente. En los errores de tipo tenemos que un símbolo de proceso debe usarse siempre para representar un componente de procedimiento, etc. Las reglas de tipo se introducen para eliminar ambigüedades que puedan causar confusión cuando se intenta comprender una especificación o la transformación en código.

2.6.4.3 VERIFICACION DE INTEGRIDAD Y CONSISTENCIA

El propósito de esta verificación es asegurarse que el diagrama ha sido completado y que contiene toda la información necesaria. Esta verificación se la realiza cuando se haya terminado de elaborar el diagrama. Podemos mencionar, como ejemplo, la comprobación de que todos los componentes del diagrama estén debidamente etiquetados. También debe asegurarse de que cada objeto o elemento conste en las definiciones en el depósito CASE.

2.6.4.4. VERIFICACION DE DESCOMPOSICION FUNCIONAL

Esta verificación se la realiza en una estructura de árbol jerárquico. Observando la jerarquía, los componentes de cada nivel sucesivo contienen funciones que definen las funciones en el nivel anterior. Toda metodología estructurada contiene reglas para la descomposición funcional.

La verificación de la descomposición funcional es una forma de control de calidad que puede realizarse en un diagrama de estructura de árbol.

Una clase especial de verificación de la descomposición funcional se denomina refinamiento semántico, en ella cada descomposición de una función en subfunciones se comprueba para asegurar que proporciona información más detallada sobre la función. Como ejemplo se puede mencionar que; si en un nivel una función se describe como realizadora de una operación de entrada/salida, en el siguiente nivel debe definirse el tipo de operación de entrada/salida.

2.6.4.5. VERIFICACION DE LA METODOLOGIA

Al verificar la metodología depende de las particularidades de la metodología que se haya implantado en el CASE Workbench. Al verificar los errores, diagramas, sintaxis, se está comprobando la metodología, debido a que cada una tiene diferentes reglas y procedimientos.

2.6.4.6. EL DEPOSITO CASE

Podemos definir a la palabra depósito como todo objeto o persona que se considera como centro de acumulación o almacenamiento.

Durante las primeras fases de la historia del desarrollo del software, el depósito era en realidad una persona, el programador que tenía que recordar la ubicación de toda la información relevante para un determinado proyecto de software, que tenía que recordar información que nunca se había escrito y reconstruir información que se había perdido. En la actualidad, el depósito es una base de datos que actúa como centro tanto para la acumulación como para el almacenamiento de información de ingeniería del software. El papel de la persona (del ingeniero del software) es interactuar con el depósito empleando herramientas CASE integradas con él.

Se utiliza un determinado número de términos distintos para hacer alusión al lugar de almacenamiento de la información de la ingeniería del software: base de datos

CASE, base de datos del proyecto, base de datos de entorno de apoyo de proyecto integrado, diccionario de datos (una base de datos limitada) y depósito. Aún cuando existen sutiles diferencias entre algunos de estos términos, todos ellos se refieren al centro de acumulación y almacenamiento.

El depósito de un entorno I-CASE es el conjunto de mecanismos y de estructuras de datos que consiguen la integración entre datos y herramientas y entre datos y datos. Proporciona las funciones y herramientas de un sistema de gestión de bases de datos, pero además, el depósito lleva a cabo o precipita las funciones siguientes:

- › Integridad de datos: incluye funciones para validar las entradas efectuadas en el depósito, para asegurar la consistencia entre objetos relacionados, cuando un cambio efectuado en un objeto exige algún cambio en otros objetos relacionados con él.
- › Información compartida: proporciona un mecanismo para compartir información entre múltiples desarrolladores y entre múltiples herramientas, gestiona el control multiusuario a los datos y bloquea/desbloquea objetos para que los cambios no se superpongan inadvertidamente.
- › Integración datos-herramientas: establece un modelo de datos al que pueden acceder todas las herramientas del entorno I-CASE, controla el acceso a los datos y lleva a cabo las funciones de gestión de configuración adecuadas;
- › Integración datos-datos: el sistema de gestión de bases de datos relaciona los objetos de datos de tal modo que se puedan llevar a cabo las demás funciones;
- › Imposición de la metodología: el modelo E-R de datos almacenado en el depósito puede implicar un paradigma específico de ingeniería del software, como mínimo, las relaciones y los objetos definen un conjunto de pasos que será preciso realizar para construir el contenido del depósito; y
- › Estandarización de documentos: la definición de objetos de la base de datos da lugar directamente a un enfoque estándar para la creación de documentos de ingeniería del software.

Para realizar estas funciones, se define el depósito en términos de un metamodelo. El metamodelo determina la forma en que se almacena la información en el depósito, la forma en que las herramientas pueden acceder a los datos y estos datos pueden ser visualizados por los ingenieros de software, el grado hasta el cual se puede mantener la seguridad e integridad de los datos y la facilidad con que se puede extender el modelo ya existente para admitir nuevas necesidades.

El metamodelo es la plantilla en la cual se sitúa la información de ingeniería del software.

Características y contenidos

Las características y contenido del depósito se establecen a partir de determinar qué es lo que hay que almacenar en el depósito y qué servicios específicos son los que proporciona el depósito. Estos incluyen:

- El problema que hay que resolver
- Información acerca del dominio del problema
- La solución del sistema a medida que vaya surgiendo
- Las reglas e instrucciones relativas al proceso de software (metodología) que se está siguiendo
- El plan del proyecto, sus recursos y su historia
- Información acerca del contexto organizativo

Muchos requisitos del depósito son iguales a los de aplicaciones típicas que se construyen tomando como base un sistema de gestión de bases de datos (SGBD). De hecho, muchos de los depósitos CASE actuales hacen uso de un SGBD como tecnología de gestión de datos básica. Las características de un SGBD estándar en un depósito CASE que preste su apoyo para la gestión de información de desarrollo del software incluye lo siguiente:

Almacenamiento de datos no redundante. El depósito CASE proporciona un único lugar para el almacenamiento de toda la información pertinente al desarrollo de los sistemas de software, eliminando una duplicación que supone un desperdicio y es potencialmente proclive a errores.

Acceso de alto nivel. El depósito proporciona un mecanismo de acceso a los datos común de tal modo que no sea preciso duplicar las capacidades de gestión de datos en todas las herramientas CASE.

Independencia de datos. Las herramientas CASE de las aplicaciones blanco se aíslan del almacenamiento físico para que no se vean afectadas cuando cambie la configuración.

Control de transacciones. El depósito gestiona las interacciones entre múltiples partes de tal modo que se mantiene la integridad de los datos cuando existen usuarios concurrentes y en caso de fallo del sistema.

Seguridad. El depósito proporciona mecanismos para controlar quién puede visualizar y modificar la información contenida en él. Como mínimo, el depósito debería de imponer contraseñas multinivel y niveles de permiso que fueran asignados a usuarios individuales.

El depósito debería de proporcionar también asistencia para copias de seguridad y restauraciones automáticas y también para archivar grupos seleccionados de informaciones.

Apertura. Los depósitos suelen proporcionar un mecanismo de importación/exportación.

Apoyo multiusuario. Un depósito robusto debe de permitir que múltiples desarrolladores trabajen en una aplicación al mismo tiempo. Debe de gestionar el acceso concurrente a la base de datos mediante múltiples herramientas y por parte de múltiples usuarios, con arbitraje de accesos y con bloqueos en el nivel de archivos o registros.

El entorno CASE también efectúa demandas especiales con respecto al depósito que van más allá de lo que está disponible directamente en un SGBD comercial. Las características especiales de los depósitos CASE incluyen:

- ↗ *Almacenamientos de estructuras de datos sofisticadas.* El depósito debe de admitir tipos de datos complejos tales como diagramas, documentos y archivos, así como sencillos elementos de datos. Un depósito también incluye un modelo de información (metamodelo) que describe la estructura, relaciones y semántica de los datos almacenados en él. El metamodelo debe de poder extenderse para dar cabida a nuevas representaciones y a informaciones organizativas únicas.
- ↗ *Interfaz de herramientas semánticas.* El modelo de información del depósito (metamodelo) contiene una semántica que hace posible que toda una gama de herramientas interpreten el significado de los datos almacenados en el depósito. Entonces otra herramienta CASE puede interpretar el contenido del depósito y emplear la información según la necesite para su tarea. De este modo, la semántica almacenada en un depósito permite compartir datos entre una gama de herramientas.
- ↗ *Gestión de procesos/proyectos.* Un depósito contiene información no sólo acerca de la aplicación de software en sí, sino también acerca de las características de cada proyecto particular y del proceso general de la organización para ingeniería del software (fases, tareas y productos).

Las siguientes características del depósito son abarcadas todas ellas por la gestión de configuración del software.

- ↗ *Versiones.* A medida que progresa un proyecto, se irán creando muchas versiones. El depósito debe de ser capaz de guardar todas estas versiones para hacer posible una gestión efectiva de las versiones de los productos y para permitir que los desarrolladores vuelvan a las versiones anteriores durante la comprobación y depuración.
- ↗ *Seguimiento de dependencias y gestión de cambios.* El depósito gestiona una amplia variedad de relaciones entre los elementos de datos almacenados en él. Entre estas se cuentan las relaciones entre entidades y procesos de la empresa, entre las partes de un diseño de aplicación, entre componentes del diseño y la arquitectura de la información de la empresa, entre elementos de diseño y productos, etc. Algunas de las relaciones son meramente asociaciones y algunas son dependencias o relaciones obligatorias. El mantenimiento de estas relaciones entre objetos de desarrollo se denomina administración de enlaces.
- ↗ *Seguimiento de requisitos.* Una función especial que depende de la gestión de enlaces es el seguimiento de requisitos. Esta es la capacidad de seguir la pista de todos los componentes de diseño y de todos los productos que resulten de una especificación de requisitos concreta (seguimiento progresivo) así como la capacidad de identificar el requisito que haya generado cualquier producto dado (seguimiento regresivo).
- ↗ *Seguimiento de una auditoría.* Relacionada con la gestión de cambio está la necesidad de un seguimiento de auditoría que establezca informaciones adicionales acerca de cuándo, por qué y por quién son efectuados los cambios. La información acerca de las modificaciones efectuadas en el código fuente se puede introducir en forma de atributos de objetos específicos del depósito. Un mecanismo de activación en el depósito resultará útil para solicitar al desarrollador o a la herramienta que esté utilizando que inicie la introducción de información de auditoría (tal como la razón del cambio) siempre que se modifique un elemento de diseño.

2.6.5. CAMBIOS EN EL USO DE LAS HERRAMIENTAS CASE

La CASE representa un cambio fundamental en la actitud hacia el desarrollo del software.

A finales de los años sesenta; la introducción de las técnicas estructuradas ya representó un cambio en la actitud. Reaccionando ante el nacimiento de la crisis del software, entonces el punto de vista era que el desarrollo de software era un proceso complicado, tedioso y proclive al error. No debería ser tratado como arte privado de cada programador individual, sino como un proceso disciplinado y mecánico. El concepto de estructuración hacía hincapié en los estándares y procedimientos de programación y en el control de la gestión. La actitud era que la productividad y calidad del software podrían controlarse mejor a través de la disciplina, la formalidad y la estandarización.

Después, a finales de los años setenta volvió a suceder otro cambio de actitud en el proceso del software. Esta vez el punto de vista era que la productividad del software podría incrementarse con la utilización de herramientas que simplificasen el desarrollo de software. La crisis del software podría aliviarse si muchas personas pudieran desarrollar ciertos tipos de software para aplicaciones. Este cambio estuvo marcado por la introducción de las herramientas de cuarta generación y la programación dirigida al usuario final.

Actualmente, desde finales de los años ochenta, vemos otro cambio de actitud ante el proceso del software. En un sentido, este cambio representa un paso atrás, hacia una visión más conservadora del desarrollo de software. De nuevo, el punto de vista es que el desarrollo de software es una tarea difícil que requiere una formulación disciplinada y mecánica, y que para la realización de sistemas complejos son necesarios profesionales del desarrollo de software altamente preparados. Aunque la programación orientada al usuario final y las herramientas de cuarta generación han realizado una valiosa contribución, la experiencia ha demostrado que tienen un puesto limitado en el desarrollo de software.

El cambio más reciente es un redescubrimiento de los principios fundamentales de la ingeniería del software definidos durante los últimos 25 años. Aparecen de nuevo las técnicas estructuradas pero esta vez reforzadas con el concepto de la automatización del proceso de software. La CASE es un renacimiento de las técnicas estructuradas.

La actitud de la CASE es que el desarrollo y el mantenimiento del software debería verse como una actividad formal y disciplinada capaz de una comprobación mejor de la exactitud y automatización del proceso. La única forma real para incrementar significativamente la calidad y la productividad del software es a través de la automatización.

2.6.6. HACIA DONDE SE DIRIGE LA CASE

Las principales líneas de evolución hacia las que parecen encaminarse las herramientas CASE son:

- La utilización de la tecnología multimedia.
- La incorporación de técnicas de inteligencia artificial.
- Sistemas de realidad virtual.

2.7. ACOPLAMIENTO DE METODOLOGIAS

Opciones de Integración

Las herramientas Case pueden ser integradas de muchas formas. En un extremo se utiliza una herramienta CASE de forma aislada. Se crea un número limitado de elementos de configuración de software (documentos, programas o datos) que se manipulan mediante una única herramienta y cuya salida tiene el formato de copia de pantalla y/o documentación gráfica.

Pocas herramientas CASE se utilizan en forma aislada. Se suele disponer de las siguientes opciones:

Niveles de Integración CASE:

- a. Intercambio de datos,
- b. Acceso común a herramientas,
- c. Integración de datos,
- d. Integración total.

a) *Intercambio de datos*. La mayoría de las herramientas permiten exportar datos en forma de archivo sin estructura con un formato conocido. Esto permite un intercambio de datos punto a punto entre las distintas herramientas CASE, utilizando normalmente un filtro de transmisión intermedio.

La desventaja del intercambio de datos punto a punto está en que, a menudo, sólo parte de los datos exportados es utilizable por la herramienta receptora, ya que no fue diseñada para ser totalmente compatible. Además, a medida que evoluciona el software, la necesidad de transferir archivos cada vez que se hace un cambio pequeño puede llevar mucho tiempo. No hay posibilidad de que los cambios se reflejen en ambos sentidos y, es difícil hacer comprobaciones cruzadas de documentos y mantener la integridad de la configuración a través de las distintas herramientas que se estén utilizando.

b) *Acceso común a herramientas*. Permite al usuario utilizar distintas herramientas de forma similar. En un entorno multitarea, un usuario podría abrir simultáneamente varias herramientas, coordinando manualmente sus entradas y comparando las representaciones de diseño a medida que evolucionan. Por ejemplo, el usuario podría visualizar un diagrama de flujo

de datos, un diagrama de estructura, un diccionario de datos y un segmento de código fuente, todos mantenidos por diferentes herramientas.

c) *Integración de Datos:*

1) *Gestión común de datos.* Los datos de distintas herramientas se pueden mantener en una única base de datos lógica, que puede estar físicamente centralizada o distribuida. Hay una modalidad de fusión que permite combinar el trabajo de varias personas trabajando en diferentes partes de una aplicación.

Aunque los datos generados por las distintas herramientas se gestionan de forma conjunta en el nivel de gestión de datos comunes, las herramientas no conocen de forma explícita las estructuras de datos y la semántica de representación del diseño de las demás. Consecuentemente, se requiere una etapa de traducción para permitir que una herramienta utilice la salida generada por otra.

2) *Datos compartidos.* Las herramientas del nivel de datos compartidos tienen estructuras de datos y semántica compatible, pudiendo intercambiar datos sin necesidad de una etapa de traducción.

3) *Interoperabilidad.* Las herramientas que combinan las características de acceso común y la capacidad de compartir datos, tienen la capacidad de interoperación. Esto representa el mayor nivel de integración entre herramientas diferentes.

d) *Integración total.* Para alcanzar la integración total del entorno CASE se necesitan dos características más: gestión de metadatos y capacidad de control. Los metadatos representan información sobre los datos de ingeniería generados por las distintas herramientas CASE. Esta información incluye:

- 3 Definiciones de objetos (tipos, atributos, representaciones y relaciones válidas).
- 3 Relaciones y dependencias entre objetos (un proceso en un DFD, una entidad única o un fragmento de código de una subrutina).
- 3 Reglas de diseño del software (las distintas formas válidas de dibujar y equilibrar un diagrama de flujo de datos).
- 3 Procedimientos (fases estándar, informes, etc.) y sucesos (revisiones, finalizaciones, informes de problemas, peticiones de cambios, etc.) del flujo de trabajo (proceso).

CAPITULO III

3. ESTUDIO Y ANALISIS COSTO / BENEFICIO DE PLANEACION Y DESARROLLO INFORMATICO

3.1. ESTUDIO DE EQUIPO DE PROCESAMIENTO

Las herramientas CASE abarcan también al análisis de la plataforma hardware existente en una organización, en razón de que depende mucho del equipamiento existente para poder implantar una herramienta que pueda correr bajo aquellos equipos.

Además, la debida importancia que debe tener el equipamiento de hardware radica en gran medida en que la información que generan nuestras organizaciones ha crecido en grandes volúmenes, por lo que, estas ayudarán a manejar con una mejor efectividad dicha información.

Pero no solamente manejar, almacenar o distribuir información y/o datos debe ser la finalidad de las computadoras. Estas también deben servir para mejorar las relaciones de trabajo, fomentar la administración, elevar la productividad, incrementar las capacidades humanas, etc.

3.1.1. REQUERIMIENTOS DE LA ORGANIZACIÓN

Toda organización actualmente, requiere de cambios innovadores para sus tareas administrativas y operacionales. Los requerimientos son variados y de diversa índole. Pueden ir desde la adquisición de grandes equipos hasta los más sofisticados.

Para tener una mejor valoración de qué recursos debe tener una organización, es importante asegurarse que quien elabora dichos requerimientos, sea una persona que conozca el ambiente de la organización y que además sepa cual es la misión estratégica de la misma. De ahí, que es importante el contar con un comité que se encargue de la valoración informática organizacional.

3.1.2. CONSIDERACIONES DE EQUIPO EXISTENTE EN LA ORGANIZACIÓN

Si una institución cuenta ya con algún equipamiento, se debería realizar un inventario de lo existente. Con este detalle, se puede elaborar una propuesta de reestructuración y/o reasignación de equipos. Esto estará condicionado al compromiso del personal y motivación que se haya dado al respecto, debido a que, puede generar un sentimiento de rechazo o descontento.

3.1.3. CONSIDERACIONES DE EQUIPO DEL MERCADO

La oferta de hardware ha evolucionado a pasos agigantados. Conocer hoy en día, cuales son las nuevas tecnologías puede ser una labor de muchas personas que se dediquen solo a investigarlas, desplazando, como puede ser obvio, a las labores y tareas administrativas que deban ejecutar.

Al realizar implementaciones informáticas, es imprescindible tener el conocimiento adecuado de la oferta existente. Antes de realizar adquisiciones de equipo que puede ser catalogado como clave en nuestra organización, es importante conocer que otras instituciones ya lo poseen y cual ha sido su rendimiento y grado de satisfacción.

Solo entonces se podrá tener una visión más clara de lo que se va a adquirir y del equipo en análisis.

3.2. ESTUDIO DE SOFTWARE

En muchas áreas de aplicación del software, a menudo es más rentable adquirir el software de computadora que desarrollarlo. Los gestores de ingeniería del software se enfrentan con la decisión de desarrollar o comprar, que se puede complicar aún más con las opciones de adquisición:

- El software se puede comprar (con licencia) ya desarrollado,
- Se pueden adquirir componentes de software y entonces modificarse e integrarse para cumplir las necesidades específicas; o
- El software puede ser construido de forma personalizada por una empresa externa o por equipo interno, para cumplir las especificaciones de la organización.

Los pasos dados en la adquisición del software se definen según el sentido crítico del software que se va a comprar y el coste final, se pueden aplicar las directrices siguientes:

1. Desarrollo de una especificación para la función y rendimiento del software deseado. Definición de las características medibles, siempre que sea posible.
2. Estimación del costo interno de desarrollo y la fecha de entrega.
3. Selección de tres o cuatro aplicaciones candidatas que cumplan mejor las especificaciones.
4. Selección de componentes de software reutilizables que ayudarán en la construcción de la aplicación requerida.

5. Desarrollo de una matriz de comparación que permita una comparación una a una de las funciones clave. Alternativamente, el seguimiento de las pruebas de evaluación para comparar el software candidato.
6. Evaluación de cada paquete de software o componente según la calidad de productos anteriores, soporte del vendedor, dirección del producto, reputación y otros.
7. Contacto con otros usuarios de dicho software y petición de opiniones.

En el análisis final, la decisión de desarrollar-comprar se basa en las condiciones siguientes:

- * Fecha de entrega del producto de software.
- * Tiempo y facilidad de traslado para mantenimiento.
- * Costo de adquisición vs. costo de desarrollo interno del software.
- * Costo del soporte externo vs. costo del soporte interno.
- * Costo de modificaciones externo vs. costo interno de modificaciones.
- * Costo de traslados para mantenimiento externo vs. interno.

Subcontratación (outsourcing)

Las actividades de ingeniería del software pueden ser contratadas con un tercero quien hace el trabajo y que debería asegurar una alta calidad. El trabajo de software llevado dentro de la compañía se reduce a una actividad de gestión de contratos.

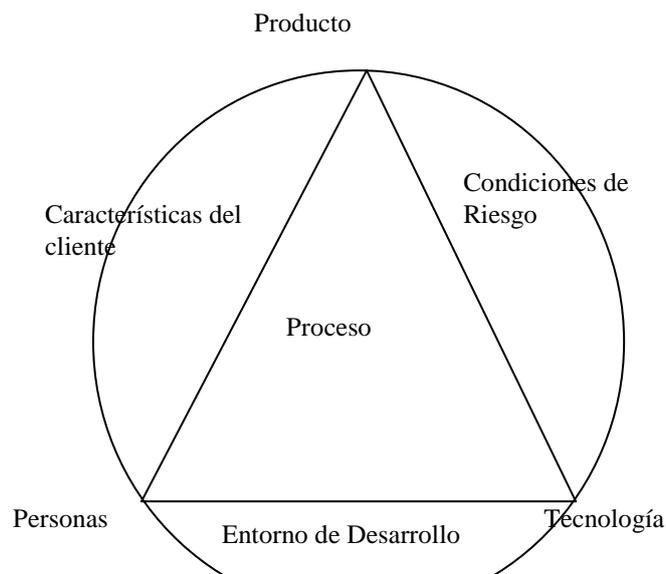
Con independencia de la amplitud del enfoque, la decisión de elegir “outsourcing” es a menudo financiera. Los ahorros de costos se pueden lograr reduciendo el número de personas y las instalaciones (por ejemplo: computadoras, infraestructura) necesarios para desarrollarlos internamente; en cambio, se pierde control sobre el software que necesita. Como el software es una tecnología que

diferencia sus sistemas, servicios y productos, una organización corre el riesgo de poner su destino en manos de un tercero.

3.2.1. EVALUACION DE SOFTWARE DEL MERCADO

Al evaluar el software que exista en el mercado, debemos entre otros aspectos ya mencionados anteriormente, tomar muy en cuenta los siguientes criterios:

- a. Precio del software
- b. Requerimientos del sistema:
 - ┌ Espacio en disco duro,
 - ┌ Necesidad de Ram,
 - ┌ Tipo de Procesador,
 - ┌ Sistema operativo a usar,
 - ┌ Tipo de capacidad gráfica,
 - ┌ Accesorios adicionales (unidad de cinta, unidad escritora óptica, equipo de digitalización, etc.)
- c. Incorporación con otras tecnologías (internet, intranet, multimedia, etc.)
- d. Capacidad de manejar acceso en red (cliente/servidor, centralizado, distribuido, etc.)
- e. Interconectividad con otras herramientas.
- f. Capacidad de ampliación del software.
- g. Portabilidad.



Al evaluar un tipo de software, es determinante medir la calidad de mismo. Como se muestra en la Figura 15, el proceso incluye tres elementos: Tecnología, personas, producto (bien o servicio). Estos están dentro de un entorno de riesgo, desarrollo y rasgos del cliente. Tanto los elementos del proceso, como las condiciones de su interacción, deben desenvolverse eficazmente para poder desarrollar un buen servicio u ofrecer un producto. En este ambiente, el software al ser parte de la tecnología, toma el papel de apoyar el proceso, mediante el cual las personas (usuarios del software), podrán determinar una demora o agilidad en el procesamiento.

En el Anexo 1 nos permitimos exponer la recopilación de varias herramientas CASE con su respectivo análisis.

3.2.2. EVALUACION PARA EL DESARROLLO DE SOFTWARE EN LA ORGANIZACIÓN

Al decidir si el desarrollo del software se lo realizará por parte de equipos propios de la institución, esta debe considerar varios parámetros claves, que permitan que los riesgos del proyecto sean reducidos al máximo para obtener un buen resultado. Podemos mencionar que uno de estos parámetros, es el convencimiento tácito que deben tener los directivos y además del apoyo incondicional a dicho proceso. Antes de tomar esta decisión podríamos mencionar que se debe estructurar una serie de requisitos, entre los que señalamos:

- ® Identificar el nivel estratégico que deben tener los sistemas.
- ® Identificar la magnitud de problemas a resolver en la Institución.
- ® Evaluar los recursos de hardware y software disponibles en la Institución y el medio.
- ® Evaluar el nivel del personal.
- ® Efectuar un estudio de costo-beneficio, definiendo metas a lograr.
- ® Elegir las herramientas apropiadas para la Institución.
- ® Establecer un programa de capacitación de personal de sistemas y usuarios
- ® Elegir una aplicación que reúna la mayor parte de los siguientes requisitos:
 - © Gran impacto de resultados.
 - © Disponibilidad de recursos.
 - © Mínimo nivel de riesgos.
 - © Máxima colaboración de usuarios.
 - © Tamaño reducido de solución.

La decisión, como es lógico, siempre la tomarán los directivos de la organización. Esta responde a decisiones de la política global que se haya implantado; sin embargo, es importante que antes de tomar tal decisión exista al menos mecanismos o enunciados técnicos de qué es lo más conviene a la institución.

3.2.2.1. TECNOLOGIAS DE DESARROLLO

Al evaluar las tecnologías de desarrollo, debe hacerse mucho hincapié en el grado de conocimiento y experiencia que tenga el equipo de desarrollo. Como quedó manifestado anteriormente en la sección 1.5.2.4, actualmente están vigentes las tecnologías de cuarta y quinta generación; y algunos todavía se inclinan por usar las de tercera generación.

Cada una es aplicable al desarrollo del software, la diferencia radica en las potencialidades que las generaciones vigentes poseen sobre las otras; de esta manera, el equipo de desarrollo puede centrarse en actividades claves tal como son el análisis y diseño.

3.2.2.2. LA CASE EN EL ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN

Los errores más comunes al desarrollar un sistema informático, se los comete más profundamente en las primeras fases del desarrollo. Por esto, depende como se realice el análisis y diseño del sistema para obtener resultados más favorables. Con una herramienta CASE, se puede dedicar más tiempo al análisis y diseño. Las especificaciones de diseño adoptan la forma de diagramas estructurados, que representan la estructura de los datos del programa, las entidades y las funciones de los procedimientos. Los prototipos rápidos reemplazan a los métodos manuales en la definición de los requerimientos de los usuarios eliminando de esta manera extensos documentos de especificaciones..

3.2.3. CAMBIOS EN EL PROCESO DE GENERACION DEL SOFTWARE

Durante los primeros años de la era de las computadoras, el software era un añadido, se diseñaba a medida para cada aplicación. Durante la segunda era (mitad de la década de los sesenta hasta finales de los setenta), se introdujeron la multiprogramación, sistemas multiusuario, sistemas de tiempo real, además del aparecimiento de varios productos de software. Aquí aparece la crisis del software debido a los fallos detectados en los códigos cuando los usuario se trasladaban a otro tipo de plataformas hardware, empezó entonces a tomar forma el mantenimiento de software. Para la tercera era (hasta finales de los ochenta) se incorporan los sistemas distribuidos, incorporación de conceptos de inteligencia. La cuarta era se caracteriza por proporcionar tecnologías orientadas a objetos, sistemas expertos, redes

neuronales, computación en paralelo y la incorporación masiva de las redes de computadoras.

En cuanto a las técnicas de desarrollo de software podemos decir que se encuentran vinculadas con el desarrollo de los lenguajes de programación. Inicialmente estas se dieron al nivel de lenguaje de máquina, posteriormente mediante una programación por lotes en la cual el traductor ejecuta instrucción por instrucción, lo cual generaba extensos programas. Los lenguajes de tercera generación hacen uso de las técnicas estructuradas, técnicas de orientación a objetos y de inteligencia artificial. Para los lenguajes de cuarta generación se caracterizan ya por tener soporte y generación inteligente propia de cada lenguaje y además de la incorporación de programación multiusuario, distribuida, multimedia, etc.

Cada una de estas generaciones tienen acoplado metodologías propias de cada época, ya que las metodologías de desarrollo de sistemas de información no hubiesen tenido el efecto deseado sin el acompañamiento de herramientas que puedan soportar las fases de cada metodología.

En lo que se refiere a la evolución y soporte de las herramientas CASE ya se lo trató en el apartado 1.5.

3.3. ANALISIS COSTO BENEFICIO

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas, la estimación del costo y del esfuerzo la planeación y desarrollo del software nunca será una ciencia exacta, son demasiadas las variables: Humanas, técnicas, de entorno, políticas, etc., que pueden afectar el costo final del proceso y el esfuerzo aplicado para desarrollarlo.

3.3.1. COSTO VS. CARACTERISTICAS TECNICAS

Una importante consideración en la selección de máquinas es su flexibilidad y capacidad. Si la máquina que se está considerando puede utilizarse con efectividad para varios trabajos que se realizan en la oficina.

La velocidad de procesamiento puede ser un factor contribuyente, pero esto no debe ser el factor decisivo.

Además, existe el factor de los valores estéticos; la apariencia resultante de la oficina al contar con lo último y lo mejor en máquinas y equipo de oficina. Tales valores tienen un lugar, ya que las máquinas y el equipo de oficinas no sólo representan un medio físico de ayudar a los empleados a desahogar su trabajo, sino que también sirven como estímulo mental. El proporcionar la unidad apropiada no sólo forma una actitud positiva y de cooperación sino que ayuda a colocar al empleado en el marco mental apropiado para que trabaje con eficiencia.

3.3.2. COSTO VS. PRODUCTIVIDAD

En la mayoría de los casos se hace la adquisición de equipo sobre la base del incremento en la productividad, expresada por costos más bajos, mayor volumen, mejor calidad o ahorro esperado. Los factores que determinan la productividad de la máquina por lo general, son:

- ↳ El precio total corriente, incluyendo entrega e instalación;
- ↳ Si se trata de un reemplazo, la marca, el modelo, el tipo y la condición de la unidad reemplazada y su probable valor de mercado
- ↳ El porcentaje de tiempo de trabajo futuro de la unidad
- ↳ El porcentaje de devolución de la inversión para la compañía

3.3.3. COSTO VS. BASE DE LA ADQUISICION

El usuario de la máquina dispone de cuatro opciones:

- 1) Comprar: el cual incluye un gasto de capital además de un gasto anual por operación y mantenimiento.
- 2) Rentar a corto plazo: Implica una erogación menor inicial, con gastos estándar de operación.
- 3) Rentar a largo plazo: La renta a largo plazo representa un mínimo del 75% de la vida útil de la máquina, las rentas varían.
- 4) Utilizar un servicio de máquina: Los cobros varían de acuerdo al tipo de máquina y la cantidad de trabajo.

3.3.4. COSTO VS. PRACTICA ADMINISTRATIVA

La depreciación y el mantenimiento son factores administrativos de importancia que afectan al costo de una máquina de oficina. En la depreciación, se debe remitir a las prácticas contables que siga una institución.

Las máquinas y el equipo de oficina son activos; y después de determinado periodo se las cancela debido a la depreciación. El periodo de depreciación de una computadora varía entre 3 a 5 años. Sin embargo, debido a los avances tecnológicos este período puede ser mucho menor, todo dependerá del sitio en el cual esté asignada y tipo de trabajo que realice.

El porcentaje usado en el periodo puede calcularse por varios métodos. Son comunes la línea recta y la suma de los dígitos.

Todas las máquinas y equipo de oficina requieren una atención periódica para conservarlas en una condición satisfactoria. Debería aplicarse mayormente el mantenimiento preventivo en vez del correctivo.

3.3.5. COSTO VS. CONSIDERACIONES DEL PERSONAL

La instalación de las máquinas y del equipo de oficina cambia las necesidades de personal, tanto con respecto al número de empleados como del nivel de su especialización; por lo tanto, debe tenerse en cuenta el problema de transferir, reducir y entrenar al personal de la organización.

La preferencia del empleado también es de gran importancia. El elemento humano es vital para determinar si el equipo está adecuadamente operado o utilizado.

Como quedó manifestado, son demasiadas las variables que intervienen para realizar un análisis de costo beneficio. En este capítulo, se ha presentado un esquema general de consideraciones que deben tenerse en cuenta a la hora de hacer una evaluación de conveniencia para la organización. Sin embargo; no siendo un recetario, podemos establecer algunos lineamientos generales para decidir en primera instancia si el costo de la planeación y desarrollo informático, será retribuido mediante los beneficios que podemos alcanzar; así:

a) Consideraciones sobre planificación:

1. Establecer un presupuesto aproximado basándose en técnicas de estimación de proyectos.
2. Analizar el sistema de planificación a través de redes activas u otro método.
3. Determinar que procesos agregan valor a los bienes o servicios que la organización brinda al entorno.
4. Calcular y evaluar los factores de productividad; es decir, cuánto se demora en ser ejecutada una orden sobre el tiempo total de estar en el proceso.
5. Medir el grado de aceptación de la persona que usa el sistema (cliente de la organización).

6. Medir el grado de eficacia (hacer las operaciones correctas) y el grado de eficiencia (hacer las operaciones bien) del personal encargado de los procesos de la organización.

b) Consideraciones sobre equipamiento:

1. Realizar un inventario detallado.
2. Revisar los recursos que se necesitan en cada proceso a modificar y/o implementar.
3. Reasignar los recursos a los procesos, en donde sea factible de hacerlo.
4. Evaluar rendimiento de equipos similares en la propia organización y en otras afines.
5. Estudiar la plataforma que se desea modificar, actualizar y/o implementar; basándose en la estrategia planificada.
6. Estudiar y revisar todas las posibilidades de características técnicas del equipamiento necesario para la implementación de los procesos.
7. Analizar posibilidades de: compra, contratación, renta; en los procesos que de acuerdo a cada caso lo permita realizar.
8. Realizar diagnósticos sobre conocimientos de los usuarios que serán afectados.
9. Analizar el volumen de carga de trabajo en cada proceso y equiparlo con el equipo asignado.

c) Consideraciones sobre el software:

1. Detallar todos los procesos que serán automatizados, en el cual se haga constar fechas probables de funcionamiento del mismo.
2. Evaluar las condiciones del personal existente para desarrollo de software.
3. Evaluar las herramientas con las que cuenta la organización para desarrollar sistemas informáticos.

4. Mediante algunas métricas de software, estimar costos de desarrollo en la propia organización.
5. Analizar posibilidades de: compra, contratación a terceros; en el cual se realice un análisis de los requerimientos de mantenimiento posterior, posibles modificaciones en el software, tiempo de entrega de terceros.
6. En caso de asociación con terceros, realizar una proyección de incremento de costos por servicios.
7. Revisar cual es el requerimiento de portabilidad e integración; entre plataformas, procesos, software.
8. Analizar los niveles de aceptación del usuario, frente a un software de terceros.
9. Comparar costos de asesoría y apoyo in situ.
10. Establecer estrategias de métricas técnicas del software, que permitan mediar la calidad del mismo.

d) Consideraciones sobre el personal:

1. Realizar un estudio sobre el nivel de conocimiento tecnológico del personal.
2. Evaluar el impacto que traerá consigo la implantación de nuevas tecnologías.
3. Estimar el impacto de posibilidades de reducción de personal.
4. Definir que cambios internos se necesitan para la implantación de nuevas tecnologías.
5. Proyectar el nivel de especialización de cada miembro de la organización y el entrenamiento respectivo.
6. Determinar, implicaciones económicas por efecto del mayor nivel de conocimiento del personal.

CAPITULO IV

4. CONTROLES EN LA PLANIFICACION Y DESARROLLO INFORMATICOS

4.1. IMPORTANCIA DE CONTROLES ESPECIFICOS

La última fase de la planeación se ocupa de llevar a cabo las decisiones hechas en las fases anteriores y controlar su implementación.

El control es una actividad de comprobación que se ejerce sobre los procesos para determinar su grado de aceptación y verificar que se cumplan con las regulaciones, especificaciones y normas establecidas.

En esta tarea se puede implementar flujos de tipo PERT-CPM , en donde se pueden establecer e identificar las actividades requeridas, relaciones entre ellas y el tiempo destinado a cada una. Se puede hacer una ampliación indicando los responsables en la ejecución y además de las necesidades de recursos para dicha actividad. De esta manera se puede controlar aún mejor la fase de desarrollo e implementación.

4.2. COMITES ORGANIZACIONALES

Los comités son los que someten las disposiciones que van a dictarse, al acuerdo de cuerpos de individuos a cuyo cargo están la dirección, la administración o la vigilancia de una empresa o de una institución.²²

Los comités se pueden dividir en organizacionales, ejecutivos, de vigilancia, consultivos, deliberativos; cada uno de los cuales en función de los niveles de la organización tendrán su respectivo campo de acción, responsabilidades y trabajos específicos asignados a ellos.

4.2.1. IMPORTANCIA DE COMITES

La importancia radica en que comparten la responsabilidad de las órdenes, ya que éstas se dictan conforme al acuerdo que se toma de entre miembros.

Permite la participación de varias personas en la realización de la política de la institución y de las decisiones de funcionamiento y en la responsabilidad directa de la realización del trabajo. Evitan de esta manera la personalización o individualización del trabajo.

²² Gómez Guillermo. *Planeación y Organización de Empresas*. Pg. 203

4.2.2. RESPONSABILIDADES

Dentro de los comités, y de acuerdo al nivel de jerarquía del mismo, pueden establecerse responsabilidades como:

- 4 Estudiar y resolver los problemas de la organización.
- 4 Revisión, coordinación de la actividad individual.
- 4 Verificar o inspeccionar las labores de los funcionarios o empleados.
- 4 Dar sugerencias que le son consultados por autoridades u otros consejos.
- 4 Dictar normas, manuales, procedimientos de trabajo.
- 4 Proponer mejoras en el desempeño de las funciones, entre otras.

4.2.3. SELECCIÓN DE SUS MIEMBROS

Es importante que aparte de los encargados de la elaboración del proyecto, se incorporen las personas que se verán beneficiadas. Dichos miembros deberán tener un carácter consultivo, porque depende en gran medida de las orientaciones y experiencia que ellos tengan, para la consecución de un buen modelo de requerimientos y necesidades.

4.2.4. DEPARTAMENTO DE AUDITORIA

La palabra auditoría proviene del latín “auditorius”, y de esta proviene la palabra auditor, que se refiere a todo aquel que tiene la virtud de oír.

Por ende, debido a su importancia en el funcionamiento de una organización, existe la Auditoría Informática.

El término de Auditoría se ha empleado incorrectamente con frecuencia ya que se ha considerado como una evaluación cuyo único fin es detectar errores y señalar fallas.

El concepto de auditoría es mucho más que esto. Es un examen crítico que se realiza con el fin de evaluar la eficacia y eficiencia de una sección, un organismo, una entidad, etc.

ALTA	Sobrevive	Prospera
EFICACIA	Muere lentamente	Muere rápidamente
BAJA		
	EFICIENCIA	ALTA

La esencia de la Reingeniería de Procesos. Peppard. PG 76

Figura 16 Matriz eficacia / eficiencia según Ducker

La Figura 16, muestra que resultados obtiene una organización, cuando el rendimiento de sus procesos en general, se miden en función de la eficiencia y la eficacia.

Los principales objetivos que constituyen a la Auditoría Informática son el control de la función informática, el análisis de la eficiencia de los Sistemas Informáticos, la verificación del cumplimiento de la normativa general de la organización y la revisión de la eficaz gestión de los recursos materiales y humanos informáticos.

El auditor informático a de velar por la correcta utilización de los amplios recursos que la organización pone en juego para disponer de un eficiente y eficaz Sistema de Información.

Claro está, que para la realización de una auditoría informática eficaz, se debe entender a la organización en su más amplio sentido, ya que una Universidad, un Ministerio, un Hospital o cualquier tipo de organizaciones públicas o privadas utilizan la informática para gestionar sus "negocios" de forma rápida y eficiente con el fin de obtener beneficios económicos, de costes y/o sociales.

4.2.5. RELACION ENTRE COMITÉ ORGANIZACIONAL Y DEPARTAMENTO DE AUDITORIA.

El comité organizacional que se integre para el desarrollo informático, muchas de las veces, es de carácter operativo-técnico; es decir, se encarga de elaborar planes y diseños arquitectónicos de los sistemas por implementar en la organización. Por su parte, al ser función de Auditoría Informática la de evaluar la eficacia y eficiencia de los sistemas, el cumplimiento de la normatividad del aseguramiento de calidad para Sistemas Informáticos; es importante que estos se realicen desde el mismo proceso de planeación. De ahí que, el cruzar información en la planeación y auditar los modelos resulta un paso muy importante para detectar errores y mejorar procesos en la fase de análisis y diseño.

4.3. CONTROLES EN EL PROCESO DE PLANEACION INFORMATICA

En el desarrollo de la planeación, las actividades del equipo de sistemas de información deberán planearse de manera que los objetivos del equipo estén de acuerdo con las metas de la organización, de tal manera que se debe verificar que estas sean coherentes y estén en concordancia. A parte de los objetivos y metas se debe establecer un equipo multidisciplinario que pueda interaccionar todos los

elementos de la organización, cambios organizacionales, avances tecnológicos y posibles requisitos gubernamentales.

En este proceso de control, es de vital importancia asegurarse de que los recursos que se necesiten para su posterior implementación y aplicación sean asignados adecuadamente y a tiempo; y además, que exista el compromiso de las autoridades a adquirir dicha implementación.

4.4. CONTROLES EN LA APLICACION DE LA PLANEACION

Cada organización debe establecer o adoptar una metodología para el desarrollo del proyecto y asignar responsabilidades a cada fase que así lo determine. Se debe además evaluar si su división en varias fases es coherente, y si cada fase da lugar a un resultado cuantificable.

La asignación de responsabilidades y de tiempo para cada equipo así como de recursos pueden afectar a una buena marcha del desarrollo del proyecto.

Es importante también el que se establezca claramente cual es la naturaleza y alcance del proyecto. En este sentido, la documentación que se deba manejar debe ser concisa y estandarizada, de tal manera, que no se realicen trabajos dispersos en un mismo proyecto.

También es necesario, el que se incorpore documentación apropiada que pueda ser utilizada como pistas de auditoría en los diseños e implementaciones.

El control es una parte de la planeación, no pasos subsiguientes. Constituyen la culminación de un ciclo y el inicio de otro.

4.5. CONTROLES EN EL PROCESO DE DESARROLLO INFORMATICO

El objetivo es asegurarse de que una vez concluido todo el desarrollo o parte de él, el sistema funcione conforme a las especificaciones funcionales, a fin de que el usuario tenga la suficiente información para su manejo, operación y aceptación. Las revisiones se efectúan en forma paralela desde el análisis hasta la programación y cumple entre sus objetivos:

- ☑ Identificar inexactitudes, ambigüedades y omisiones en las especificaciones.
- ☑ Descubrir errores, debilidades, omisiones antes de iniciar la codificación.
- ☑ Buscar la claridad, modularidad y verificar con base en las especificaciones.

4.6. IMPLEMENTACION DE VERIFICACION DE ERRORES EN LA HERRAMIENTA CASE

La verificación y comprobación de errores es una de las tareas de un CASE Workbench. La comprobación de errores comienza en el mismo momento en que empieza la diagramación y esta debe ser de manera interactiva.

Al realizar el control de errores de forma interactiva, la calidad del software y las especificaciones reducen gran parte del tiempo que utilizará para efectuar las pruebas y correcciones del sistema final.

La forma de implementar, se la realiza a través de las reglas que cada metodología utiliza para especificar sus diagramas y procesos.

La verificación de errores puede ser de tipo, sintaxis, integridad, consistencia, descomposición funcional, familia de diagramas, rastreo de requerimientos. Cada uno de estos errores se los trató en el apartado 2.6.

4.7. ANALISIS DE LA LEGISLACION ECUATORIANA

Dentro de la legislación ecuatoriana, no existen normas establecidas para la utilización de metodologías, que permitan desarrollar e implantar planeaciones informáticas. Con la eliminación y limitaciones de algunos organismos estatales, entre los que podemos destacar al Consejo Nacional de Planificación, Consejo Nacional de Ciencia y Tecnología, Dirección Nacional de Informática, entre otras, la planificación para las instituciones públicas prácticamente es casi inexistente. Los intentos de planificación son cambiados constantemente debido al cambio de quienes rigen las políticas en el Ecuador. Para las instituciones privadas, estas se rigen de acuerdo a estándares internacionales de certificación.

Sin embargo, vale resaltar que la Contraloría General del Estado tiene normas de auditoría sobre “Sistemas de Información Computarizados” en el Vademécum Legal Volumen III código 133 y 134; en el cual consta de algunos lineamientos generales para instituciones del Sector Público y Organismos Autónomos, y trata sobre aspectos como: sistemas nuevos, actualización, accesos, proyectos, organización, financiamiento, entre otros.

4.8. ANALISIS DE LA LEGISLACION INTERNACIONAL GENERALMENTE ACEPTADA

La legislación internacional se resume a la aplicación de Normas de Sistemas de Calidad, que en su mayor parte son dadas a través de la ISO.

Se entiende por Sistema de Calidad la estructura organizacional, las responsabilidades, los procedimientos, procesos y recursos que se requieren para la Gestión de Calidad.²³

²³ http://web.jet.es/arroniz/documentacion/ISO_9001_2000.htm

Es importante comprender que el Sistema es propio de la organización y por ende los requisitos a él son definidos, por la necesidad de la misma y no en forma arbitraria por la Norma. En éste contexto la definición de la política de Calidad establece la relación entre la estrategia de la organización y su visión de la Calidad. Esto debe a su vez corresponder con la estructura organizacional, las responsabilidades, procedimientos, procesos y recursos que se definan para el Sistema de Calidad.

Entre las normas que se destacan tenemos:

Sistemas de Calidad ISO 9001: Modelo para el aseguramiento de la calidad en el diseño, desarrollo, producción, instalación y servicio.

Es la Norma más completa de las tres normas contractuales y fue diseñada para empresas que diseñan, producen y venden productos o servicios. Está estructurada en un total de 20 subcapítulos. Estos son:

1. Responsabilidad de la Gerencia
2. Sistema de Calidad
3. Revisión del contrato
4. Control del diseño
5. Control de documentos y datos
6. Adquisiciones
7. Control de producto suministrado por el cliente
8. Identificación y trazabilidad del producto
9. Control de proceso
10. Inspección y ensayo
11. Control del equipo de inspección, medición y ensayo
12. Condición de inspección y ensayo
13. Control de producto no conforme
14. Acciones correctiva y preventiva
15. Manipulación, almacenamiento, envasado, preservación y despacho

16. Control de registros de calidad
17. Auditorías internas de calidad
18. Capacitación y entrenamiento
19. Servicios
20. Técnicas estadísticas

Sistemas de Calidad ISO 9002: Modelo para el aseguramiento de la Calidad en la producción, instalación y servicio.

La ISO 9002 es la segunda norma contractual. A diferencia de la Norma ISO 9001 la ISO 9002 fue diseñada para empresas que no diseñan sus productos o servicios. En gran medida es una versión de la ISO 9001 sin la inclusión del subcapítulo 4 que trata del control del diseño. Está estructurada en un total de 19 subcapítulos.

Sistemas de Calidad ISO 9003: Modelo para el aseguramiento de la Calidad en la instalación y ensayos finales.

La Norma ISO 9003 es la tercera de la norma contractual. Es básicamente una norma que regula solo el Control de Calidad y se deriva directamente de las normas militares norteamericanas de los años 40. En distintas ocasiones se trató de eliminar de la serie, dado que solo aplica a empresas que no producen ni dan servicio. Adicionalmente no contiene los capítulos de acciones correctivas y auditorías internas por lo cual le falta el proceso de mejoramiento continuo que las otras dos normas si exigen. Esta norma no aplica los subcapítulos 4, 6, 9 y 19.

También se tiene una serie de estándares en lo que respecta al desarrollo del software, tecnologías de información y sistemas informáticos:

a) *Nombre abreviado / código*

ISO/IEC 14102

Título: Tecnologías de la información. Líneas de orientación para la evaluación y selección de herramientas CASE

Naturaleza: Norma internacional

Ambito:

- Desarrollo de sistemas de información
- Entornos de ingeniería del software
- Entornos de desarrollo/ herramientas CASE

Las herramientas CASE representan una parte significativa de los utensilios tecnológicos utilizados en el desarrollo y en el mantenimiento del software. Esta norma fue desarrollada por ISO/IEC JTC1/SC7 para auxiliar a los responsables de estas áreas a:

- Identificar requisitos organizativos para las herramientas CASE;
- Construir correspondencias entre tales requisitos y las características de las herramientas CASE a evaluar;
- Seleccionar la herramienta CASE mas apropiada, basada en mediciones de esas características.

Campo de aplicación y alcance

Esta norma define los procesos en un conjunto estructurado según las características de las herramientas CASE, para su utilización en la evaluación técnica y elección de una de tales herramientas.

No debe utilizarse en caso de software de índole general (procesadores de texto, hojas de cálculo), ni en el caso de cuadros de referencia en el dominio de la ingeniería del software cuyo objetivo sea proveer mecanismos para la integración de datos, control y presentación.

Estructura

La norma se subdivide en dos grandes bloques: uno respetando los procesos de evaluación y elección de herramientas CASE y el otro describiendo las características funcionales de este tipo de herramientas.

A continuación se presenta el índice resumido de la norma:

- Entorno •Referencias normativas •Definiciones y acrónimos
- Evaluación y elección de herramientas CASE •Proceso de iniciación
- Proceso de estructuración •Proceso de evaluación •Proceso de elección
- Características de las herramientas CASE •Características funcionales relacionadas con los procesos del ciclo de vida •Características funcionales relacionadas con la utilización de las herramientas CASE
- Características genéricas de la calidad •Características genéricas no relacionadas con la calidad

Conexión con otras normas

Esta norma sigue el modelo de evaluación de productos de software descrito en ISO/IEC 9126 : Tecnologías de la información - Evaluación de los productos del software - Características de la calidad y líneas de orientación para su utilización, ampliando tales características en el caso de las herramientas CASE.

Utilización:

Se espera que esta norma conduzca a opciones más efectivas de herramientas CASE, así como a una mayor uniformidad en el modo como se describen las funciones y las características de estas herramientas.

Acceso:

Las normas ISO son documentos con copyright vendidos normalmente en forma impresa y, en ciertos casos, en microfilm o CD-ROM.

El precio de una norma, debido a gastos adicionales, difiere de un país. Por lo tanto, ISO indica los precios de sus publicaciones mediante un código alfabético en lugar de un valor monetario, código que depende del número de páginas de la norma. En cada organismo nacional de normalización existe una lista que establece la correspondencia entre los códigos y los precios en vigor en el país en cuestión. La presente norma, al pertenecer a ISO/IEC JTC1/SC7, tiene 55 páginas.

b) Nombre abreviado / código

IEEE 1016

IEEE 1016-1

Título:

IEEE 1016: Recommended Practice for Software Design Descriptions. Práctica recomendada para descripciones de diseño de software

IEEE 1016-1: Guide to Software Design Descriptions
Guía para las descripciones de diseño de software

Naturaleza:

Especificación técnica internacional de un consorcio industrial

Ambito:

- Desarrollo de sistemas de información
- Procesos del ciclo de vida del software

Campo de aplicación y alcance:

IEEE 1016 es una recomendación para la “descripción de un diseño de software”, entendiendo por tal la representación que sirve para comunicar cómo está diseñado el sistema. Especifica la información que una descripción de este tipo a de contener y la organización o esquema de presentación recomendada. Puede aplicarse a software de cualquier tipo (científico, comercial, etc.) destinado a funcionar en un ordenador. Su aplicación no está restringida por ninguna consideración relativa al tamaño, complejidad o carácter crítico del software. Tampoco está condicionada por la aplicación de una determinada metodología de diseño, gestión de configuraciones o control de la calidad, pues se supone que la información relativa a la calidad o los cambios en el diseño de la descripción será gestionada por otras actividades del proyecto. Así mismo, la norma no apoya ni se ve limitada por una técnica descriptiva particular, pudiéndose aplicar a documentos en papel, bases de datos automatizadas, lenguajes de descripción de diseños, etc.

La Guía 1016-1 se ocupa de la aplicación de los métodos y la documentación de diseño contemplados en la recomendación 1016-1987, utilizando diferentes métodos de diseño comunes como ilustración y concreción de sus conceptos.

Un nuevo proyecto de la IEEE se propone conseguir la armonización de la definición de contenidos entre los diversos productos de la línea de estándares IEEE en materia de ingeniería del software y de éstos con estándares internacionales relacionados, particularmente ISO/IEC 12207, estándar para la gestión del ciclo de vida del software, cuyos conceptos quedarán correlacionados con los de la IEEE 1016 a través de un anexo que se incorporará a ésta.

Estructura / partes de la norma

- Alcance/Referencias/Definiciones
- Consideraciones para la producción de una DDS (Descripción de un diseño de software)
- Información que a de contener una DDS
- Organización de una DDS

Conexión con otras normas

- IEEE 729, Glosario estándar de terminología de ingeniería de software.
- IEEE 730, Planes de aseguramiento de la calidad del software
- IEEE 828, Planes de gestión de configuraciones
- IEEE 830, Guía para la especificación de requisitos
- ISO/IEC 12207

c) Nombre abreviado / código

IEEE 1074

IEEE 1074-1

Título:

IEEE Standard for Developing Software Life Cycle Processes.
Estándar de IEEE para el desarrollo de procesos del ciclo de vida del software

Naturaleza:

Especificación técnica internacional de un consorcio industrial

Ambito

- Desarrollo de sistemas de información.
- Procesos del ciclo de vida del software

Campo de aplicación y alcance

El estándar se ocupa del conjunto de actividades constituyentes de los procesos que son preceptivas para el desarrollo y mantenimiento de software, tanto considerado de manera independiente, como formando parte de un sistema, así como la información de entrada y salida correspondiente. Están fuera de la cobertura de la norma otras actividades complementarias como las relativas a la adquisición o al desarrollo de hardware.

La revisión propuesta difiere de la original de 1995 en los siguientes aspectos:

Todas las actividades de planificación se reúnen ahora en un solo grupo que pasa a llamarse “Actividades de Planificación del Proyecto”.

- El proceso de verificación y validación (V&V) es sustituido por un nuevo grupo de “Actividades de Evaluación” que incluye acciones específicas de V&V como revisiones, inspecciones y pruebas.

El propósito general de la norma es contribuir a que las organizaciones que desarrollan o mantienen software puedan ajustarse mejor a los plazos y costes establecidos. Prevé que el usuario principal de esta especificación en un determinado proyecto de software será el arquitecto de procesos, el individuo cuya función es seleccionar un Modelo de Ciclo de Vida para el Software, crear el Ciclo de Vida del Software concreto del proyecto y asegurarse de que se sigue este ciclo durante el tiempo de vigencia de aquel.

Estructura / partes de la norma

IEEE 1074-1995 (Rev. de IEEE 1074-1991)

IEEE 1074.1-1995 GUÍA para el desarrollo de Procesos de Ciclo de Vida del Software

IEEE 1074 (proyecto de revisión de 1074-1995)

Conexión con otras normas

IEEE 0016

ISO/IEC 12207

d) Nombre abreviado / código

IEEE 1209:1993

Título:

IEEE 1209:1993 Recommended Practice for the Evaluation and Selection of CASE Tools

IEEE 1209:1993 Prácticas recomendadas para la evaluación y selección de herramientas CASE

Naturaleza

Especificación técnica internacional de un consorcio industrial

Ambito:

- Desarrollo de sistemas de información.
- Entornos de ingeniería de software.
- Entornos de desarrollo/ herramientas CASE

Campo de aplicación y alcance

IEEE 1209 normaliza la evaluación y selección de herramientas CASE que ofrecen soporte a los procesos de ingeniería de software, incluyendo gestión de proyectos, predesarrollo, desarrollo, posdesarrollo y procesos integrales. Los procesos de evaluación y selección recomendados se basan en la perspectiva del usuario de una herramienta CASE. Los criterios utilizados, por tanto, se ocupan únicamente de los aspectos de los programas visibles para el usuario, como entradas y salidas,

funciones, interfaces con el usuario e interfaces con programas externos documentadas.

El IEEE Estándar 1209 (Evaluation and Selection of CASE Tools) presenta un conjunto de líneas generales para evaluar herramientas CASE para «procesos de gestión de proyectos, procesos de predesarrollo, procesos de desarrollo, procesos de posdesarrollo y procesos integrales».

Conexión con otras normas

Reference Model for SEE Frameworks (ECMA TR/55)

ISO/IEC 15504 (SPICE)

CAPITULO V

5. IMPLEMENTACION DE UNA HERRAMIENTA CASE PARA LAS FASES DE ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION BASADO EN LA METODOLOGIA DE LA INGENIERIA DE LA INFORMACION

5.1. OBJETIVOS DE LA APLICACION CASE

La aplicación CASE desarrollada tiene por objetivos:

- 👍 Automatizar tareas de diseño.
- 👍 Automatizar los chequeos de errores de diagramación.
- 👍 Mantener en el diccionario de datos detalles de los elementos del diagrama.
- 👍 Mantener datos para ser reutilizables.
- 👍 Controlar el proyecto de cada usuario.
- 👍 Navegar por los diferentes niveles de un diagrama.

5.2. CATEGORIA DE LA HERRAMIENTA CASE

Las categorías, en las que se han dividido a las herramientas Case son:

Herramientas Front-end.

- ® Herramientas de planificación y definición de requisitos.
- ® Herramientas de análisis y diseño.
- ® Herramientas modelización y generación de prototipos.

Herramientas back-end.

- ® Compiladores.
- ® Generadores de estructura y código.

Gestión.

- ® Gestión de proyectos (dirección proyecto).
- ® Gestión de proyectos o ciclo de vida.
- ® Gestión requisitos.
- ® Gestión cambios y configuraciones.

Workbench

- ® Conjunto de herramientas integradas que interactúan unas con otras de forma consistente.

La herramienta Case diseñada esta ubicada como una herramienta de alto nivel, debido a que las herramientas de alto nivel, (U-CASE: Upper CASE - CASE superior o front-end), están orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño de sistemas.

5.3. PROCESOS QUE SOPORTA LA APLICACION CASE

Una Herramienta Case puede tener las siguientes capacidades:

- ✓ Diccionario de Datos.
- ✓ Verificación de errores.

- ✓ Diagramación Automática
- ✓ Generador de código
- ✓ Capacidad para desarrollo Cliente / servidor y desarrollar sistemas para ser ejecutados en un ambiente cliente / servidor.
- ✓ Componentes Web
- ✓ Componentes multimedia, etc.

La aplicación desarrollada, soporta:

- ✓ Diccionario de Datos.
- ✓ Verificación de errores.
- ✓ Diagramación Automática
- ✓ Control del proyecto y niveles

5.4. COMPONENTES Y DESCRIPCION DEL SISTEMA CASE A IMPLEMENTAR

5.4.1. DICCIONARIO DE DATOS

En el diccionario de datos es un conjunto de definiciones de todos los datos que aparecen en un diagrama de flujo de datos. Este contendrá:

-  El nombre y el sinónimo (alias) del dato.
-  Descripción del contenido del dato.
-  Los datos elementales que se relacionan con el término.
-  El rango permitido del dato.
-  Elementos u objetos con los cuales se interrelaciona.
-  La longitud disponible en caracteres.
-  Información adicional

5.4.2. BANCO DE TRABAJO

Para muchos autores, Banco de Trabajo, Diccionario de Datos, Repositorio; son palabras que definen la misma operación dentro de un entorno CASE. De tal manera que creemos que esta semántica es la más acertada.

5.4.3. DIAGRAMACION AUTOMATICA

Los diagramas que serán generados son:

Diagrama de Flujo de Datos

Notación:



Entidad: Un Productor o consumidor de información



Proceso: Un transformador de información



Flujo de Datos: La cabeza de la flecha indica la dirección del flujo de datos



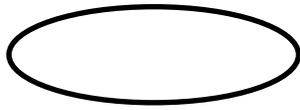
Almacén de Datos: Un depósito de datos que se va a almacenar para uso por uno o varios procesos.

Diagrama Entidad Relación

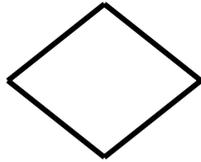
Notación:



Entidades.



Atributos.



Relaciones entre entidades.



Conectores de atributos a entidades y entidades a relaciones.

Relaciones:



Una a una



Una a muchas



Muchas a una



Muchas a muchas

5.4.4. PROCESOS DE VERIFICACION

Los procesos de verificación de errores que se aplicarán son:

Verificación de errores de sintaxis y tipo.

Verificación de consistencia e integridad.

5.5. DESARROLLO Y CODIFICACION DE LOS COMPONENTES DEL SISTEMA CASE

La Metodología de la Ingeniería de la Información de Martin, se centra en la información y el modelo de la empresa a alto nivel. Este modelado a alto nivel, se consigue a través de realizar una planificación de la estrategia de la información.

La planificación de la estrategia de la información ve a todo el negocio como una entidad y aísla los dominios del negocio importantes para la totalidad de la empresa. PEI define los objetos de datos visibles a nivel empresa, sus relaciones y cómo fluyen entre los dominios del negocio.²⁴ En este estadio (estadio 1) se comienza con la creación de un plan de los sistemas de negocio en el que se definen los objetivos del negocio para los próximos cinco a diez años.²⁵ Junto con el plan estratégico se construyen los modelos de alto nivel (funciones básicas y la estructura de la organización) y datos de la empresa (relaciones existentes de cada entidad).

El concepto de dominio, se encarga de realizar un Análisis del Area de Negocio (Estadio 2 de la Metodología). Las funciones definidas anteriormente permite dividir a la empresa en áreas lógicas (dominios); con lo cual se definirán y construirán los sistemas de información necesarios.

En la definición y diseño del sistema (estadio 3), se ocupa de las consideraciones del sistema lógico.²⁶ Es en este estadio que la metodología de la Ingeniería de la Información hace un salto hacia la Ingeniería del Software y se transforma en metodología similar de diseño y desarrollo de sistemas que abarca además al estadio 4 (construcción). Utiliza para esto: diagramas de flujo, de acción, de descomposición, de estructura, etc.

Con estos breves antecedentes; la aplicación realizada se centra a una parte del estadio 3 de la metodología de la Ingeniería de la Información. Es decir, la aplicación servirá para apoyar al análisis y a las tareas de diseño de sistemas, los

²⁴ Pressman Roger. *Ingeniería del Software. Un enfoque práctico*. Pg. 161

²⁵ McClure Carma. *CASE La Automatización del Software*. Pg. 153

²⁶ McClure Carma. *CASE La Automatización del Software*. Pg. 157

cuales incluye Diagrama de Flujo de Dato, Diagrama de Entidad Relación, Diccionario de Datos y Verificación de Errores.

Conceptos De Diseño En La Aplicación

La implementación de los componentes del Sistema Case se los realizó de la siguiente manera:

EL BANCO DE TRABAJO: Un conjunto integrado de Herramientas CASE diseñados para trabajar conjuntamente y automatizar (o proporcionar ayuda automatizada) totalmente el ciclo de vida del software, que incluye el análisis, el diseño, la codificación y la prueba.²⁷

Dado este concepto, la aplicación realizada alcanza la fase de diseño y permite de manera conjunta trabajar en:

- a) Control del proyecto y niveles (Child)
- b) Capacidad gráfica de diagramación
- c) Comprobación de errores
- d) Depósito de Información

Control del proyecto y niveles

Al plantearse la construcción de un diagrama, así sea este uno solo, necesariamente deberá ser visto como un proyecto a desarrollar. La definición de proyecto, en esta aplicación es: conjunto de uno o más diagramas, uno o más diccionarios, que permiten mantener información acerca del análisis de un sistema y su correspondiente diseño.

²⁷ McClure Carma. *CASE La Automatización del Software*. Pg. 36

De esta manera, cada sistema que se desee analizar y diseñar, deberá tener su propia identificación y nomenclatura, que en este caso el usuario deberá ser quien proporcione estos datos.

En el control de niveles subsiguientes, la aplicación es la responsable de administrar dichos niveles, tanto en manera ascendente (child de elementos del diagrama que lo posean) como descendente (parent de elementos).

Una vez realizado un diagrama child, este puede ser visualizado independientemente; situación ante la cual, no pierde vínculo permanente con todo el proyecto, sino momentáneo mientras el usuario lo requiera. Sin embargo, esto no quita, que a partir de ahí, el usuario pueda seguir generando más childs que sean necesarios. El momento que abre el proyecto respectivo, estos childs, se vinculan automáticamente a la totalidad del proyecto.

Capacidad gráfica de diagramación

Los diagramas que se pueden realizar son: Diagrama de Flujo de Datos y Diagrama Entidad Relación, siguiendo las reglas convencionales de estos tipos de diagramas las cuales son verificadas de manera interactiva.

En ambos casos de diagramas, la graficación se muestra en un diagrama en árbol, a partir del cual se puede realizar las respectivas modificaciones.

Comprobación de errores

Para un mejor control de errores, esta herramienta establece primero que debe ingresarse los datos a través de las opciones correspondientes.

Los errores que se verifica en la realización de los distintos diagramas son:

- a) Verificación de errores de sintaxis y tipo
- b) Verificación de consistencia e integridad

Verificación de errores de sintaxis y tipo.-

La verificación de errores de tipo, permite eliminar ambigüedades y, además, esta se encarga de examinar la construcción de los diferentes diagramas. En este caso, cada símbolo utiliza exclusivamente su respectiva representación. Además, se encuentra estandarizado el uso de la simbología de tal manera que símbolos equivalentes entre diagramas tienen la misma representación. Por ejemplo: La siguiente simbología, que representa a una entidad, se utiliza tanto para el diagrama de flujo de datos como para el de entidad relación.



En la comprobación de sintaxis, se determinan prácticamente las reglas que se aplican para la elaboración de diagramas, entre los cuales podemos mencionar:

Cada proceso debe tener al menos un flujo de datos de entrada y un flujo de datos de salida.

Dos o más almacenes de datos (registros de información), no pueden conectarse directamente. Deben pasar siempre por un proceso de datos.

Una entidad, al ser considerado como un elemento del cual viene o se entrega información o procesos, no necesariamente deberá tener un flujo de datos entrante, pero sí un flujo de datos saliente conectado a un proceso.

Verificación de consistencia e integridad.-

El propósito de esta comprobación es determinar si el diagrama a realizar tiene toda la información necesaria y suficiente para asegurar que él está completo. Esta comprobación se la realiza activamente, en forma interna, sin necesidad que el usuario realice una orden de verificación.

Como normas de consistencia e integridad podemos mencionar:

- a) Todo elemento debe ser etiquetado
- b) Cada elemento debe tener un nombre completo y un alias, que en lo posible deberá ser corto.
- c) Para los diversos diccionarios, la información pedida, permitirá ser más consistente y claros en la definición de cada elemento.
- d) Realizar un control de las explosiones de los objetos que lo permitan.
- e) Determinar si los elementos del diagrama están adecuadamente conectados o si es posible que falte alguna conexión.
- f) Cada elemento del diagrama debe tener conexión, no pueden existir elementos aislados o desconectados de la totalidad del diagrama.
- g) Chequear cada conexión ilegal.
- h) Nombres de cada elemento deben ser identificadores únicos, es decir no pueden existir dos elementos con las mismas características (duplicación de elementos).
- i) Revisar la coherencia de información de un nivel a otro conocido como análisis de equilibrio o de conservación.

Estas normas, se las detecta automáticamente; es así, que al no etiquetar un elemento, no podrá ingresar el siguiente elemento. Las conexiones ilegales

se las detecta el momento que se produce el error, no permitiendo realizar dicha conexión. Los elementos aislados, son vistos en el diagrama para visualizar el error y luego deben ser corregidos. Las explosiones de procesos, son controladas secuencialmente tanto hacia arriba (child), como hacia abajo (parent); sin embargo, esto no limita a que si se desea ver un diagrama exclusivo se lo pueda visualizar sólo dicho diagrama, pero sin conexión a la totalidad de cada proyecto de diagramas. En el caso de identificador que se desee ingresar duplicado, causará advertencia de existencia de ese elemento y confirmación de sobre escritura o redefinición del elemento, en cualquier caso se deberá tener en cuenta el objetivo de cada elemento. El proceso parent puede existir en el diagrama de nivel superior y con sus almacenes de datos asociados.

La comprobación del diagrama, genera un informe de errores, para tener presente que tipo de fallas se ha tenido en la construcción de los diagramas respectivos.

Depósito de información

El depósito de la información, mantiene metadatos, o datos sobre otros datos, como definiciones de objetos o elementos que se están utilizando para crear el diseño del sistema.²⁸ Este es la referencia para localizar la definición de cada elemento del diagrama. Este sirve para en una fase superior definir y construir las tablas de información del sistema, además, se encuentra incorporado, diccionarios o catálogos de elementos de Procesos de Datos, Almacenes de Datos, Atributos de Datos, Entidades; en los cuales se podrá ingresar información relativa a cada uno de estos elementos y tener un almacenamiento de información más completa acerca del diseño del sistema.

²⁸ Dorsey Paul, Koletske Peter. *Manual de Oracle Designer/2000*. Oracle Press. Pg. 23

Este depósito, el momento de su visualización está integrado en un pequeño entorno de edición básica que permitirá transportar el reporte a formato RTF o a formato DOC.

El manual de usuario es presentado en el Anexo 2.

El manual técnico consta en el Anexo 3.

El código fuente, desarrollado en Visual Basic 6.0, se encuentra disponible en el CD anexo; así como también toda la documentación aquí presentada..

CAPITULO VI

6. VERIFICACION DE HIPOTESIS Y CONCLUSIONES

VERIFICACION DE HIPOTESIS

- De acuerdo a la primera hipótesis que se plantea: Aplicando correctamente los métodos de la Ingeniería de la Información a los procesos de planificación y desarrollo informático se obtendrán mejores técnicas en la construcción de Sistemas de Información; se puede establecer que al enfocar a los Sistemas de Información como un elemento más de la Planeación Estratégica en una Institución; estos pueden aportar de mejor manera al plan institucional, así como también, el desarrollo informático tendrá una guía formal a seguir. Esto se logra, debido a que, la Ingeniería de la Información parte de un análisis de alto nivel; esto es, con la planificación estratégica de la información de toda la organización.
- Todo sistema de información necesita cumplir ciertas fases para su normal desarrollo y potencial éxito. De tal forma; que, utilizando

herramientas que aporten y apoyen a su desarrollo, se convierten en un mecanismo eficaz que permite dedicar mayor tiempo a la investigación y estudio de los procesos por automatizar; por lo que, se puede enfatizar que al utilizar una herramienta CASE se obtienen mejores resultados en el Análisis y Diseño de Sistemas de Información. Además, podemos hacer extensivo esta afirmación a otras fases de la Planeación y Desarrollo Informático. La incorporación de Herramientas I-CASE, que permitan automatizar y apoyar a otros procesos tanto de la planificación como del desarrollo de sistemas informáticos, conducirá a obtener mejoras significativas en la producción de nuevos procesos y/o sistemas automatizados.

CONCLUSIONES

- Para aplicar correctamente una herramienta Case, esta se debe fundamentar sobre la base de la arquitectura del entorno y ambiente de una organización, ya que en la tecnología Case existen aspectos del software, del hardware, de la metodología y de la gestión que deben ser estudiados y analizados, para que estas herramientas puedan cumplir efectivamente el rol que se desea.
- La aplicación adecuada y minuciosa de los procedimientos de la Metodología de la Ingeniería de la Información, permite obtener los resultados deseados, debido a que esta metodología enfoca a la información como el factor primario y fundamental de una organización. Visto de esa forma, la información se la entiende como un recurso corporativo de vital importancia para todo el contexto organizacional.
- La administración de la informática toma el rol de complementar e integrar a todos los departamentos de la organización. Esto se debe a

que la información es un activo valioso e indispensable para la empresa, ya que es necesaria para su funcionamiento correcto.

- La Planeación Estratégica y la Ingeniería de la Información establecen fases similares en el análisis de las organizaciones, la cual empieza por el nivel estratégico de la empresa.
- La implementación de un proceso de planeación, debe estar acompañada del apoyo de las autoridades de la organización; así como también, de la asignación adecuada de recursos.
- Al desarrollar software para la utilización de la organización, estos influyen sobre toda la estructura de la misma, por lo que deben estudiarse y analizar los cambios que se producirán.
- El uso de las herramientas CASE, ayudan a los diseñadores de software a centrar y mejorar su trabajo en la realización de análisis y diseños más eficientes.
- El uso correcto de las metodologías, permite a los equipos de planeación y desarrollo de sistemas tener mejores instrumentos que permiten reducir errores.
- La aplicación de los métodos de la Ingeniería de la Información a los procesos de planificación y desarrollo informático permiten obtener mejores resultados en la construcción de Sistemas de Información.
- Mediante el uso eficiente de herramientas CASE, se pueden obtener mejores resultados en el Análisis y Diseño de Sistemas de Información, reduciendo los errores y ambigüedades.

CAPITULO VII

7. RECOMENDACIONES

- Todo proceso que involucre la participación de personas, debe ser tratado conforme lo establece la Teoría General de Sistema para la Administración; además, siempre involucrará las expectativas que reúna cada persona dentro de la organización. Cambiar un proceso implica cambiar la forma en que las personas realizan su trabajo y por ende necesita de nuevas adaptaciones y formas de trabajo.
- Los Planes Informáticos, deben ser partes de la Planeación Estratégica Institucional, para de esta manera tratar a los Sistemas Informáticos como una política institucional; y, permitiendo de esta manera, el aprovechamiento y beneficios de la potencialidad de las tecnologías de la información.

- Las Herramientas Case, por su cobertura en las diversas metodologías de desarrollo de sistemas deben ser difundidas y estudiadas más profundamente; además, debido a su diversidad, las herramientas que brindan asistencia a la planeación estratégica deben ocupar el mismo grado de importancia..
- Antes de establecer planes de desarrollo informático, cada organización debería definir políticas de metodologías, control y evaluación, a las diferentes fases del desarrollo estratégico.
- El desarrollar aplicaciones Case, permite conocer más acerca de cómo funciona y se comporta un sistema de este tipo; sin embargo, sería oportuno desarrollar un Case que permita automatizar el proceso de planificación informática.
- Ningún documento esta terminado; por lo que, el presente estudio está abierto para ampliaciones y refutaciones.

ANEXOS

ANEXO 1

ANÁLISIS DE DIVERSAS HERRAMIENTAS CASE

Dentro de las herramientas CASE presentamos aquí, una recopilación de información acerca de varias herramientas existentes en el mercado. Debemos recalcar, que lo presentado aquí, es una recopilación de información:

HERRAMIENTAS CASE PARA EL ANÁLISIS Y DISEÑO ESTRUCTURADO

a) Características Generales

ERWIN 3.0

- Logic Works Inc.,
- Requerimientos del sistema: 10MB en disco duro,

16MB en RAM, procesador Intel 486, Pentium o SUN SPARC, plataformas Windows 3.x, Windows NT o Windows '95, Solaris 2.x.

Erwin es una herramienta para modelar, que ayuda a diseñar bases de datos de alto desempeño para cliente/servidor y *web/intranet*, así como aplicaciones de *data warehousing*. La herramienta Erwin no solo ayuda a diseñar modelos de datos lógicos, también construye automáticamente estructuras de datos físicos con la información del diagrama.

Cuando el modelo de datos esta listo para usarse, simplemente se selecciona el servidor donde se quiere construir la base de datos y se eligen las opciones de generación de esquema que se quieran incorporar. Erwin automáticamente construye la base de datos física, incluyendo todas las tablas, índices, procedimientos almacenados, *triggers* de integridad referencial y otros componentes necesarios para manejar exitosamente los datos usados en la organización.

ER/STUDIO 2.5

- Embarcadero Technologies, Inc.
- Requerimientos del sistema: 17 MB en disco duro, 32MB en RAM, procesador Pentium o mayor, plataformas Windows NT (3.51 y 4.0) o Windows '95.

Es una herramienta de modelado de datos y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas.

ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico,

sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

SYSTEM ARCHITECT 4.0

- Popkin Software Inc.,
- Requerimientos del sistema: 10MB en disco duro, 12MB en RAM, procesador 486 o mayor, plataformas Windows '95 o NT.

System Architect es una herramienta poderosa de modelado estructurado de datos, tiene la capacidad de identificar y clasificar personal para autorizar su entrada al sistema. Los usuarios de red trabajan en un diagrama de proyecto y una llave de registro de diccionario de datos. Soporta la mayoría de los paquetes de red incluyendo Novell, 3Com, Banyan, DecNet, LAN Manager, STARLAN y otras.

Define propiedades para cualquier entrada de diccionario, incluyendo definiciones, símbolos y diagramas. Construye ligas entre varios objetos del diccionario. Especifica y define requerimientos, planes de prueba, cambio de requerimientos, objetivos de negocios, metas y más. Especifica que símbolos o grupo de símbolos son afectados.

POWER DESIGNER 6.1

- Sybase, Inc.,
- Requerimientos del sistema: para Windows 95 o NT 3.51, procesador 486 o mayor, 12MB en RAM, monitor VGA, CD-ROM, 10 MB en disco duro. Para Windows 3.1: procesador 486 o mayor, 8MB en RAM, monitor VGA, CD-ROM, 10MB en disco duro.

Es una herramienta para crear bases de datos y aplicaciones cliente/servidor basadas o no en *Web*. Permite a los diseñadores de

aplicaciones complejas de cliente/servidor tener una descripción general de los procesos particulares para comprender mejor a la organización.

Exporta información del modelo físico y extiende atributos al diccionario de 4GL. Importa atributos extendidos de PowerBuilder. Soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse, Axiant y NS-DK.

b) Diagramas

ERWIN 3.0

Los diagramas de modelos de datos en Erwin se usan para generar o actualizar bases de datos relacionales, o sea, sistemas de almacenamiento de datos ordenados en tablas. Cuando se crea un diagrama Erwin, el modelo de la información se representa por entidades (gente, lugares y cosas), atributos (hechos acerca de una entidad, tales como nombre de la persona, dirección, edad, etc.), y relaciones entre entidades. Cada entidad corresponde a una tabla en la base de datos, con instancias de entidades que corresponden a los renglones de la tabla y atributos de entidades correspondientes a encabezados de columnas. Las relaciones, usadas por DBMS (*data base management system*) para ligar renglones de datos en tablas diferentes, están representadas como frases verbales en una línea conectando a dos entidades. Cuando se actualiza una base de datos física, Erwin automáticamente genera un *script* de definición de datos SQL, para crear tablas de bases de datos, incluyendo llaves, constraints y códigos *trigger* SQL para reforzar la integridad referencial entre tablas relacionadas.

ERSTUDIO 2.5

La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de

las líneas, los colores, tipos de letra, niveles de acercamiento y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El *Explorer Navigator* facilita el trabajo hasta con los diagramas más grandes. Se usa el *browser Explorer* para encontrar y seleccionar entidades.

SYSTEM ARCHITECT 4.0

Es posible ver los diagramas en las siguientes opciones: tamaño actual, página completa, área usada, o porcentaje de reducción. Permite la edición de un diagrama en cualquier modo de vista, seleccionar y mover objetos individualmente o usando el ratón para obtener la porción del diagrama que se desee, y cambiar el tamaño de objetos individuales proporcionalmente o no, usando el ratón. La herramienta *Leveling Automatically* nivela diagramas y usa un mecanismo simple para cambiar la herencia en cualquier dirección. Automáticamente crea *Decomposition Diagrams* (Diagramas descompuestos) de la herencia del *Data Flow Diagrams* (Diagramas de flujo de datos).

POWERDESIGNER 6.1

PowerDesigner cuenta con herramientas para la creación y control de diagramas como son: *Off-page Connector*; que representa los flujos de entradas y salidas en un proceso, *Business Rules* que define las reglas de uso para Procesos, Almacenamiento de datos, Entidades externas y Flujos de datos; y *CRUD Matrix*, que define el efecto de un proceso de datos en términos de Crear, Leer, Actualizar y Borrar operaciones (CRUD)

c) Esquema de la base de datos

ERWIN 3.0

Para diseñar un modelo de datos, Erwin proporciona propiedades específicas de DBMS y del servidor de editores que permiten definir nombres físicos y propiedades para las tablas, columnas y relaciones que genera cuando se crea el esquema de la base de datos.

Cuando Erwin crea un esquema de bases de datos, genera un *script* de cliente DDL (*data definition language*) usando la sintaxis correcta de SQL para el servidor seleccionado. Se puede ver el código que genera Erwin y, si se desea, se puede modificar antes de que se cree la base de datos.

Si el servidor soporta elementos avanzados como procedimientos almacenados y *triggers* de integridad referencial, Erwin proporciona editores de plantillas especiales y macros para guardar la información en tiempos determinados para acelerar la creación de estos objetos en el servidor. En varios servidores, se pueden hasta crear objetos de almacenamiento físico como espacios de tablas y segmentos de enrolamiento de Erwin así como especificar la ubicación y parámetros de almacenamiento para las tablas de bases de datos que Erwin genera en el servidor.

ERSTUDIO 2.5

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se desnormalizan los diseños físicos para su desempeño. ER/Studio mantiene ligas entre todos los niveles de su diseño por lo tanto puede mezclar cambios en cualquier dirección entre ellos. ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos.

SYSTEM ARCHITECT 4.0

System Architect proporciona todos los elementos para diseñar un nuevo sistema o modificar un sistema actual.

Es posible crear modelos lógicamente normalizados y modelos de datos físicamente desnormalizados usando el conjunto de herramientas de System Architect. También se puede crear un modelo conceptual de las entidades y especificar su relación con otras. Al avanzar el proyecto, se pueden incluir llaves primarias, atributos, reglas, *triggers* personalizados y cualquier otra información que se elija para mantenerla en el modelo.

Si se diseña un nuevo sistema usando un diccionario amplio de datos es posible especificar los requerimientos de los datos antes de comenzar el modelado, mientras se está construyendo el modelo, o después de haber completado el diseño lógico.

Si se está modificando un sistema existente es posible usar la ingeniería de reverso de SA para crear un diagrama de modelo de datos físicos para el sistema actual. SA crea automáticamente un diagrama entidad relación de un modelo de datos físico. Entonces se puede modificar el diagrama entidad relación, creando un modelo lógico normalizado del nuevo sistema. Una vez que se ha completado el diseño lógico, se pueden generar modelos físicos. Si se planea implementar una base de datos desnormalizada, se puede documentar el proceso de desnormalización usando diagramas *Local View* (Vista Local). SA mantiene ligas entre el modelo lógico, las vistas lógicas y el modelo físico; por lo tanto los cambios al modelo lógico se reflejan automáticamente en el modelo físico. Al final, se tienen dos modelos físicos separados: uno del sistema actual y otro del sistema propuesto. Para cualquier tipo de proyecto en el que se este trabajando, SA proporciona flexibilidad para completar el trabajo. Se pueden elegir modos de despliegue en cualquier tiempo durante el proceso

de diseño: conceptual, basado en llaves, totalmente atribuido, o despliegue físico. También, una vez que se ha completado el modelo lógico, se pueden ejecutar una serie de reglas revisadas y reportes de normalización para validar la integridad del diseño. Se prueba las Formas Normales: Primera, Segunda, Tercera y Boyce Codd.

POWERDESIGNER 6.1

Data Architect proporciona capacidades de modelado de datos tradicional, incluyendo diseño de bases de datos, generación, mantenimiento, ingeniería de reversa y documentación para arquitecturas de bases de datos. Permite que los diseñadores de bases de datos creen estructuras de datos flexibles, eficientes y efectivas para usar una ingeniería de aplicación de bases de datos.

También proporciona un diseño conceptual de modelo de datos, generación automática de modelo de datos, diseño de normalización física, sistema de manejo de bases de datos múltiples (DBMS) y soporte de herramientas de desarrollo, y elementos de reportes con presentación y calidad.

El diseño se realiza en dos niveles:

- ® Nivel conceptual: entidades, relaciones, dominios, tipos de datos conceptuales, identificadores y reglas de negocios. Basado en la notación de Ingeniería de Información de Martin.
- ® Nivel físico: tablas, columnas, dominios, llaves primarias, llaves foráneas, llaves alternadas, índices, vistas, parámetros de almacenamiento físico, reglas de negocios, *triggers* y procedimientos almacenados.

d) Código que genera el lenguaje

ERWIN 3.0

Erwin combina bases de datos *back-end* y desarrollo de aplicaciones *front-end* en un ambiente unificado. Tiene soporte para multi-clientes, Erwin genera formas de entrada de datos en Visual Basic, DataWindows de Power Builder y PROGRESS SmartObjects del mismo modelo de datos, logrando que los desarrolladores incorporen aplicaciones altamente productivas en tres de los ambientes de desarrollo de bases de datos. Erwin extiende el editor estándar Column Property Editor de tal forma que se pueden asignar rápidamente propiedades de columna del lado del cliente, tales como tipo de control por omisión. Despliega formato y reglas de validación de cliente para cada columna y genera formas de entrada de datos en uso y otros componentes de aplicación directamente del mismo modelo Erwin que crea la base de datos *back-end*. Para simplificar aún más el desarrollo de aplicaciones en Visual Basic, Logic Works también ofrece DataBOT, un robot de software avanzado que genera dinámicamente todo el código de acceso de datos SQL requeridos, permitiendo hasta que los programadores novatos creen rápidamente aplicaciones sofisticadas de bases de datos de alto desempeño en los ambientes actuales.

ERSTUDIO 2.5

Genera otros objetos de base de datos: vistas, procedimientos almacenados, defaults, reglas y tipos de datos de usuario, lo cual ayuda a la auto ordenación de tipos de objetos para eliminar errores de dependencia al construir la base de datos.

Tiene una opción para generar código fuente o para construir bases de datos. Soporte para crear bases de datos para Servidores SQL; y otra, para

incluir código SQL y verificar la creación de objetos. Además de la opción para incluir encabezados de comentarios.

SYSTEM ARCHITECT 4.0

Genera archivos de definición de almacenamiento de trabajo (.WKS) y sección de pantalla (.SCS) para implementaciones de COBOL. SA/PowerBuilder Link permite el intercambio de la información de diseño entre SA y PowerBuilder, incluyendo DDL y atributos extensos.

POWERDESIGNER 6.1

Mediante el incremento del modelo de la base de datos, AppModeler genera instantáneamente objetos, componentes *data-ware* y hasta aplicaciones básicas listas para ejecutarse inmediatamente en PowerBuilder, Power++, Visual Basic, Delphi y Web-based objects. El AppModeler permite a los desarrolladores: diseñar modelos de bases de datos físicas o crearlas instantáneamente a través de la ingeniería de reversa de bases de datos existentes, generar, documentar y mantener bases de datos, generar rápidamente objetos de aplicación y componentes de datos para PowerBuilder 4.0 y 5.0; Visual Basic 3.0, 4.0 y 5.0; Delphi 2.0; y el Web.

- Generación de objetos en PowerBuilder. Soporta todas las ediciones de PowerBuilder 4.0 y 5.0. Genera objetos personalizables de PowerBuilder y componentes basados en modelos de bases de datos físicos y plantillas que se encuentran dentro de las librerías de clases de su elección. Genera objetos ventana y ventana de datos basadas en tablas, vistas y relaciones de llaves primarias-foráneas. Genera y hace ingeniería de reverso a los atributos. Incluye plantillas personalizables para la librería PowerBuilder Foundation Class (PFC).

- Generación de objetos en Visual Basic. Soporta todas las ediciones de Visual Basic 3.0, 4.0 y 5.0. Incluye add-in de Visual Basic para la fácil manipulación de plantillas predeterminadas personalizables. Genera formas basadas en tablas, vistas y relaciones de llaves primarias-secundarias. Genera proyectos basados en modelos de propiedades. Genera controles tales como menús, listas, etc.

- Generación de objetos Delphi. Soporta todas las ediciones de Delphi 2.0. Incluye add-in de Delphi para una manipulación de plantillas personalizables predefinidas. Genera aplicaciones y objetos (proyectos, formas y controles) de tablas, columnas y referencias.

e) Ingeniería hacia delante y de reversa

ERWIN 3.0

Ingeniería hacia adelante

El desarrollo del modelo de Logic Works usa información específica en un modelo de datos Erwin para acelerar la creación y mantenimiento de soporte, migración y documentación de bases de datos relacionales. El proceso de generar una base de datos físicos de un modelo de datos lógico se llama ingeniería hacia delante (*forward-engineering*). Es posible llevar a cabo la ingeniería hacia delante en Erwin (que se llama operación de diagrama) mediante la generación automática de un esquema de base de datos física directamente del modelo de datos sin codificar laboriosamente las definiciones de datos necesarias en SQL y los *triggers*.

Ingeniería de reverso

La capacidad de ingeniería de reversa de Erwin, la cual puede inferir exitosamente las relaciones entre tablas, permite que se capture rápidamente la estructura de una base de datos existente y convertirla en un modelo lógico independiente del DBMS. Se puede usar el modelo de

datos para análisis detallado, se define sobre tiempo y lo distribuye como parte de la documentación requerida a través del ciclo de desarrollo. Si se desea migrar la base de datos existente de una plataforma a otra, Erwin puede hacer la ingeniería de reverso de la base de datos existente, crear un modelo de datos, modificar o agregar nuevos elementos según sea necesario y después construir la base de datos física en cualquier ambiente de servidor de los que soporta.

ERSTUDIO 2.5

Ingeniería hacia adelante

Una vez que se ha diseñado la base de datos, se puede construir o generar código fuente para todo o parte de los diseños de la base de datos. Propiamente hace la secuencia de la creación de tipos de objetos diferentes para asegurar eficiencia y construir bases de datos libres de errores. Actualiza una base de datos del diagrama. ER/Studio permite aplicar cambios de diseño del modelo de datos directamente a la base de datos. Cuando se comparan las diferencias entre los dos, formula una estrategia de alteración inteligente que implementa el diseño de las modificaciones mientras se preserva la tabla con los datos existentes, privilegios de objetos y dependencias en la base de datos.

Ingeniería de reversa

Cuenta con ingeniería de reverso, cuando necesite iniciar un trabajo de una base de datos existente, ER/Studio puede hacer una ingeniería de reverso al esquema completo para cualquier plataforma de bases de datos. La operación de la ingeniería de reverso extrae eficientemente definiciones de objetos y construye un modelo de datos gráfico.

SYSTEM ARCHITECT 4.0

Ingeniería hacia delante

Común a los anteriores.

Ingeniería de reversa

Genera y hace ingeniería de reverso en los *triggers*, reglas, defaults, tipos de datos definidos, dispositivos y bases de datos.

Elementos de la Ingeniería de reverso de SA:

- Captura datos del sistema de bases de datos más popular: Access, DB2, Informix, Oracle, SQL Server o SYBASE.
- Ingeniería de reverso de un archivo DDL, a través del manejador ODBC, o a través de la interfase directa del SQL Server.
- Genera diagramas de entidad relación que incluyen definiciones para llaves, no llaves y relaciones.
- Genera diagramas de modelo de datos físicos que incluyen definiciones para tablas, columnas y constantes.
- Crea llaves primarias, índices alternados y rutas de acceso de los índices en la estructura de la base de datos.
- Importa *triggers*, procedimientos almacenados, defaults y roles de definiciones de bases de datos existentes.
- Captura, modifica y regenera esquema usando SA RDE con el Schema Generator.
- Captura, modifica y regenera pantallas y menús de aplicaciones de Windows usando el SA RDE con el SA Screen Painter

POWERDESIGNER 6.1

Ingeniería hacia delante

Común a los anteriores.

Ingeniería de reversa

Visualiza estructuras de bases de datos existentes directamente vía ODBC o usando archivos de script DDL. Genera el modelo conceptual del modelo físico. Retro- documentación de bases de datos existentes.

Re-orientación de la base de datos existente a un DBMS diferente. Interfaz con herramientas de desarrollo: exporta información del modelo físico y extiende atributos al diccionario de 4GL, importa atributos extendidos de PowerBuilder, soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse.

f) Sincronización de la Base de Datos

ERWIN 3.0

El manejo de cambio comprensivo de Erwin y los elementos de generación de script de alteración facilita el almacenamiento del modelo de datos y la base de datos física de manera sincronizada. Durante la sincronización, Erwin desempeña una comparación comprensiva de todos los objetos físicos y lógicos, incluyendo definiciones de tablas y columnas, llaves, índices y parámetros de almacenamiento físico, resaltando cualquier discrepancia y permitiendo que se migren las definiciones de la base de datos modelo al modelo de la base de datos.

Si se exportan cambios de un modelo a una base de datos, Erwin genera *scripts* de alteración de SQL para actualizar el esquema de la base de datos. Erwin automáticamente revisa todas las dependencias del esquema y traduce y preserva los datos existentes cuando se actualiza el esquema, facilitando los cambios de diseño migrados para activar el desarrollo, probar y producir bases de datos.

Además de la sincronización del modelo de toda la base, Erwin soporta sincronización de modelo a modelo y modelo a *script*.

ERSTUDIO 2.5

Sincronización entre el diagrama físico y el lógico. Mezcla entre cualquier par de diagramas físicos para la misma plataforma de bases de datos.

Comparación lado a lado de las diferencias. El usuario puede decidir que diferencias mezclar o ignorar. Objetos reusables. Construir atributos reusables. Aplicarlos a atributos y columnas.

Propagación global de actualizaciones. Construir tipos de datos personalizables. Submodelado. Crear cualquier número de subvistas personalizables sobre un diagrama físico o lógico. Cualquier objeto puede existir en cualquier número de subvistas (relaciones de muchos a muchos entre objetos y subvistas).

Crear rápidamente subvistas eligiendo un área del diagrama. Control independiente sobre el despliegue de la subvista, incluyendo posición del objeto, colores y letras.

Utilidad de búsqueda rápida. Editores en tabla para evitar la necesidad de poner en modo cascada los diálogos. Diferenciación de color de llaves primarias y secundarias inherentes. Sombreado de cajas de entidad.

SYSTEM ARCHITECT 4.0

El generador de esquemas (Schema Generator) automatiza la creación y mantenimiento de esquemas para DBMS de SQL y 4GL. SA Schema Generator traduce modelos de datos (ER o diagramas físicos DM), y diagramas de clases orientadas a objetos en las definiciones de los esquemas para manejo de sistemas con múltiples bases de datos. La transferencia de esquemas en la DBMS se puede llevar a cabo en vivo, vía conexión ODBC, o a través del uso de un archivo. SA Schema Generator simplifica el desarrollo y asegura la integridad de los diseños permitiendo crear y mantener esquemas de bases de datos para una amplia variedad de administradores de sistemas de bases de datos SQL y no SQL. Cuando se genera un DBMS múltiple, los tipos de datos definidos en el modelo de

datos se mapean automáticamente a los tipos correctos del DBMS seleccionado. Se usa el SA Schema Generator para generar:

- DDL para la creación de tablas y mantenimiento desde el repositorio de SA.
- Definir Create Table o Alter Table, llaves primarias y llaves secundarias.
- Crear definiciones para reglas, procedimientos almacenados, defaults y mensajes.
- Triggers de integridad referencial.
- Usar tipos de datos definidos por el usuario.
- Definiciones de Disk Init, Disk Mirror, Add Segment y Create Database.
- Índices de rutas de acceso y llaves.
- Definiciones de tipos para C y libros de COBOL.
- El generador de esquema (Schema Generator) prueba las palabras reservadas de COBOL y automáticamente modifica con extensiones –XX.

POWERDESIGNER 6.1

Soporta más de 30 DBMS, incluyendo Sybase SQL Server, Oracle, Informix, Ingress, Sybase SQL Anywhere, Microsoft SQL Server, SQLBase, Progress, Access, Paradox, FoxPro, etc.

Creación directa de bases de datos vía ODBC o usando archivos de scripts DDL.

Elige la generación del modelo entero, ciertos submodelos, u objetos individuales.

Reglas de negocios definidas de usuario pueden integrarse en triggers y procedimientos almacenados.

Para el mantenimiento de la base de datos, incluye:

- Modelos de archivo.

- Comandos alterados generados para preservar datos contenidos.
- Mantenimiento vía ODBC o a través de archivos de scripts DLL.
- Elegir para generar el modelo entero, ciertos submodelos, u objetos individuales.

g) Diseño

ERWIN 3.0

Complete-Compare es una tecnología de Logic Works que cambia la forma de modelar la interacción de modelos o bases de datos. Provee de una comparación comprensiva de todas las diferencias. Resaltando cualquier discrepancia, los cambios pueden migrarse de un modelo-a- base de datos o de una base de datos-a-un modelo. Erwin también integra la base de datos en el desarrollo del proceso de la aplicación. Cuando se ligan las herramientas de desarrollo, Erwin sincroniza el *back-end* de la base de datos con las formas del *front-end*.

ERSTUDIO 2.5

ER/Studio ayuda a prolongar la inversión que se ha hecho. Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.

SYSTEM ARCHITECT 4.0

SA cuenta con la herramienta Screen Painter, con la que se pueden diseñar pantallas y menús para aplicaciones de Windows y pantallas de caracteres para aplicaciones de COBOL. Los archivos de Windows .DLG y .MNU se pueden generar automáticamente por SA, incluyendo posición, leyenda, *hot key*, orden de tablas y número de identificación para cada control o elemento de menú incluido en la aplicación original de Windows. Pueden crearse pantallas usando controles estándar de Windows, incluyendo botones de presión, cajas de revisión, botones de opciones, cajas combo, cajas de listas, cajas de texto. Se pueden generar archivos de diálogos

(.DLG), encabezado(.H), y forma de Visual Basic (.FRM) para aplicaciones de Windows. Usa una rutina simple de captura para crear un nuevo menú de cualquier menú de aplicación de Windows. Dibujar menús usando los elementos del menú, submenú y separadores. Agregar accesos directos para elementos del menú y submenú. Los elementos del submenú y del menú se activan usando el cursor. Los diálogos seleccionados se abren de elementos seleccionados del menú.

POWERDESIGNER 6.1

MetaWorks es un sistema diseñado para proveer los módulos gráficos de PowerDesign con la habilidad de compartir y almacenar modelos de datos en un sólo punto de control, el Diccionario MetaWorks. MetaWorks se ejecuta en una PC y almacena los modelos de datos en un servidor de bases de datos, que puede ser Sybase, SQL Anywhere o cualquier otro como Oracle, Informix, DB2, MS SQL Server y CA OpenIngres. MetaWorks provee de tres funciones principales: *Data Model y Submodel Extraction/Consolidation*, *Project (or Dictionary) Management* y *Environment Administration*. El MetaBrowser presenta vista de árbol en una línea jerárquica de la aplicación bajo el estudio (base de datos, proyecto, modelo, objeto y submodelo), expande o colapsa vista de objetos, crea, modifica, borra o imprime objetos seleccionados, habilita comparación entre modelos del mismo tipo, en el nivel de objeto, trabaja con listas de objetos a través de cualquier proyecto, modelo o submodelo.

h) Web

ERWIN 3.0

Logic Works Net Results Suites integra el modelado de datos Erwin con las mejores herramientas de desarrollo actuales. Lo que provee de una solución integral para diseñar aplicaciones de Web y el servidor de base de datos *back-end*. ERwin/NetDynamics Suite combina el ambiente de

desarrollo de aplicación NetDynamics Java-based, ayuda a que los desarrolladores construyan rápidamente aplicaciones de Web de alto desempeño.

ERSTUDIO 2.5

Publicación automática en el *web*. ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.

SYSTEM ARCHITECT 4.0

No tiene publicación en Web.

POWERDESIGNER 6.1

Genera reporte en archivo tipo HTML.

i) Metodología

ERWIN 3.0

Soporta metodología para diagramas de Relación-Entidad, modelos de datos IDEF1 y Yourdon/DeMarco.

ERSTUDIO 2.5

Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

SYSTEM ARCHITECT 4.0

El editor de reporte SA también incluye más de 125 reportes pre-definidos, incluyendo reportes de metodología para Diagramas de Relación de Entidad, modelos de datos IDEF1, Diagramas de Flujo de Datos para Word & Mellor, Gane & Sarson y Yourdon/DeMarco.

POWERDESIGNER 6.1

Soporta Modelos Funcionales y Notaciones de Diagramas de Flujo Modelo Funcional de Objeto (OMT) Yourdon/DeMarco Gane & Sarson SSADM (Análisis de sistema estructurado y metodología de diseño, Structured System Analysis & Design Methodology)

j) Reportes

ERWIN 3.0

Los modelos de datos de Erwin pueden editarse, verse e imprimirse de diferentes maneras. Se incluye un *RPTwin*, que es un generador de reportes gráfico y fácil de usar y un *Report Browser built-in* con opciones predefinidas de reportes, que proveen un control completo sobre la apariencia y contenido de los reportes. Además, una sola plantilla de interfaz permite aplicar un estándar de diseño y desplegar preferencias para todos los modelos.

ERSTUDIO 2.5

Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que esta disponible para presentaciones profesionales.

SYSTEM ARCHITECT 4.0

El editor de reportes SA es un elemento estándar en System Architect, esta herramienta permite especificar reportes personalizados para el proyecto con un front-end fácil de usar. El editor de reporte SA también incluye más de 125 reportes pre-definidos, incluyendo reportes de metodología para Diagramas de Relación de Entidad, modelos de datos IDEF1x, Diagramas de Flujo de Datos para Ward & Mellor, Gane & Sarson y

Yourdon/DeMarco. Reportes definidos por el usuario. Los reportes pre-escritos, también pueden ser modificados con el editor de reportes GUI. Si un nuevo reporte es necesario el reporte de GUI provee de una forma fácil y rápida de crear reportes personalizables. Todos los reportes pueden incluir una imagen del diagrama relevante. El editor de reportes GUI también provee de flexibilidad agregada para controlar las letras, bordes, orientación, encabezados y más. Cuando el reporte está listo crea un archivo ASCII y/o RTF en lugar del reporte impreso.

Se usa el Editor de Reportes SA para:

- Reportar propiedades definidas por el usuario.
- Crear reportes de matrices para analizar los datos del repositorio.
- Crear algo específico, leer, actualizar, borrar (CDRU).
- Desempeñar Análisis de Afinidad.

POWERDESIGNER 6.1

Creación flexible de reportes estructurados a través de plantilla de reportes. Estructura de árbol de elementos seleccionados para facilitar la organización. Objetos drag-and-drop con estructura de árbol para facilitar los ajustes. Salva plantillas de reportes. Vista previa del reporte antes de imprimirlo. Seleccionar un lenguaje por omisión para el reporte. Dirigir la impresión o exportar a Microsoft Word, Word Perfect, PageMaker, etc.

HERRAMIENTAS CASE PARA SISTEMAS ORIENTADOS A OBJETOS

Paradigm plus

Tiene la herramienta OOCL (Object Oriented Change and Learning), que les permite a los gerentes capturar, modelar y comunicar estrategias para desarrollar equipos por medio del mapeo de requerimientos de negocios directamente a use-case, objetos y bases de datos físicas. Funciona en base a diagramas, el diseño orientado a objetos interactúa muy bien con las bases de datos. Los componentes pueden compartirse y reusarse en diferentes equipos, proyectos y hasta en la red.

Software through pictures

Automáticamente prueba las especificaciones del proyecto, ahorra tiempo de desarrollo y mejora la productividad habilitando administradores para planear y evaluar el proceso de prueba mientras se elimina todo proceso manual posible.

Incluye varias opciones de metodología que se pueden cambiar sin dificultad. Esta herramienta permite usar un ambiente multi-usuario integrado que comparte una arquitectura común y un repositorio central.

Rational Rose

El área de primordial es la ingeniería de reverso. El proceso de ingeniería de reverso con esta herramienta sirve para ejecutar "analyze", configurarlo para determinar que información se debe usar para la ingeniería de reverso, iniciar el proceso y exportarlo a un archivo. Este archivo tiende a ser leído por la herramienta de modelado. Si se hace cualquier cambio al código que necesite reflejarse en el modelo, entonces el proceso de ingeniería de reverso total se hace una vez más. Una versión futura del producto tendrá la habilidad de desempeñar personalización usando el lenguaje básico.

Esto puede proveer de una manera para integrar totalmente el Versant y la habilidad de la ingeniería de reverso.

Select Enterprise

Algunas de sus ventajas son, que tiene un soporte sólido para UML, genera IDL CORBA y además el Select Perspective enseña el proceso, no sólo la herramienta; lo cual es poco común además de ser muy útil.

Algunas de sus desventajas son que los elementos de la herramienta no están integrados en un único IDE. El IDE no está actualizado para soportar la última versión de la guía que se usa con Microsoft Word 97. Aunque posee una herramienta muy útil para elaborar reportes, esta utilidad está integrada con Microsoft Word.

Select Enterprise está empaquetado en base al ambiente de desarrollo objeto para lo cual se provee el código. Se incluye también el Object Animator, una herramienta interesante que te permite explorar la secuencia de diagramas de manera amigable.

System Architect 4.0

Este producto tiene soporte para UML y las metodologías de punta en cuanto a orientación a objetos se refiere.

Incluye el Screen Painter que soporta prototipo de interfase de usuarios como parte del proceso de diseño, lo cual no es muy común en este tipo de herramientas. Este elemento también está integrado con el generador de código por lo que las pantallas se acarrean a los proyectos creados.

ANEXO 2

MANUAL DE USUARIO

IMPORTANTE:

Tal como se planteó anteriormente, se debe ingresar todos los datos antes de proceder a la graficación de los Diagramas, de esta forma, mantenemos un mejor control de errores y se posibilita que el diseño deba ser lo más correcto posible.

El programa TOOLCASE es una aplicación CASE que permite realizar: Diagrama de Flujo de Datos, Diagrama Entidad Relación, Diccionario de Datos, Comprobación de Errores, Seguimiento de Niveles.

La primera pantalla es de referencia y para poder continuar debe hacer click en cualquier lugar de la pantalla.



En esta pantalla, solo se debe realizar click en cualquier lugar para pasar a la siguiente.

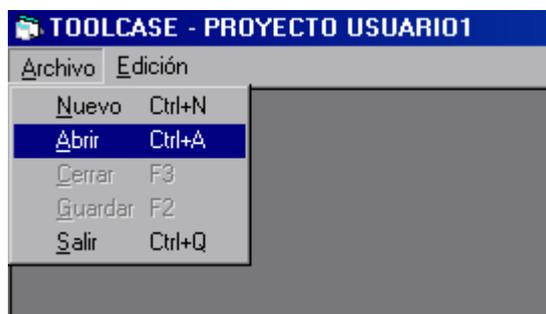
A continuación la pantalla nos muestra un requerimiento de login y password, con lo cual podemos determinar que usuario es el que realiza el trabajo respectivo. Debe ingresar un nombre de usuario que este registrado en las bases de datos en este caso existe el usuario USUARIO1 y su contraseña es USUARIO1.



Para poder registrar más usuarios se debe ingresar como ADMIN y su contraseña es ADMIN. Luego de hacer click en Aceptar o Enter se ingresa al menú principal.

En este menú se tiene las opciones de Archivo, Edición.

En el Menú Archivo, se puede abrir un proyecto ya existente, hacer un nuevo, guardar los cambios, cerrar el proyecto sin grabar y salir del programa.



En la Opción Nuevo tendrá que ingresar el nombre de un nuevo proyecto, un identificador que usted desee asignarlo y, además, el tipo de diagrama que va a realizar.



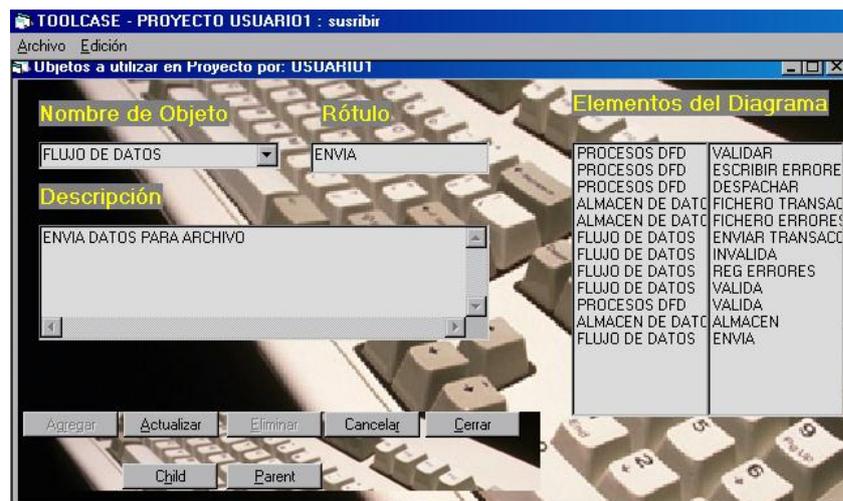
En la opción Abrir, se tiene la posibilidad de escoger y abrir un proyecto ya existente.

Para abrir el proyecto ya existente se debe dar doble click en la selección realizada.



La opción Cerrar y Guardar se activan una vez que se haya abierto o generado un nuevo proyecto. Estas presentan solo ventanas de información en el caso de que el proyecto ya existiera.

En el menú Edición, se tienen las opciones para ingresar los datos de los respectivos diagramas. Antes de realizar la grafica en el caso de que el proyecto sea nuevo, se debe ingresar la información del DFD o del DER.



En nombre de objeto se debe escoger de los objetos disponibles en la Lista. El rótulo es un valor que se desplegará el momento del grafico y la Descripción sirve para tener más información del objeto ingresado.

Se debe ingresar aquí todos los objetos que se vaya a utilizar en el Diagrama.

Siempre que se quiera Ingresar un nuevo objeto, se debe primero hacer click sobre el Botón Agregar y luego ingresar los valores. Una vez ingresado, se debe hacer click sobre el botón Actualizar, para que los valores se registren.

Cuando se hayan terminado de ingresar los objetos, se debe hacer click en el Botón Cerrar.

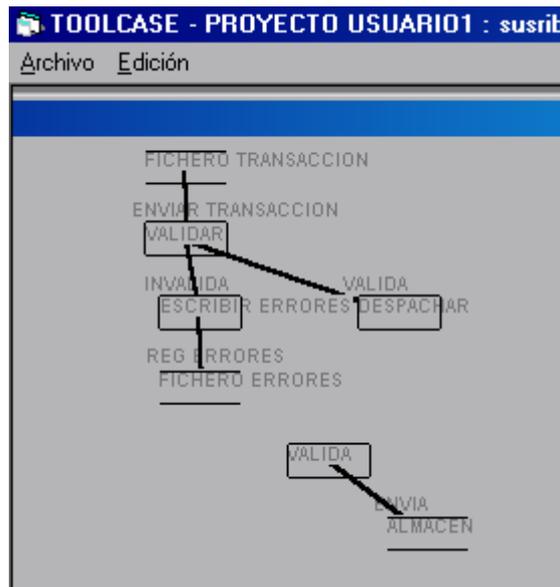


Una vez ingresado los objetos, se debe ingresar las conexiones, Para lo cual se debe escoger el menú Edición, opción Diagrama de Flujo de Datos, opción Flujo de Datos.

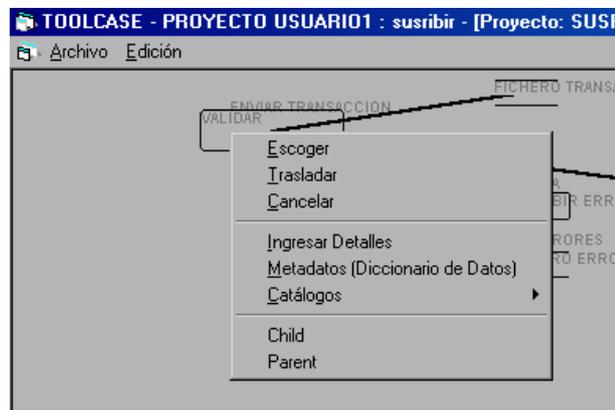
Al igual que el anterior se debe hacer click sobre Agregar y luego digitar los valores. Para registrar los cambios se debe hacer click en Actualizar.

Si se ha terminado de digitar todas las conexiones hacer click en Cerrar.

Una vez ingresado los objetos y las conexiones si se desea observar el diagrama se debe hacer click sobre Graficar, ante lo cual se le mostrara toda el diagrama de flujo de datos. Este Diagrama inicial esta diseñado en una estructura de árbol.



En este diagrama, se pueden mover los objetos al lugar que se desee, para lo cual se debe apuntar al objeto que se va a mover, luego sobre é hacer click DERECHO con el mouse y le saldrá un menú de Escoger, Trasladar, Cancelar, Ingresar detalles, Ver child o parent, etc.

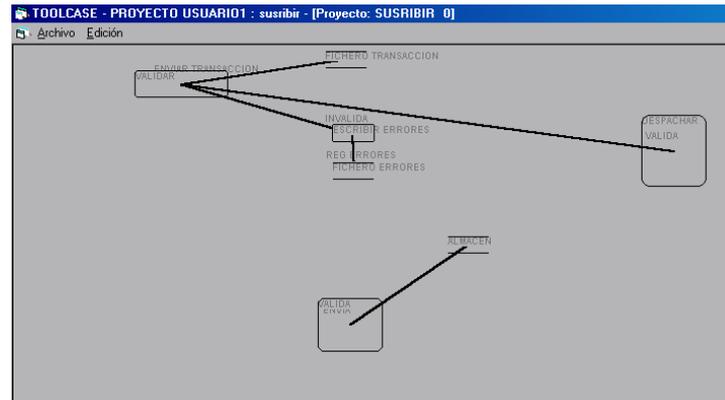


Al hacer click en Escoger, el objeto seleccionado cambiará de aspecto.

Luego en cualquier parte del formulario hacer click derecho del mouse, y podrá escoger entre Trasladar o Cancelar.

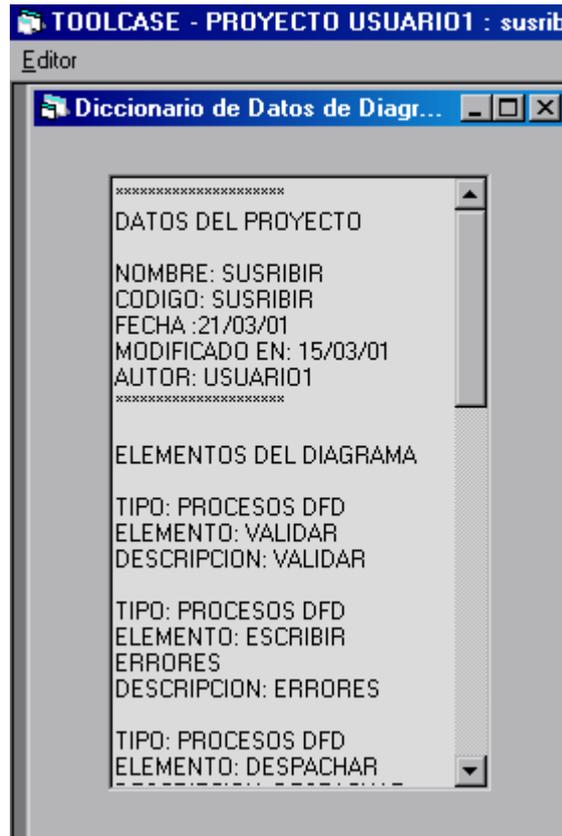
Si elige Cancelar, se objeto que escogió y que se desapareció, volverá a su posición original.

Si elige Trasladar, deberá con el puntero del mouse señalar las ubicaciones iniciales y finales para el objeto elegido.



En la Opción de Edición, existe la subopción de Diccionario de Datos, que le permitirá ver los Datos que usted ha ingresado.





En el menú emergente, del diseño gráfico, usted puede escoger child o parent para ver el respectivo nivel superior o inferior respectivamente. Si un elemento no contiene child o parent, aparecerá un mensaje.



Para completar el diccionario de datos de cada elemento, el menú emergente posee la opción de ingresar detalles y metadatos. En estas opciones se presentará una ventana de requerimientos para poder ingresar la información que dé a lugar.

The image shows a screenshot of a software application window titled "TOOLCASE - PROYECTO USUARIO1 : susibir". The window has a menu bar with "Archivo" and "Edición". Below the menu bar, there is a status bar that says "ecto por: USUARIO1". The main area of the window contains a form with the following fields:

- NOMBRE**: FICHERO TRANSACCION
- ALIAS**: (empty text box)
- DESCRIPCION**: (empty text area with scrollbars)
- RELACIONES CON OBJETOS**: (empty text area with scrollbars)
- TIPO DE DATO**: (empty dropdown menu)
- LONGITUD**: (empty text box)
- INFORMACION ADICIONAL**: (empty text area with scrollbars)

El procedimiento para elaboración del Diagrama de Entidad Relación sigue los mismos pasos aquí descritos.

ANEXO 3 MANUAL TÉCNICO

En este anexo se presentarán los diagramas utilizados para el desarrollo de la aplicación case; así como también referencias de las estructuras más importantes usadas el código fuente.

El software está desarrollado en Visual Basic 6.0.

La base fundamental de este software, está en las funciones contenidas en los módulos y en el formulario FRMCASE. Adicionalmente se tiene como apoyo a los formularios FRMGRAFAYU y COORDENADAS. Antes de explicar esos elementos; debemos conocer las estructuras principales que se están usando.

MATRICES Y VARIABLES

Matriz grafo: al ingresar las conexiones en esta matriz dinámica se determina cuales son los puntos de conexión

Matriz fija grafodfd: esta matriz contiene cuales son las conexiones que los diagramas pueden soportar. Por ejemplo: Si se desea conectar directamente dos almacenes de datos, en esta matriz se chequeará si esa conexión es válida.

Matriz dinámica grafico: contiene un indicativo del tipo de gráfico que se va a dibujar así: 1 ó 5 proceso, 3 conector, 4 almacén de datos, 6 atributo, 15 texto. Esta matriz sabe que tipo de gráfico se ha realizado, conjuntamente con la matriz coord que lleva las coordenadas dentro del formulario. Para hacer referencia a estas matrices se usa la variable posgra, que es la que contiene la posición del gráfico dentro de las matrices, que se desea escoger o generar.

Matriz dinámica coord: lleva un registro de las coordenadas de cada objeto dentro el formulario.

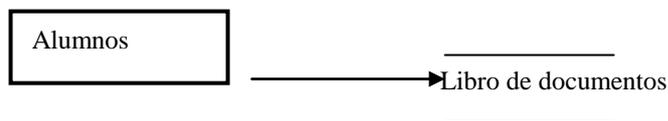
Matriz dinámica phc: Matriz dinámica que contiene el número del padre, número del hijo y que tipo de conector va a usar.

Matriz dinámica clisto: determina si la línea de unión entre dos elementos del diagrama ya ha sido graficada (booleana)

Matriz dinámica poscoor: indica el orden en que los objetos deberán ser desplegados en la graficación

Matriz dinámica dibujado: determina si un elemento ya ha sido dibujado.

Matriz dinámica numerodiag: contiene el número de entradas y/o salidas que tiene un elemento del diagrama. Por ejemplo:



En el diagrama anterior, la entidad alumnos tendrá registrado en la matriz numerodiag una salida y el almacén libro de documentos tendrá registrado una entrada.

Matriz dinámica conecciones: Matriz dinámica para determinar cual es el flujo entre los elementos. Es decir, esta matriz sabe cuál es el elemento origen y cual es el elemento destino.

Matriz dinámica conectores: Esta matriz dinámica actúa junta con la matriz conexiones y en ella se registra cuál es el conector que une a los dos elementos.

Matriz dinámica par_ch: Matriz que contiene unificadamente, el identificador del padre y del hijo; es decir, del origen de una conexión y el destino de la misma. En la posición (0,x) está el padre y en la posición (1,x) está el hijo; donde “x” es la variable posgra.

Variable posgra: Variable que contiene las posiciones de los elementos del diagrama. Mediante esta variable puedo referenciar a un valor de un objeto dentro de las matrices citadas anteriormente. La variable posicoord funciona cuando el diagrama ya existe, es decir, el diagrama ya ha sido generado previamente.

Variable cont: Mediante esta variable se realiza la carga inicial de los elementos del diagrama. Cada objeto debe estar en una posición diferente para asegurar que no se contrapongan los elementos.

Variable ruta: Esta variable contiene la ruta (path) desde donde se ejecuta el programa.

Variable usuario: Contiene el identificador del usuario que está utilizando el programa.

Variable nomproy: Contiene el nombre que el usuario le asignó al proyecto de diagrama.

Variable idproy: Contiene el identificador de abreviación que se le asignó al proyecto.

Variable tipdiag: Contiene el tipo de diagrama diseñado; si es Flujo de Datos es 1 y si es Entidad Relación es 2.

MODULOS

VARIABLES.BAS

Este módulo contiene la declaración de las variables y matrices que indicamos en la sección anterior.

FUNCIONES.BAS

Contiene funciones para inicializar las matrices y variables; además, de los formularios de apoyo FRMGRAFAYU (que contiene toda la información de los elementos del diagrama, sus identificadores, sus conexiones; y COORDENADAS (que contiene la información de las coordenadas de cada elemento). Estas funciones son inicio (para matrices y variables), e iniciolist (para los formularios de apoyo).

Funciones para graficar automáticamente los elementos de los diagramas propuestos. Estas funciones son: ALMDFD (para los gráficos de almacenes de datos), ENTDFD (para los gráficos de entidades tanto para los diagramas de flujo de datos, cómo para los de entidad relación), PROCDFD (para los gráficos de proceso de datos), GRAFICARLINEA (para los gráficos de conexiones entre gráficos), ATRIBDER (para los gráficos de atributos de datos), RELER (para los gráficos de relaciones de datos).

Funciones que mantienen las coordenadas de los diferentes tipos de elementos de un diagrama: COORDENADA (mantiene el registro de las coordenadas de los objetos del diagrama), COORDLINEA (mantiene el registro de las coordenadas de las líneas que conectan a los objetos del

diagrama), COORDTEXT (mantiene el registro de las coordenadas de los rótulos de cada elemento del diagrama). Es necesario aclarar que estas coordenadas se almacenan en la misma matriz coord.. La variable cont lleva el registro de que tipo de gráfico es. Además, la matriz poscoor, mantiene también el registro de cuál objeto acaba de ingresar o ser graficado.

Función para llevar el registro de los objetos que se encuentran vinculados; es decir, los objetos que tienen conexión de origen y de destino.

REGRAFICO.BAS

Las funciones aquí incorporadas, se las utiliza cuando el diagrama ya ha sido diseñado y se desea sólo hacer modificaciones al mismo.

Las funciones son PROCDFD1, ALMDFD1, ATRIBDER1, ENTDFD1, COORDENADA1, COORDLINEA1, COORDTEXT1. Su funcionamiento es similar a las funciones anteriores.

RUTA.BAS

En este módulo se tiene la base inicial de las funciones principales para llamar a las funciones anteriormente citadas, en los casos en que el gráfico aún no exista ó en el caso de que se desee editar un gráfico ya existente.

La función GRAFICARDFD, contiene las llamadas a las funciones: ENCGRAFICARDFD (función donde se asigna los valores iniciales para realizar un diagrama, en el caso de que el diagrama no exista, es decir de que sea nuevo los valores se pondrán a nulos; en el caso de que el diagrama si exista, las matrices, variables y demás estructuras, tomarán los respectivos valores almacenados), GRAFICARNUEVO (Si el diagrama no existiese o si se le da la orden de rehacer el gráfico, los valores que

podieran haber sido recuperados se eliminarán y a cambio se reestructurará el diagrama. Esta función tiene implementado el mecanismo de dibujo en estructura de tipo árbol), LINEASNUEVO (idéntico al anterior, cuando se grafica nuevamente o por primera vez, las líneas de conexión se regraficarán con valores en estructura de árbol.

Las funciones GRAFICAR y GRAFICAR1 funcionan; la primera cuando el diagrama es nuevo o regraficado y el segundo cuando el diagrama ya existe.

EDITOR.BAS

Este módulo contiene funciones que son aplicables para editar un texto, el momento que vemos los diccionarios de datos, se activa el formulario EDITOR.FRM y con éste las funciones de este módulo. Son estructuras de un editor básico desarrollado en Visual Basic, que son ya comunes a todo desarrollador en este lenguaje.

FORMULARIOS

FRMCASE.FRM

Este es el formulario más importante, porque en este se realizan el resto de procesos de la diagramación automática. La programación consta en los procedimientos MOUSEDOWN, MOUSEMOVE y MOUSEUP. El procedimiento LOAD detecta si el diagrama es nuevo o ya existe; para poder proceder a abrir sus valores almacenados o inicializar todas las matrices y variables.

En el procedimiento mousedown detecta sobre cual objeto se ha realizado un evento click derecho, luego de lo cual comienza a buscar en toda la matriz coord, que es la que almacena las coordenadas de los objetos

diagramados. El momento que encuentra las coordenadas procede a la función ESCOGER, en la cual mediante el vector gráfico sabe cual es el objeto que se ha escogido.

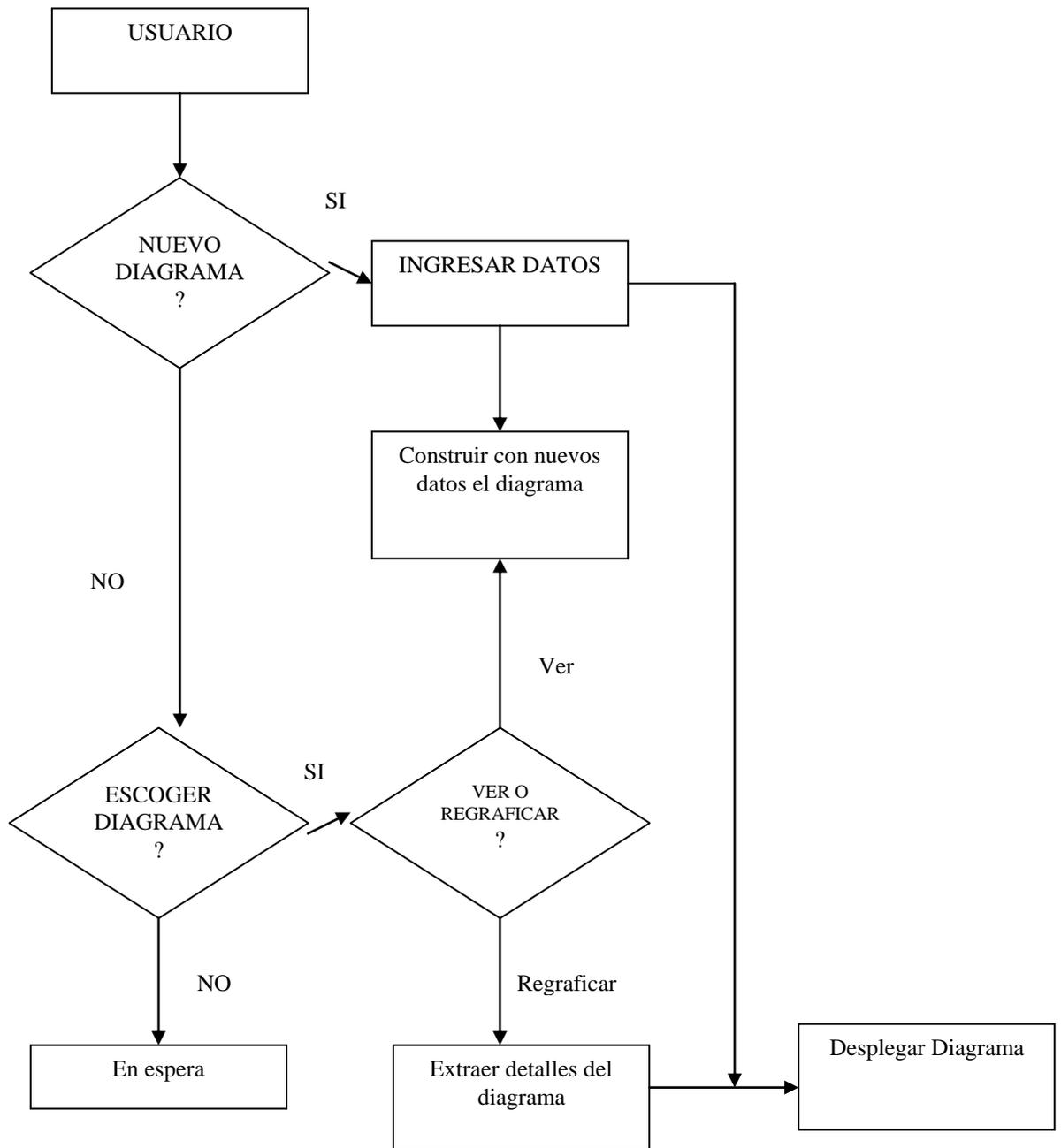
En el procedimiento mouseup, en cambio detecta las coordenadas iniciales y finales del objeto que ha sido diagramado, mediante el botón izquierdo del mouse.

Además también detecta si se ha escogido otra opción como ver child, parent, ó algún diccionario o catálogo, para lo cual chequea la variable OPDATO que es la que tiene la información de que operación va a realizar.

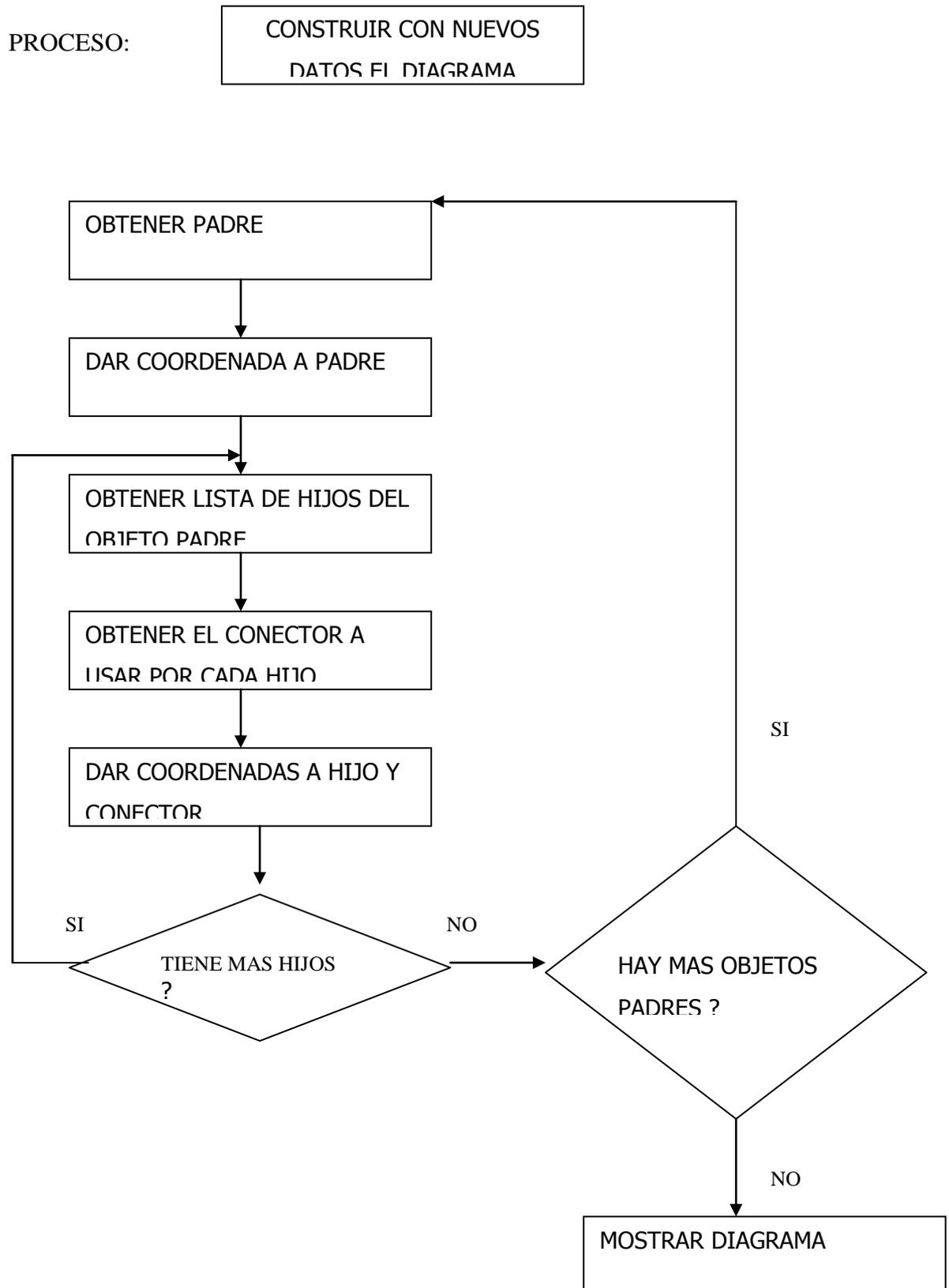
El resto de formularios son de apoyo, y contienen estructuras conocidas de programación en Visual Basic.

Se presenta también diagramas de bloque del funcionamiento del programa:

PROCESO PRINCIPAL

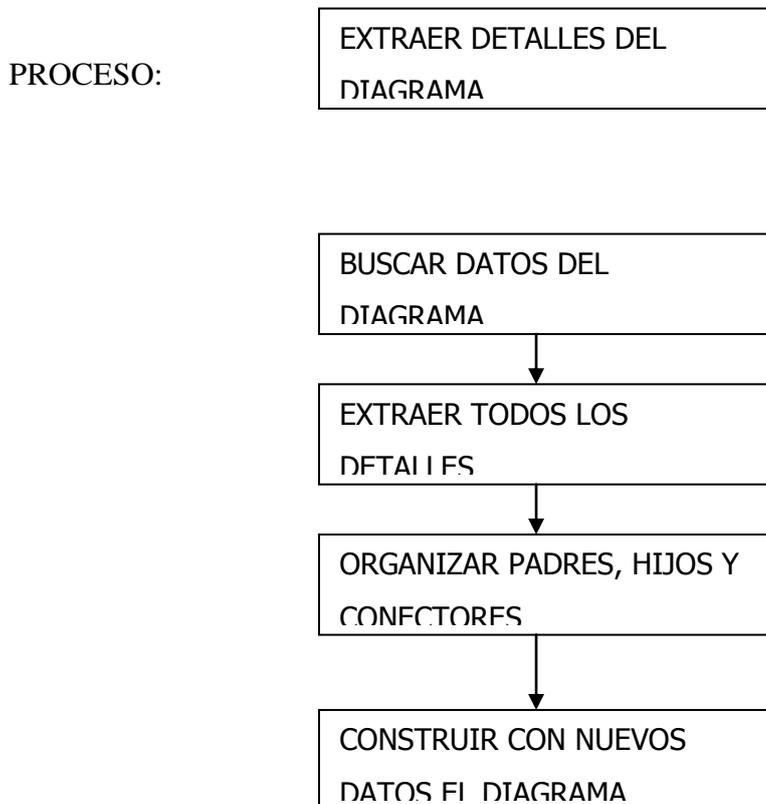


En este diagrama, se presenta el bloque de visualización y/o construcción de un diagrama.



En este diagrama, nos muestra los pasos a seguir para la construcción de un nuevo diagrama. Debemos obtener de la lista ingresada o recuperada, la representación

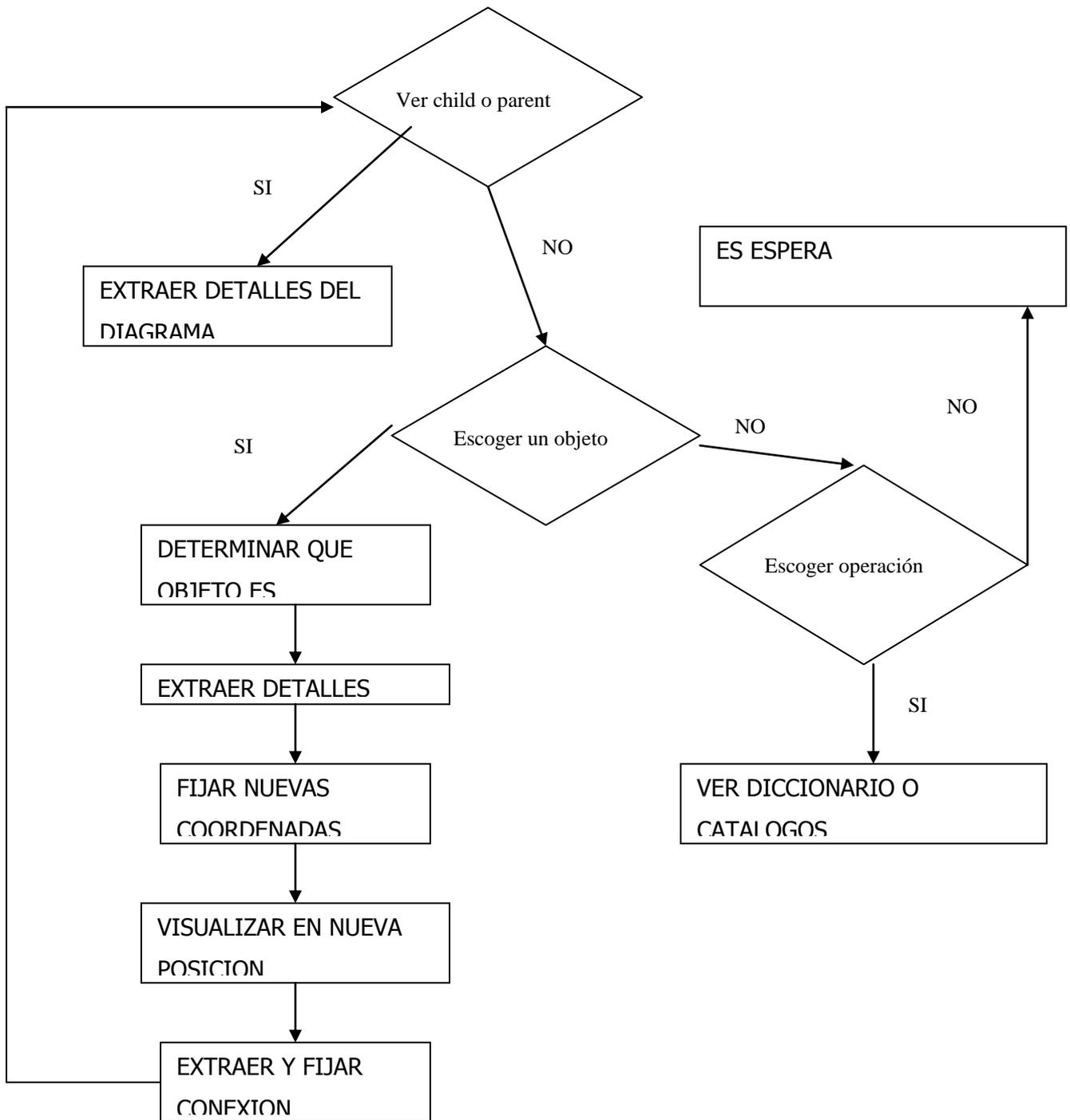
de los padres, hijos y conectores (matriz phc, coord., grafico y otros). Al obtener esta lista la diagramación se la realiza en forma de árbol.



La extracción de detalles del diagrama, empieza con la búsqueda de los datos del diagrama, una vez encontrado estos datos, se los pasa todo a las matrices, vectores y variables, para de trabajar con estas. En esta mismo proceso se debe realizar la reestructuración de padres, hijos y conectores, para saber luego que objetos están en el diagrama.

PROCESO:

MOSTRAR DIAGRAMA



El proceso de mostrar Child o Parent, sigue las mismas reglas anteriores. Es decir, que el diagrama child o parent debe ser tratado como un elemento del proyecto global. De esta forma buscamos mediante el código asignado y procesamos

Extraer Detalles del Diagrama. En caso de que no escoja uno de estos, puede tener la opción de elegir un objeto puntual del diagrama a fin de colocarlo en una nueva posición. Para lo cual se extraen los detalles de este objeto y se pasan a las nuevas coordenadas. En el caso de que elija otra operación, esta pasará a visualizar datos de los Diccionario o catálogos.

PROCESO:

EN ESPERA

MENU DE OPCIONES EN
ESPERA A DIGITAR UNA
ACCION

El proceso en espera significa, que el usuario no ha escogido ninguna opción de procesamiento, el sistema esta esperando una orden, que en último de los casos debería ser salir del sistema.

BIBLIOGRAFIA

- ACHIG, Lucas Guía de Planeamiento Universitario, Universidad Estatal de Bolívar, Guaranda Ecuador, 1997.
- ACKOFF, Russell Planificación de la Empresa del Futuro, Noriega Editores, México D.F. México, 1994.
- ANDREU Rafael,
RICART Joan,
VALOR Josep Estrategia y Sistemas de Información. Segunda Edición, Serie McGraw-Hill de Management. Barcelona España, 1996.
- BURCH, Jhon
GRUDMITSKI Gary Diseño de Sistemas de Información. Teoría y Práctica. Megabyte. México. 1994
- CONTRALORIA
GENERAL DEL ESTADO Biblioteca Jurídica Ecuatoriana. Vademecum Legal, Volumen III
- DMITRIEV, V.I. Teoría de Información Aplicada, Ed. Mir Moscú, Moscú Rusia, 1991
- DORSEY, Paul y
KOLETZKE, Peter Manual de Oracle Designer/2000, Ed. Osborne / McGraw Hill, Madrid España, 1997.
- GRANT, William Planeamiento de Fábricas. Ed. Hispano Europea, Madrid España, 1990.
- HOYLE, David ISO 9000 Manual de Sistema de Calidad. Ed. Paraninfo, Madrid, 1995.
- KENDALL, Kenneth y
KENDALL, Julie Análisis y Diseño de Sistemas, Ed. Prentice Hall Hispanoamericana, México, 1997
- KORTH, Henry y
SILBERSCHATZ,
Abraham
LI, David Fundamentos de Bases de Datos, Ed. McGraw-Hill, España, 1997
- Auditoría en Centros de Cómputo. Ed. Trillas, México, 1993.
- McCLURE, Carma CASE, La Automatización del Software, Ed. Addison Wesley Iberoamericana, Delaware Estados Unidos de Norteamérica, 1993.
- PEPPARD, Joe y
ROWLAND, Phillip La Esencia de la Reingeniería en los Procesos de Negocios, Ed. Prentice Hall Hispanoamericana, México, 1996.
- PRESSMAN, Roger Ingeniería del Software. Un Enfoque Práctico, Ed. McGraw Hill, Madrid España, 1998

- SENN, James Análisis y Diseño de Sistemas de Información. Ed. McGraw Hill, México, 1992
- STEINER, George Planeación Estratégica: Lo que todo Director Debe Saber, Ed. CECSA, México, 1991
- TERRY, George Administración y Control de Oficinas, Ed. CECSA, México, 1993.

REFERENCIAS A PAGINAS WEB

<http://www.logicworks.com>

<http://www.cayennesoft.com>

<http://www.platinum.com>

<http://www.aonix.com>

<http://www.rational.com>

<http://www.selectst.com>

<http://www.popkin.com>

<http://www.logicworks.com/products/bpwin/bpwin.html>

<http://www.aisintl.com/case/products/product.html>

<http://www.sybase.com/products/powerdesigner/>

<http://www.sybase.com>

<http://www.information-engineer.com>

<http://www.inei.gob.pe/cpi/bancopub/lifree/lib606/n00.html>

<http://agamenon.uniandes.edu.co/~magister/areas/informa.html>