

36
2 y



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

C A S E

INGENIERIA DE SISTEMAS ASISTIDA POR
COMPUTADORA COMO BASE PARA EL
DESARROLLO DE SISTEMAS

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

P R E S E N T A N :

ENRIQUE MONTFORT STEIGER
ERNESTO VAZQUEZ BRAVO

DIRECTOR ING ALEJANDRO JIMENEZ

México, D. F.

1991

FALLA GEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS CON FALLA DE ORIGEN

INDICE

INTRODUCCIÓN	1
<u>CAPÍTULO 1</u> INGENIERÍA DE SISTEMAS ASISTIDA POR COMPUTADORA (CASE)	7
1.1 Antecedentes	9
1.2 Tecnología y Evolución de CASE	10
1.2.1 CASE: Primera Generación	10
1.2.1.1 Upper-CASE	11
1.2.1.2 Middle-CASE	12
1.2.1.3 Lower-CASE	12
1.2.2 CASE: Segunda Generación	13
1.2.2.1 Pre-CASE	14
1.2.2.2 Incorporación de la Tecnología de Bases de Datos	14
1.2.2.3 Post-CASE	15
1.3 Tendencias de CASE	17
<u>CAPÍTULO 2</u> ESTUDIO DE CASE*METHOD	19
2.1 Descripción de CASE*Method	20
2.1.1 Desarrollo de proyectos	23
2.2 Etapa de Estrategia	25
2.2.1 Objetivos y descripción	25
2.2.2 Actividades principales	26

2.2.3 Factores críticos para tener éxito	29
2.2.4 Productos resultantes	29
2.3 Etapa de Análisis	30
2.3.1 Objetivos y descripción	30
2.3.2 Actividades principales	30
2.3.3 Factores críticos para tener éxito	33
2.3.4 Productos resultantes	33
2.4 Etapa de Diseño	34
2.4.1 Objetivos y descripción	34
2.4.2 Actividades principales	34
2.4.3 Factores críticos para tener éxito	39
2.4.4 Productos resultantes	39
2.5 Etapa de Construcción	40
2.5.1 Objetivos y descripción	40
2.5.2 Actividades principales	40
2.5.3 Factores críticos para tener éxito	42
2.5.4 Productos resultantes	42
2.6 Etapa de Documentación	43
2.6.1 Objetivos y descripción	43
2.6.2 Actividades principales	43
2.6.3 Factores críticos para tener éxito	44
2.6.4 Productos resultantes	45
2.7 Etapa de Transición	45
2.7.1 Objetivos y descripción	45
2.7.2 Actividades principales	46
2.7.3 Factores críticos para tener éxito	49

2.7.4 Productos resultantes	49
2.8 Etapa de Producción	50
2.8.1 Objetivos y descripción	50
2.8.2 Actividades principales	50
2.8.3 Factores críticos para tener éxito	53
2.8.4 Productos resultantes	54
2.9 CASE*Method dentro de la Tecnología CASE	54
<u>CAPÍTULO 3</u> HERRAMIENTAS CASE	57
3.1 CASE*Method	60
3.2 CASE*Designer	60
3.3 CASE*Dictionary	60
3.4 CASE*Generator	61
3.5 Las Herramientas dentro de la Tecnología CASE	61
<u>CAPÍTULO 4</u> CASO PRÁCTICO	63
Elaboración de un Sistema de Computo para el Área Educacional de una Empresa	
Introducción	65
4.1 La Etapa de Estrategia del Sistema	66

4.2 El Análisis del Sistema	71
4.3 El Diseño del Sistema	79
4.4 La Construcción del Sistema	92
4.5 La Documentación del Sistema	95
4.6 La Transición del Sistema	98
4.7 La Etapa de Producción del Sistema	99
CONCLUSIONES	101
APÉNDICE A (Análisis Funcional-Jerárquico).....	105
APÉNDICE B (Diagrama de Entidad-Relación).....	175
BIBLIOGRAFIA	195



INTRODUCCIÓN

INTRODUCCIÓN

En el vasto mundo de la computación existen ciertas tendencias innegables. Por un lado se tiene el avance acelerado en la tecnología del hardware, por otro lado se tiene un problema muy grande con el desarrollo de las aplicaciones. Si bien es cierto que es muy beneficioso contar con computadoras cada vez más poderosas y rápidas, también lo es que el software que utilizan estas máquinas se está volviendo cada vez más difícil de desarrollar, y que su complejidad y tamaño irremediablemente tienen que crecer.

Estas tendencias nos ponen en una situación muy difícil, ya que el desarrollo de las aplicaciones es hecho por el hombre y se ha comprobado que en los últimos años la brecha que existe entre el desarrollo de hardware y el de software es cada vez más amplia.

Los métodos y ambientes para el desarrollo de programas no han evolucionado a la par con respecto a los grandes descubrimientos y logros de la electrónica utilizada en la fabricación de los circuitos integrados.

La demanda de sistemas cada vez más complejos y grandes aumenta día a día. La brecha que existe entre la producción del software y su demanda se vuelve cada día más grande. Esta problemática ya fue mencionada en los años 60's por el alemán Edgar Dijkstra, y hoy día presenta uno de los problemas más serios de la computación. Tal es la gravedad del caso que se le ha bautizado con el nombre de *la crisis del software*. Estadísticas realizadas en Japon revelan que para el año 2000 la diferencia entre la demanda y la existencia de desarrolladores de software llegará a la cantidad de 970 000, siendo que en el año de 1985 solo llegaba a 10 000, relación que representa un porcentaje mayor al 4400%¹.

Existen otros factores que contribuyen al lento desarrollo de sistemas, como son las diferencias en las aplicaciones y en el análisis del mismo, de tal manera que esto ocasiona el que multiples cambios sean necesarios una vez finalizado el conjunto de programas o que se tengan que realizar adaptaciones y enmendaduras durante la etapa del desarrollo. También es muy común que el sistema ya finalizado no provea información confiable que la información no sea oportuna, o bien, que no satisfaga las expectativas del usuario.

Estos y muchos problemas similares contribuyen a atrasarse los servicios del sistema.

¹ Module Design Methodology, Training for Information Processing, Okinawa International Centre, Japan International Cooperation Agency.

¿Cómo solucionar este problema? ¿Sería posible utilizar alguna herramienta que ayude a desarrollar sistemas de manera más eficiente? ¿Existen acaso herramientas que contribuyan a la automatización del desarrollo de software?

Afortunadamente la respuesta a estas preguntas es afirmativa. Hoy día se cuenta con una serie de herramientas de software que ayudan a los programadores a construir mejores sistemas en menos tiempo. A estas herramientas se les conoce con el nombre de CASE (Computer-Aided Systems Engineering, o bien, Ingeniería de Sistemas Asistidos por Computadora). La promesa que ofrece CASE es excitante: promete incrementar la productividad de los programadores, aumentar la calidad de nuevos sistemas y disminuir el laborioso trabajo manual en un proceso altamente automatizado. CASE pretende crear sistemas confiables, que sean flexibles al crecimiento y que estén altamente integrados, sistemas cuyo mantenimiento sea sencillo y rápido y no en base a adaptaciones improvisadas, en donde el único que sabe qué modificaciones se realizaron, es aquel que hizo la modificación. Si vemos nuevamente estadísticas realizadas en Japón, concernientes al porcentaje de tiempo invertido en mantenimiento de sistemas existentes y el referente al desarrollo de nuevos sistemas, encontraremos que el primero se lleva el 70%, mientras que el segundo sólo abarca un 30%¹.

Un punto clave para lograr esta automatización del software es la existencia de una metodología. Bajo metodología se entiende un proceso bien definido, con pasos prácticos y lógicos a seguir para poder realizar el desarrollo de un sistema desde su comienzo hasta el final.

Se trata de identificar y analizar progresivamente los detalles y pormenores de cada una de las diferentes etapas del ciclo de vida de un sistema.

CASE, a través de diversas herramientas y con la ayuda de una tecnología, trata de proveer una interfase con el usuario, un diccionario donde se pueda guardar la información obtenida, facilidades para verificar la consistencia de la misma y utilerías que permitan utilizar la información registrada en un diccionario, para poder generar automáticamente en un futuro gran parte del sistema deseado.

Estimaciones indican que sólo un 5% de los analistas y diseñadores de sistemas en los Estados Unidos hacen uso de herramientas CASE.

¿Por qué no utilizar estas facilidades de la tecnología moderna y hacer a un lado los esquemas tradicionales de desarrollo?

¹ Module Design Methodology Training for Information Processing, Okusawa International Centre, Japan International Cooperation Agency.

Este texto trata el concepto de CASE y pretende ampliar la visión de los futuros desarrolladores de software. Para lograr esto, hemos dividido el contenido en tres partes: la primera parte se centra en la explicación de los orígenes y en el concepto de CASE, la descripción de una metodología y hace mención de algunas herramientas CASE. La segunda parte abarca el desarrollo de un sistema que controla el área educacional de una empresa, en base a la metodología expuesta, haciendo uso de herramientas CASE. Finalmente, en la tercera parte se analizan ventajas y desventajas de la metodología presentada y del uso de las diversas herramientas de desarrollo.



CAPITULO 1

INGENIERÍA DE SISTEMAS ASISTIDA POR COMPUTADORA (CASE)

1.1 Antecedentes

En el mundo de hoy la utilización de la computadora se ha hecho indispensable, ha invadido todas las actividades humanas, especialmente el área de los negocios, los cuales requieren estar acorde con el constante desarrollo tecnológico para poder seguir compitiendo.

La gran cantidad de información que se maneja en una empresa, hace necesaria la utilización de la computadora para el manejo, control y análisis de dicha información.

En la actualidad se cuenta con una gran diversidad de alternativas para el uso de la computadora. Por una parte encontramos una gran cantidad de programas de software existentes en el mercado, los cuales ofrecen una gama de posibilidades para resolver ciertos requerimientos de una empresa, sin embargo estos programas (paquetes) tienen el inconveniente de estar diseñados para solucionar los problemas en una forma bastante particular.

La utilización de paquetes de software obliga al usuario a acoplarse al funcionamiento de su empresa a un producto específico, el cual rara vez cubre los requerimientos totales de ésta, por lo que el usuario se ve en la necesidad de establecer y crear sistemas que resuelvan la totalidad de su problemática, que como ya dijimos, suele ser muy particular.

Hoy en día cada empresa diseña sus propios sistemas de acuerdo a sus propias necesidades. Es aquí cuando intervienen los analistas, consultores, diseñadores, programadores y diversos profesionales dedicados a la elaboración de sistemas.

Como todos sabemos, el proceso de construcción de un sistema, está determinado por una serie de acciones a seguir que en su conjunto llamamos metodología. Ésta nos lleva a través de diversas etapas de desarrollo por las que tiene que pasar un sistema. A estas etapas se les conoce como *Ciclo de Vida del Software*. Las diferentes etapas por las que pasa un sistema en forma muy general son las siguientes: análisis, diseño, elaboración, documentación e implementación. Sin embargo existen diversas etapas (dependiendo de la tecnología) que pueden variar la forma, el nombre, el orden en que se realizan y hasta las acciones que conforman.

Pero hay que hacer notar que en esencia todas pasan de una u otra forma por las fases mencionadas. Uno de los puntos que nos interesa tratar en este momento es el de los problemas a los que se enfrentan las personas que intervienen durante todo el desarrollo de un sistema.

Dentro de la problemática con la que se enfrenta el analista y el diseñador, está el manejo de la información recolectada en las primeras etapas

Cada analista recopila información, llegando a un punto donde surge la necesidad de concentrarla, unir la para hacer el estudio de un todo, y es aquí cuando estas tareas se vuelven sumamente difíciles y tediosas, como puede ser por el excesivo papeleo que se maneja y la gran cantidad de información redundante

Por tanto, el realizar estas tareas conlleva a una utilización excesiva y costosa de recursos, tanto humanos como materiales, además de una pérdida considerable en cuanto a tiempo se refiere

Durante cada etapa se recopila y se genera gran cantidad de información, que define en su inicio la funcionalidad de una empresa y la forma en que opera ésta

Mucha de la documentación se convierte en información aislada, encontrándose principalmente en las mentes de unas cuantas personas, quedando ajena para la mayoría del personal involucrado en el desarrollo del sistema

Otro de los problemas que se encuentran es que en el momento en que comienza otra etapa de desarrollo, mucha información y cierta documentación tiene que ser recopilada otra vez, teniendo así una pérdida de trabajo considerable

Con todos estos problemas a los cuales generalmente tiene que enfrentarse el desarrollo de un sistema, surge la necesidad de sistematizar estas actividades. Para esto se crearon programas y sistemas computarizados que auxilian en las actividades y tareas a realizar durante el ciclo de vida de un sistema

Es aquí donde surge la tecnología CASE, que son las siglas inglesas de *Ingeniería de Software Asistida por Computadora* (Computer Aided Software Engineering)

1.2 Tecnología y Evolución de CASE

CASE es una tecnología que puede proveer a cualquier persona que trabaje elaborando sistemas de computación, una ayuda considerable para el desarrollo del software

1.2.1 CASE: Primera Generación

CASE ha causado mucha confusión entre los vendedores y desarrolladores de software, los

cuales han fabricado una infinidad de productos CASE, sin definir exactamente lo que CASE significa o lo que CASE hace.

Lo que se puede considerar como el nacimiento de CASE fue un artículo presentado el 17 de abril de 1984 en una conferencia en Washington D.C., titulado "Computer Aided Software Engineering", donde el autor, el DR. John H. Manley, sugiere la forma de utilizar la computadora para generar sistemas de software.

Dividiremos a CASE en diferentes partes con objeto de su estudio, lo que no quiere decir que CASE está dividido de esa forma en el ambiente real de sistemas, ya que se pueden entremezclar diferentes etapas en una misma herramienta, lo cual entonces quedaría fuera de toda clasificación.

1.2.1.1 Upper-CASE

El artículo hecho por el Dr. Manley fue originalmente un estudio de ingeniería de software asistido por computadora, donde se analizaban las tareas que desarrollaban los analistas, diseñadores y programadores de software. Propone la automatización de estas tareas por medio de sistemas de computación dirigidos a la ejecución ágil y efectiva de estos procesos. Las herramientas CASE hacen más fácil y rápido para desarrolladores el uso de convenciones y técnicas de una programación estructurada, que es considerada como parte esencial en un ambiente de desarrollo de sistemas.

Con CASE el desarrollador puede dibujar, actualizar, editar y guardar varios diagramas utilizados en las fases iniciales del ciclo de vida del desarrollo del software. La computadora también llega a ser una herramienta que puede realizar verificaciones acerca de los diagramas y de la consistencia reconocida en la primera parte del ciclo de vida, dando una mayor información como resultado de estos análisis "inteligentes".

Estos productos CASE son utilizados en las fases de recolección de requerimientos y especificaciones durante las primeras etapas, donde se plantean el análisis de estrategias a seguir y se conciben como herramientas Upper CASE.

Un gran número de metodologías han sido desarrolladas dirigiéndose hacia la construcción de modelos. Estas metodologías generalmente consisten en una combinación de técnicas gráficas (diagramas) y descripción de textos. Los diagramas expresan gráficamente las actividades de la empresa y el cómo ésta utiliza la información para soportar estas actividades. Las técnicas de diagramación utilizan figuras para representar los componentes de una empresa, como

pueden ser datos, recursos, personal, además de las relaciones entre éstos. Las descripciones de textos incluyen suficiente información para describir ampliamente el objeto del mundo real que las figuras representan. Sin embargo, estos modelos requieren frecuentes modificaciones, lo cual, si son realizados manualmente, implican un gasto considerable de recursos. Para solucionar esto, se han incorporado herramientas CASE orientadas a gráficos donde se pueden hacer, corregir y almacenar estos diagramas. Además de las herramientas que soportan el manejo de texto, se han creado herramientas llamadas diccionarios, las cuales están dentro de un *Manejador de Bases de Datos* (DBMS).

Upper CASE utiliza el software para describir lo que la compañía hace, utilizando estos diagramas para descomponerla en los aspectos que la describen. Algunos de estos aspectos pueden ser las funciones y procesos de los distintos departamentos, otros incluyen las metas, objetivos, responsabilidades, recursos, problemas, etc., proporcionando así información gráfica y textual para tener una comprensión completa de las actividades que realiza la empresa.

El crear estos modelos requiere de un enorme trabajo intelectual. Sin embargo, después de que se construyen, nos encontramos con que se pueden reutilizar muchas especificaciones introducidas en el diccionario de datos, reduciendo y ahorrando una cantidad considerable de tiempo y esfuerzo.

1.2.1.2 Middle-CASE

En el Middle CASE se analizan problemas con la información y la forma de solucionarlos, atendiéndose así el proceso de diseño. La mayoría de los sistemas a nivel de Middle CASE utilizan diagramas y diccionarios operacionalmente similares a los utilizados en el Upper CASE, sin embargo la metodología utilizada en Middle CASE es diferente, ya que estas herramientas son utilizadas a manera de analizar y estudiar la información recolectada en Upper CASE, provocando una eficiente utilización de ésta.

1.2.1.3 Lower-CASE

En contraste al Middle CASE, designamos herramienta Lower CASE a aquellas tareas compartadas en las últimas fases de vida del sistema. Las herramientas Lower CASE automatizan las tareas posteriores a la fase de diseño, incluyendo generación de códigos, de programas, pruebas, compilaciones, realización de prototipos e implementaciones.

Diversos vendedores obtuvieron en CASE un trampolín para llamar a sus generadores de código herramientas CASE. En este momento, el término llegó a lo que se puede llamar CASE **Primera Generación**.

Estas herramientas estaban disponibles para auxiliar a las tareas de las primeras fases del ciclo de vida, sin embargo los datos y la información no necesariamente fluye de una fase a otra.

1.2.2 CASE: Segunda Generación

La segunda generación de herramientas CASE aparece en 1985-1986 con la introducción de productos que soportaban tareas en varias fases del ciclo de vida de un sistema. Fabricantes de software utilizaban información obtenida en las fases de requerimientos, para desarrollar especificaciones que en su momento utilizaban en las fases de diseño.

Los fabricantes de software utilizaron esta forma de conectar la información de diferentes fases para elaborar productos CASE que soportaran todo el ciclo de vida del desarrollo del sistema.

Las herramientas CASE permitieron una integración de los requerimientos, especificaciones, diseño, generación de código, aplicaciones, pruebas, implementación y mantenimiento. Estas surgen entonces como una forma de integrar, dinamizar y analizar la información generada en el desarrollo de un sistema.

CASE prosiguió su desarrollo tecnológico, aumentando con esto sus habilidades y capacidades, adicionando técnicas de programación en tiempo real y obteniendo alguna parte del código a partir de los diagramas y especificaciones obtenidas. Además, los fabricantes de software descubrieron que la utilización de sistemas expertos desarrollados en el área de Inteligencia Artificial (IA) podían acortar una parte del desarrollo de sistemas.

Con la Inteligencia Artificial los sistemas son representados como modelos lógicos, desde los cuales pueden ser generados automáticamente. Con sistemas de IA se pueden entonces probar y validar diagramas sumamente grandes y complejos, y aún el crear las aplicaciones con éstos. Esta fue una de las formas que nevaba al pasar desde la determinación de requerimientos, hasta la operación de un sistema.

1.2.2.1 Pre-CASE

Definiendo CASE en estos términos, podemos definir lo que llamaremos Pre-CASE, el cual se refiere a herramientas que tratan con el nivel superior del desarrollo de un sistema, como pueden ser estrategias a tomar para el desarrollo de un sistema o el manejo de resultados esperados en el desarrollo del software. Pre-CASE representa las herramientas de planeación y determinación de estrategias, para que los usuarios contribuyan a determinar el alcance del sistema antes de empezar a realizar las actividades típicas de la ingeniería de software.

Las herramientas Pre-CASE están dirigidas hacia la planeación de estrategias, tales como aquellas actividades relacionadas para desarrollar un modelo de funcionamiento de una empresa en un sistema real (llamemos sistema real a aquella forma en la que una empresa se encuentra trabajando sin tener en cuenta si los procesos están bajo algún tipo de automatización), o el análisis de los sistemas manuales y automatizados con los que se cuentan en un momento dado, así como el análisis de las expectativas que se tienen para el momento en que el sistema real sea convertido en un sistema automatizado.

Pre-CASE se relaciona directamente con las funciones involucradas en la planeación estratégica de un proyecto, así como con las estimaciones y análisis de los resultados esperados.

1.2.2.2 Incorporación de la Tecnología de Bases de Datos

Un adelanto importante que se agrega a los productos CASE, es la tecnología de bases de datos. Los sistemas de bases de datos proveen el lugar donde las herramientas CASE pueden almacenar la información generada por las herramientas Upper-CASE y Pre-CASE, además de ser la forma de almacenar la información propia que fluye en el sistema.

La sección de la base de datos utilizada por los productos CASE se conoce como *diccionario* o *depositos*. Muchos diccionarios de CASE son manejadores de bases de datos relacionales (RDBMS o Relational Database Management System), los cuales utilizan lenguajes estándares para explotar la base de datos como puede ser el SQL (*Structured Query Language*).

Sin embargo, estos diccionarios pueden también usar otra estructura de bases de datos, tales como modelos jerárquicos o de red. Al igual que los vendedores de bases de datos desarrollaron productos CASE, los vendedores de CASE desarrollaron bases de datos. Estos diccionarios de CASE usados para el desarrollo de sistemas y aplicaciones estaban regularmente separados de la base de datos que se utilizaba para almacenar los datos del sistema en producción.

Gradualmente, en 1985 y 1986 los vendedores de software comenzaron a reconstruir los diccionarios incluidos en sus herramientas CASE, utilizando las bases de datos que comercialmente eran importantes. Una vez que los vendedores hicieron esos cambios, los diccionarios de las herramientas CASE pasaron a ser la misma base de datos que utilizaban los desarrolladores. Al igual que la tecnología de bases de datos fue muy importante para el desarrollo de herramientas CASE, algunas de estas herramientas han utilizado tecnologías orientadas a datos para desarrollar aplicaciones utilizando diagramas entidad-relación, normalizadores de datos y generadores de programas en SQL. Con estas propuestas, el núcleo total a base de datos de producción es parte del proceso para diseñar aplicaciones de software.

Teniendo una apropiada construcción de la base de datos de producción se agudiza el proceso de desarrollo de aplicaciones. Las herramientas CASE están para apoyar al análisis de sistemas y a los desarrolladores en la construcción de un modelo de datos, que represente el funcionamiento de la empresa.

Los vendedores de software tomaron ventaja de la utilización de las bases de datos para desarrollar un método de herramientas enfocadas a esta tecnología. Se crearon generadores de formas, reportadores, diccionarios de datos para trabajar con programas de la generación. Esto permitió a los programadores el desarrollo de prototipos utilizando la información contenida en el diccionario de datos y chequeando esta producción de programas.

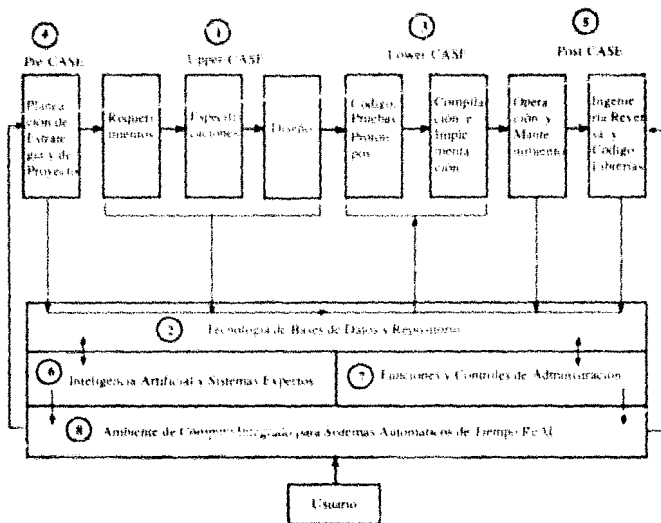
1.2.2.3 Post-CASE.

Con la tecnología de base de datos estabilizada como una parte de la estructura de CASE, los desarrolladores necesitan herramientas para transferir códigos existentes a un ambiente CASE, así como dar mantenimiento a los sistemas existentes. Códigos estructurados pueden ser almacenados para su reutilización como módulos de programas futuros. Estos códigos no están estructurados de la misma forma que aquellos que han sido generados en un ambiente CASE, por lo tanto es necesario hacer este proceso en sentido inverso para convertirlo a un código estructurado y guardarlo en un ambiente CASE. A este proceso se le llama Post CASE. El siguiente paso en el proceso de evolución de CASE consiste en juntar todas las piezas para tener lo que llamaremos un sistema integrado CASE.

Los vendedores han comenzado a integrar todas las funciones en productos CASE actuales, incluyendo módulos para el soporte de programación de procesos, las acciones no automáticas y operaciones de control. Estos son solo algunos de los muchos posibles módulos que pueden ser integrados en productos CASE para soportar un completo ciclo de vida del software. Esta integración modular ha dado un nuevo significado a CASE. El acónimo CASE

es utilizado en conjunto con las herramientas, hardware, metodologías y ambientes, dependiendo de cuántas funciones y tareas son integradas, cubriendo así casi todos los aspectos de la ingeniería de software.

La estructura completa puede visualizarse en la siguiente figura.



I → R el camino de la realización e integración de CASE

→ flujo de datos e información

< CICLO DE VIDA DEL SOFTWARE >

1.3 Tendencias de CASE

La evolución de CASE sigue desarrollándose en todos los aspectos de la ingeniería del software, creándose así el concepto de *fábricas de software*, ya que todas las actividades incluidas en este proceso caen bajo el concepto de CASE. Como se ha visto, podemos decir que CASE es más que un generador de códigos, es una herramienta que ayuda en el análisis completo del funcionamiento de una empresa y en desarrollar un sistema de cómputo que automatice parte de estas funciones.

Los productos CASE del futuro automatizarán la mayoría de las funciones técnicas y gerenciales, planeación de *estrategias y tácticas* involucradas en el desarrollo de software. CASE es una combinación de herramientas de software y de metodologías de desarrollo estructurado de software. Las herramientas automatizan el proceso de software y la metodología define el proceso a ser automatizado. Concluyendo podemos decir, que CASE es simplemente definido como la automatización del software.



CAPÍTULO 2

ESTUDIO DE CASE*METHOD

Para realizar el desarrollo de un sistema de cómputo bajo la filosofía CASE se necesitan tener básicamente dos cosas: herramientas CASE y una metodología completa y bien definida.

En el presente capítulo se dará un panorama general sobre lo que es la metodología y posteriormente se proporcionará una descripción detallada de cada una de las siete etapas que la conforman.

Se resaltarán los objetivos, las actividades más importantes, los factores críticos para lograr el éxito así como los productos resultantes de cada etapa o subetapa.

Es importante tener siempre presente, que una metodología no lo es todo en el desarrollo de sistemas. Aunque CASE*Method nos proporciona una manera ordenada y estructurada para el desarrollo de proyectos, no es necesario seguir exactamente cada etapa conforme a lo que estipula la metodología. Aquí prevalece el sentido común, la experiencia y la capacidad de razonar de todos los involucrados en la creación del sistema.

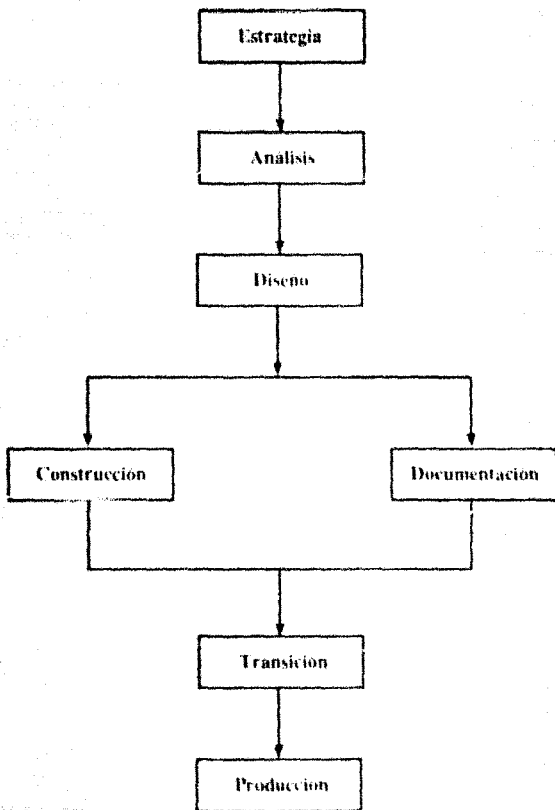
2.1 Descripción de CASE*Method

CASE*Method es la formalización de una serie de etapas estructuradas que van desde la *Estrategia* hasta la *Producción*, con el fin de dar soporte al desarrollo de algún sistema durante todo su ciclo de formación.

Las diferentes etapas por las que tiene que pasar cualquier sistema planeado y correctamente desarrollado son:

<u>Etapas</u>	<u>Descripción</u>
- Estrategia	determinar lo que se debe de hacer, el tiempo y el costo aproximado para realizarlo.
- Análisis	especificar las necesidades del usuario.
- Diseño	decidir cómo debe trabajar el sistema, y qué herramientas y componentes deben ser utilizados.
Implementación	
- Construcción	desarrollar el sistema.
- Documentación	escribir la documentación para el usuario.
- Transición	implementar acciones para pasar del sistema viejo al nuevo.
- Producción	utilizar el sistema y crear un ambiente para mantenimiento y mejoras posteriores.

El siguiente diagrama muestra la secuencia lógica de las diferentes etapas en el desarrollo de un sistema.



<ETAPAS DEL DESARROLLO DE UN SISTEMA>

El desarrollo de sistemas hoy día está lejos de ser sencillo y rápido. Existen muchos problemas asociados a la producción de sistemas como lo pueden ser la falta de dirección, el rápido desarrollo tecnológico, requerimientos conflictivos, diferencias en el estilo de desarrollo, la existencia de múltiples soluciones posibles, la falta de control y calidad y muchos otros más.

Tal vez los problemas más serios a los que se puede uno enfrentar son los relacionados con la gente misma: los usuarios muchas veces no saben siquiera lo que desean tener o lo que necesitan, hay problemas de comunicación - todos saben hablar, pero pocos saben escuchar - y el significado de las palabras difiere de individuo a individuo. También es muy común que el desarrollador produzca lo que él cree que el usuario necesita, y no lo que efectivamente se requiere crear. También se dan muchos casos en los que el desarrollador pretenda seguir trabajando en un esquema tradicional y arcaico y se niega a colaborar con la tecnología nueva.

CASE Method provee un acercamiento estructurado, detallado y metódico para asegurar el éxito en el desarrollo de sistemas.

Todos los pasos a seguir se agrupan en las siete etapas mencionadas, y cada etapa arroja una serie de puntos esenciales, necesarios para seguir definiendo y construyendo el sistema en etapas posteriores a ésta. La llave del éxito de un sistema es que tanto se puede este definir y adaptar en términos del mundo real, y en el entendimiento de cada detalle del funcionamiento de una empresa o de un organismo, antes de seleccionar alternativas de diseño y de tecnología.

Es importante mencionar que una metodología aunque pueda ser muy completa y estar bien definida, no debe de utilizarse como una receta de cocina. Ninguna metodología sustituye la capacidad de pensar de un ser humano. Un buen programador, diseñador o analista no se conformará solamente con hacer uso de ciertas reglas y de un procedimiento mecánico. Usará su poder de razonamiento y de deducción, sin caer en una rutina y sin dejar a un lado su capacidad creativa.

Es de vital importancia que el producto resultante de la etapa de *Estrategia* y del *Análisis* sea completamente independiente de alguna herramienta existente en el mercado, así como de cualquier computadora y sistema operativo. Esta independencia asegura que la inversión hecha en el desarrollo de un proyecto grande, queda protegida también en un futuro si se decide partir la aplicación de una plataforma a otra.

1.1.1 Desarrollo de proyectos

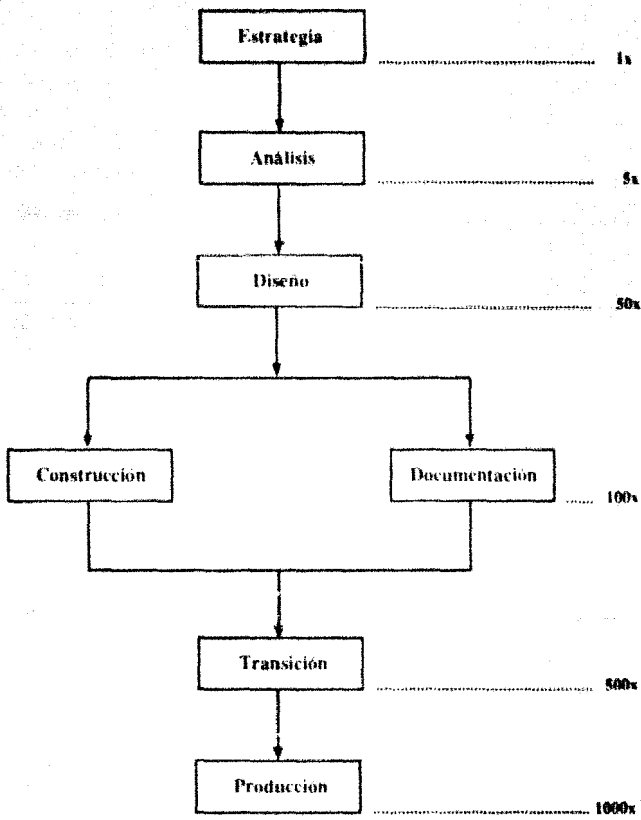
Cada proyecto tiene sus características particulares. No existen proyectos estándares, por lo cual es a veces difícil determinar cuál será el punto de partida para el desarrollo y su creación. No solo eso, normalmente existe una gama de posibles soluciones para cualquier problema o necesidad, y se vuelve difícil señalar cuál de todas es la mejor o la más apropiada.

Hay que realizar un estudio para identificar lo que se quiere hacer, para poderlo organizar en un plan de trabajo global, compuesto por una serie de etapas que constan a su vez de diversas actividades, con fecha de inicio, término y control de cada actividad.

La necesidad de estimar el trabajo, dividirlo y supervisarlo es sumamente importante. Hay que tener constantemente información del progreso obtenido y de los problemas encontrados. Asegurar la calidad del proyecto, por medio de revisión de planes y con la ayuda de reportes de actividades realizadas, será imprescindible. Finalmente, hay que mantener informados a todos los involucrados en el proyecto sobre el estado de avance del mismo. Un diagrama de tiempos para el control de cada etapa es siempre de gran ayuda para llevar un mejor control sobre las actividades a realizar.

Es más simple corregir anomalías o definiciones incorrectas cuando se ha hecho sólo un 10% del trabajo total, que cuando ya se realizó un 90%. En el desarrollo de sistemas es más fácil prevenir que corregir.

La figura que aparece en la siguiente página muestra el costo relativo que representa hacer una modificación durante las diferentes etapas del desarrollo de un sistema.



< COSTO RELATIVO EN LAS ETAPAS DEL DESARROLLO DE UN SISTEMA >

De la figura se puede ver que la diferencia entre la etapa de *Estrategia* y la de *Producción* es de mil unidades. Esto muestra la importancia de identificar cualquier anomalía o inconsistencia que se encuentre desde las etapas más tempranas del desarrollo de una sistema. De no ser así, cambiar una especificación durante alguna de las etapas finales representará mucho trabajo adicional, un mayor costo para la realización del proyecto y muchos disgustos entre todos los involucrados.

2.2 Etapa de Estrategia

2.2.1 Objetivos y descripción

La etapa de estrategia es la etapa inicial en el desarrollo de un sistema. Su objetivo es producir un conjunto de modelos y de recomendaciones, así como también un plan estipulado para el desarrollo de sistemas de información, siempre con la ayuda de los usuarios. Estas estipulaciones estarán en función de las necesidades actuales y futuras de una empresa, y tomarán en consideración restricciones financieras, técnicas y generales de la organización.

Para lograr esta meta es necesario tener una buena relación con el universo de usuarios, de tal manera que éstos cooperen gustosamente.

Hacer un análisis completo y detallado de una organización sería una base excelente para el desarrollo posterior del sistema a implantar, pero resulta ser muy caro. En lugar de eso, se realiza un análisis completo, mas no detallado, que sirve para construir un modelo base del organismo.

Los resultados obtenidos de la etapa de estrategia deberán ser revisados por el usuario, para que éste constate que el modelo propuesto se adapta al mundo real.

La manera de trabajar aquí es utilizando el conocimiento de la gente que conoce muy bien el funcionamiento de cada área en particular, para poder derivar todas las funciones que realiza dicho organismo, así como para determinar las necesidades de informaciones del mismo.

El esquema que se sigue es uno que va de lo general a lo particular, denominado *arriba-abajo* o *top-down*, por lo tanto se debería de comenzar desde un panorama general del organismo, resultando sus objetivos primordiales, hasta lograr un modelo completo que abarque todo. Es una etapa corta pero intensiva de recopilación de información.

2.2.2 Actividades principales

Administración y control del proyecto

Esta es una actividad que engloba control, revisión, realización de reportes y actividades administrativas referentes al plan de la segunda y tercera actividad. Esta es una actividad que se repite varias veces. Hay que comparar el progreso con el plan estipulado, revisar la calidad, controlar el funcionamiento y desempeño de cada participante, generar reportes sobre el progreso obtenido y finalmente reconfirmar pautas y compromisos con los usuarios.

Dimensionar el estudio

Aquí se trata de dimensionar la etapa y la manera de trabajar. Hay que tomar en cuenta la cultura de la organización, límites de horario, restricciones por seguridad, etc. Hay que definir el contenido del reporte de la estrategia y estimar el número de entrevistas y de sesiones de retroalimentación. Se dan lineamientos a seguir y se delegan responsabilidades.

Planear un estudio de estrategia

Esta es la preparación del estudio completo: se identifican áreas de investigación, se realizan lecturas de documentos diversos, se definen las entrevistas globales y las de detalle, y se registran en un calendario con fecha y hora. Es importante dedicar tiempo para estudiar los esquemas de jerarquía del organismo, ver sistemas existentes, reportes anuales y otros, dedicados a proporcionar información sobre diferentes asuntos.

Entrevistas y obtención de información

Durante esta actividad es necesario prepararse para las entrevistas, de modo que habrá que documentarse previamente sobre generalidades del organismo y preparar los cuestionarios. Se realizan las entrevistas y se resumen posteriormente, tratando de identificar los objetivos principales. Un modelo jerárquico de funciones y un diagrama de entidad-relación son preparados. El modelo jerárquico de funciones sirve para mostrar las funciones de cualquier organismo y las dependencias entre las mismas. El diagrama de entidad-relación es un esquema que muestra las relaciones que existen entre las diferentes entidades.

Finalmente hay que ir registrando las diferentes entidades, sus sinónimos, significados y atributos. A los atributos de mayor importancia se les asigna un dominio.

En esta actividad se ha en básicamente entrevistas y modelaje, información que puede ser ya registrada en un diccionario común de datos.

Es de suma importancia que los entrevistados no sean solamente usuarios finales, sino personas que conozcan la estrategia futura del organismo.

Modelaje del organismo

Hay que identificar los objetivos del organismo, sus prioridades, limitantes, sus puntos fuertes y los débiles, las oportunidades e indicadores críticos para lograr el éxito. Se siguen ampliando las funciones en el modelo jerárquico de funciones y se completan las definiciones de entidades y sus interrelaciones. De gran utilidad es llevar a cabo un revisión cruzada de entidades y funciones.

Es importante que se muestre qué es lo que hace el organismo, independientemente de cómo y quién lo haga.

Preparación de sesiones de retroalimentación

Se preparan los materiales para la presentación como son los diagramas de entidad-relación, el jerárquico funcional y el de futuras tendencias de dirección.

Notese que las sesiones de retroalimentación conforman la parte más importante del estudio, no son sesiones de presentación de resultados finales, sino son el camino a resolver dudas existentes y/o corregir concepciones erróneas de los modelos.

Se tendrán tantas sesiones como sean necesarias hasta llegar a un consenso general.

Sesiones de retroalimentación

Durante las sesiones de retroalimentación hay que tener en cuenta los objetivos de la junta, para que se logre lo propuesto. Se presentan los resultados obtenidos, los modelos propuestos y se dan ejemplos prácticos para ver si los modelos cumplen su cometido. Se registran cambios por hacer y hay que asegurarse que todos entiendan todo y que haya representatividad de todos los sectores asociados con el modelo.

Consolidación de resultados de sesiones de retroalimentación

Hay que asegurar que las modificaciones propuestas sean incorporadas a los modelos propuestos. Se hace un revisión posteriormente para verificar la consistencia en todo y en caso necesario incluir nuevos comentarios.

Completar los planteamientos del modelo

Registrar toda la documentación en un diccionario de datos es muy importante, para que ésta se encuentre en un formato predefinido y quede accesible para etapas posteriores. Se registran entidades, relaciones y atributos en el diccionario.

Desarrollo de la arquitectura del sistema

Se identifican necesidades y prioridades del organismo, se seleccionan áreas de aplicación y el contexto del sistema en términos del modelo funcional. Se pueden identificar posibles tecnologías, el hardware, el software y paquetes a utilizar. Se puede dar una visión de cómo trabajará todo en conjunto, cómo funcionará el sistema, qué recursos se necesitarán, qué factores serán críticos para el éxito y cómo se medirá este.

Se propone una estrategia global de trabajo, a la vez que se dimensiona el proyecto completo.

Determinar el plan de desarrollo

Se divide el desarrollo total en unidades de trabajo con actividades predefinidas. Es necesario hacer estimaciones de recursos para cada actividad, así como de duraciones.

El plan es detallado para la siguiente etapa y constituye una buena referencia para etapas subsiguientes.

Preparación del reporte verbal

Esta actividad se dedica a preparar todo el material posible para exponer los resultados finales de la etapa de estrategia. Un reporte detallado que contiene información resumida, los modelos propuestos, comentarios y sugerencias, es generado.

Reporte final verbal

Se convoca y realiza una presentación formal de todo el trabajo realizado durante esta etapa y de las conclusiones definitivas. El objetivo de esta presentación verbal es llegar a la aceptación del sistema propuesto por parte de las personas que finalmente harán el proyecto.

Se presenta el plan de trabajo, el costo y los beneficios asociados a la estrategia propuesta. Aquí se aclara cualquier duda existente con respecto al modelo y forma de trabajar.

Preparación y entrega del reporte escrito

Se asienta todo lo expuesto durante el reporte final verbal en un documento. Es aconsejable

corregir errores de consistencia, gramática, ortografía y de estilo.

No existen tópicos nuevos en este reporte a esta altura de la etapa, todo ha sido dicho y aceptado como válido. Es importante que el documento sea fácil de entender y no contenga conceptos ambiguos o confusos.

2.2.3 Factores críticos para tener éxito

Durante esta etapa corta e intensa de recopilación y filtrado de información, se requieren cuidar los siguientes factores, para asegurar el éxito de la primer etapa del desarrollo:

- Contar con la participación de usuarios que tengan un profundo conocimiento de las funciones y necesidades de la organización, así como la colaboración de ejecutivos que conozcan los planes futuros.
- Correcciones oportunas de ideas, de opiniones, y de los modelos propuestos.
- Sesiones detalladas de retroalimentación.
- Tomar en cuenta el trabajo ya hecho y sistemas existentes.

2.2.4 Productos resultantes

Los productos obtenidos durante la etapa de estrategia son:

- Diagrama entidad-relación.
- Modelo jerárquico funcional.
- Recomendaciones.
- Definición de limitantes del sistema.
- Arquitectura del sistema.
- Plan de desarrollo en fases.
- Indicación de recursos.

2.3 Etapa de Análisis

2.3.1 Objetivos y descripción

Durante la etapa de análisis se toman y verifican los modelos y conclusiones de la etapa previa, para ampliarlos y lograr una descripción aun más detallada, que garantice un modelaje exacto de la realidad. Esta etapa es utilizada para documentar todos los atributos, explorar nuevas dependencias, establecer nuevas condiciones y el uso de los datos, así como de *asentar la lógica del sistema*.

Elementos que afectan la transición deben ser entendidos detenidamente. Limitantes y consideraciones que influyan durante el diseño o con el tiempo planeado para el desarrollo, deben ser identificadas.

Analistas y usuarios trabajan conjuntamente para establecer y revisar con detalle cada requerimiento. El modelo propuesto debe ser reflejo claro de la realidad.

2.3.2 Actividades principales

Administración y control del proyecto

Durante esta actividad se revisa el plan y se mantiene el avance con respecto a lo estimado. Se revisa el desempeño de cada participante y se generan reportes del progreso obtenido. Se provee de soporte administrativo general y se mantiene en estricto control de calidad en el trabajo de los analistas. Se aconseja organizar sesiones de trabajo regularmente, para que todos estén enterados de los avances y cambios de especificaciones.

Planear el análisis detallado

Es indispensable acordar la manera de trabajar durante toda la etapa, y el tiempo estimado para finalizarla. Se fijan las entrevistas.

Es importante que se traten puntos básicos, como pueden ser interfaces con otros sistemas, problemas, limitantes y procedimientos de la vida real, sin perder de vista los objetivos primordiales de la organización.

Se tratan de identificar nuevas fuentes de información y se decide quién se hará responsable del control de problemas, de reportes, de la calidad y de otros procedimientos.

Revisión de entidades, restricciones y características potenciales de diseño

Se llega a un convenio sobre la nomenclatura a usar y sobre el contenido de los documentos a entregar. Es necesario hablar sobre tecnologías preferidas, instrumentación de requerimientos de seguridad, sobre un posible manejador de la base de datos u otros aspectos de tipo organizacional.

Investigación de requerimientos detallados

Se analiza a fondo y con lujo de detalle el diagrama entidad-relación y el modelo jerárquico funcional con la ayuda del usuario.

Se realizan entrevistas detalladas. En el modelo jerárquico funcional se revisa que las funciones sean entendibles y estén completas, incluyendo también casos especiales.

Se hace una referencia cruzada entre entidades y funciones. Se recolecta información relacionada a volúmenes de entidades y frecuencias de cada función.

De gran valor es la mención de valores permitidos para ciertos atributos, la derivación de valores, restricciones de integridad, así como procedimientos de validación.

Revisión de descubrimientos

Esta actividad realiza la revisión de los hallazgos de problemas y de las ideas de la actividad anterior, para asegurar el éxito de la etapa. En caso de toparse con alguna incongruencia o limitante para la futura evolución del proyecto, hay que cambiar el enfoque del problema, revisar las ideas originales o darle la vuelta con nuevas alternativas tecnológicas; el usuario debe ser informado sobre la situación actual.

Proveer descripción detallada de entidades

Hay que registrar todas las descripciones detalladas de entidades y atributos, así como cerciorarse de que se han encontrado todos los atributos para cada entidad.

Hay que definir y encontrar todas las dependencias existentes debidas a eventos, al uso de datos o por alguna constante de tiempo. Hay que acordar el grado de automatización para cada función y mencionar procedimientos manuales pendientes.

Proveer estrategia inicial de transición

Esta actividad considera diferentes aspectos a realizar en paralelo con la recopilación de

datos y funciones.

Se define un plan de entrega y aceptación para el proyecto, un plan de documentación, se hacen estimaciones sobre el trabajo a realizar, un plan de instalación y transición, así como un recuento de factores críticos para el éxito del sistema.

Definir necesidades de control y auditoría

Es necesario definir requerimientos legales y límites al sistema, acerca de seguridad en el acceso de la información y en cualquier aspecto en el que haya la necesidad de hacerlo. Hay que derivar controles de integridad, controles sobre las aplicaciones y referencias a las auditorías, así como definir el número de errores.

Definir procesos de respaldo y recuperación

Hay que especificar las acciones a seguir en caso de que se presente una falla del sistema. Se definen los procesos de respaldo de la información, de la recuperación y el almacenamiento de datos. Es muy común que un sistema pueda dejar de trabajar por irregularidades mecánicas o eléctricas. Si no se posee un sistema tolerante a fallas, hay que tomar estos factores muy en cuenta para poder dar un servicio ininterrumpido.

Búsqueda del tamaño y predicción de la ejecución

Predicir el tamaño total de la base de datos es de gran ayuda, así como indicar posibles tasas de crecimiento. Predicir la ejecución de las diferentes funciones del sistema también ayuda para que el usuario tenga una idea del sistema total, así para poderlo comparar contra el sistema final. En donde exista algún problema para lograr las expectativas de crecimiento, habrá que proponer alternativas tecnológicas nuevas.

Revisión de resultados del análisis detallado

Esta actividad está dirigida a revisar los detalles adicionales propuestos para la solución. Hay que revisar que los objetivos originales se sigan respetando en términos de otros sistemas, de la organización y de las expectativas de los usuarios. Los modelos deberán estar completos, satisfacer la consistencia de la información y cumplir con el mundo real.

Acordar el fin de la etapa

Se acordará el contenido de la documentación que se debe entregar en esta etapa y que ser-

virá como base para la etapa de diseño

Hay que revisar los costos y producir un plan detallado para la siguiente etapa. Se produce un reporte final y se presenta a todos los involucrados, de ser necesario. Hay que obtener posteriormente la aceptación del análisis por los interesados en el sistema.

2.3.3 Factores críticos para tener éxito

Durante esta etapa de profundo y detallado análisis se tienen que cuidar ciertos factores para asegurar el éxito de esta etapa de análisis:

- Ayuda activa de los usuarios
- Revisión exhaustiva de calidad y terminación
- Identificación de los puntos claves para el diseño y la transición
- Estimaciones certeras de volúmenes de información
- Definir el nivel de calidad que implica la palabra *definiendo*.

2.3.4 Productos resultantes

- Diagrama final de entidad-relación
- Detalle descrito en el análisis de funciones.
- Volúmenes de datos, frecuencias de funciones y expectativas de la ejecución.
- Definición del estilo de trabajo
- Estrategias iniciales de transición
- Necesidades de control, recuperación y respaldo
- Descripción de procedimientos manuales.
- Criterio de aceptación del usuario

• Fundamentos y suposiciones

• Esquema de trabajo para las etapas de diseño y construcción

• Aceptación del plan de desarrollo

2.4 Etapa de Diseño

2.4.1 Objetivos y descripción

La etapa de diseño tomara las especificaciones detalladas de los requerimientos provenientes de la etapa de análisis para llevarlos a cabo de la mejor manera posible y para lograr el nivel de exactitud prometido. Se genera en esta etapa los planos de hardware y software que se usaran en el desarrollo.

El modelo de entidad-relacion sera transformado a un diseño de base de datos y a otras especificaciones, que no necesariamente corresponden a la base de datos.

Las funciones del modelo entidad-relacion seran traducidas a modelos y a procedimientos manuales con las características elegidas de control, respaldo y de recuperación de datos. El desarrollo de prototipos puede ser empleado para tomar decisiones en áreas de duda.

Finalmente se producen las especificaciones del programa y un plan de pruebas del sistema. Analistas, programadores y diseñadores de bases de datos trabajan conjuntamente en esta etapa y se pide la intervención del usuario en caso de haber multiples opciones en la solución del problema.

2.4.2 Actividades principales

Control y administración del proyecto

Se mide el progreso con respecto al plan estipulado y se generan reportes del avance logrado durante las diversas actividades de la etapa. Es necesario verificar los recursos de la calidad del trabajo realizado. El líder del proyecto, esta encargado de controlar el desempeño de los involucrados en esta etapa, así como de generar si tienen algún problema.

Se organizan juntas para analizar y reportar el progreso así como para obtener información adicional. Esta actividad también sera usada para proveer soporte administrativo general tal

como organizar las juntas, reservar cuartos, etc.

Diseño

Durante esta actividad se transformarán las funciones a módulos y se proveerán sus especificaciones. Hay que seleccionar las funciones que se instrumentarán en el sistema de cómputo y se decidirá sus estructuras.

Se diseña y documenta la arquitectura del sistema, las estructuras de los menús, las pantallas, los procesos batch, los manuales de procedimientos, las clases de los usuarios, las interfaces del usuario, los menús y los reportes. Es necesario mantener toda una comunicación con los usuarios, auditores, administradores de la base de datos y con todo el equipo técnico para ver si les parece adecuado el marco del sistema.

Para cada módulo se tendrá que definir y documentar lo siguiente: el detalle del procesamiento, el uso de entidades y atributos, los procesos de modificaciones a la información, los procedimientos de control y validación, así como los de respaldo y recuperación de la información, procedimientos batch y de corrección de errores, los procedimientos excepcionales y los especiales.

Finalmente se genera una referencia cruzada entre los módulos y el diagrama de entidad-relación para verificar que todo esté completo.

Diseño y construcción de la base de datos

Esta es una actividad muy importante, ya que se crean las tablas y los archivos necesarios para el sistema, en base a las entidades existentes. También se crean los índices y las vistas. Todo se hace considerando la óptima ejecución de las funciones y de acuerdo a las posibilidades de guardar cierta cantidad de información en disco. Se realizan predicciones necesarias de conversión de datos y se optimiza el diseño de la base de datos o de los archivos. Se hace un análisis detallado de capacidades y volúmenes.

Una vez acordados todos estos puntos, se lleva a cabo la creación final de la base de datos o de los archivos.

Para lograr el mejor diseño posible hay que tener en mente, que éste se hará en función de las actividades más representativas e importantes del organismo, siguiendo el criterio que a continuación se menciona:

- las 10 funciones más críticas
- las 10 funciones más complejas
- las 10 funciones más urgentes
- 20 funciones adicionales seleccionadas al azar

Diseño de comunicaciones y red

Esta actividad está enfocada a diseñar la arquitectura de red que se empleará, y a resolver los procedimientos para soportar la red requerida.

Se establecen las localidades de las computadoras, los tipos de terminales y procesadores que se utilizarán, así como la arquitectura de red y los protocolos de comunicación necesarios. No hay que perder de vista que el diseño de la red deberá ser sencillo de configurar y con capacidad de un crecimiento futuro. También hay que identificar otros requerimientos de comunicación, como por ejemplo soporte a módems, fax y otros equipos de automatización de oficina.

Es importante predecir el tráfico en la línea, sobre todo durante las horas pico y analizar el impacto que tendrá sobre perifericos y sobre los procesadores.

Hay que revisar que haya mecanismos adecuados para el diagnóstico de errores y su corrección, para lograr el óptimo manejo de red y en caso de no haberlos, poderlos diseñar.

El usuario será consultado sobre los procedimientos realizados y las decisiones tomadas para obtener su aprobación, en caso de que éste no quede conforme, habrá que revisar nuevamente la base de datos y el diseño de los archivos.

Finalmente se crearán matrices de nodo/procesador, base de datos/nodo, usuarios/nodo, localización geográfica/nodo, etc. para verificar que todo haya quedado incluido.

Diseño de necesidades de control

Se trata de diseñar las necesidades de revisión y acceso identificadas durante la etapa de análisis.

Se diseñan pruebas para validar las necesidades legales y de la empresa. Se archiva la documentación de los planes de pruebas del sistema.

Se documenta la seguridad para el acceso de cuentas, programas, tablas y campos que se lleva a cabo por medio de claves de acceso.

También se documentan otros tipos de medidas de seguridad como pueden ser el restringir el acceso a los cuartos donde se encuentran las computadoras, protección de los programas fuente y de los respaldos, el encriptar la información, etc.

Especial cuidado habrá que tomar para revisar y garantizar la integridad de los datos.

Disaster Recovery Plan y de respaldo y recuperación

Se documentan los procedimientos de respaldo y recuperación de la información manejada por el sistema, haciendo énfasis en casos especiales que pueden ser tabla de la acción. Se deben de tomar en cuenta las funciones del sistema en el caso de una recuperación de datos y determinar el tiempo posible que tardara el sistema en estar nuevamente disponible parcial o totalmente.

Se revisan las especificaciones de los módulos definidos y de ser necesario se incluyen nuevas especificaciones a estos.

La documentación de la operación del sistema se complementa con un esquema general de programas y datos, tareas y con los ciclos de cada proceso.

Revisión del diseño y producción de especificaciones del programa

Esta actividad pretende llevar a cabo una revisión completa de las especificaciones de los diferentes módulos para ver que cumplan con las necesidades del sistema.

Se revisan las definiciones de módulos, el diseño de la base de datos y de los archivos, las interfaces con el usuario, el diseño de comunicaciones y de red, así como los mecanismos de control, respaldo y recuperación de la información. Se confirman los niveles de servicio, las políticas de almacenamiento, así como las necesidades centralizadas y distribuidas.

Se identifican los módulos a ser implantados por técnicas incrementales de desarrollo (por ejemplo por prototipo) y los desarrolladores que puedan crear los diferentes módulos, en base a la experiencia de varios uno de ellos.

Finalmente se preparan las especificaciones detalladas para la implementación de cada módulo que no será sometido a un desarrollo incremental.

Completar el plan de pruebas del sistema

Durante esta actividad se derivan las condiciones de prueba apropiadas para esta etapa: la validación de datos, archivos vacíos y criterios de ejecución apropiada.

Se revisan las notas hechas durante el análisis y se producen pruebas para probar la entrada, salida y ejecución de cada programa. Se hace especial énfasis en necesidades de la empresa y en la integridad de la información.

Se diseñan pruebas para los procedimientos interactivos, batch, de recuperación y respaldo, de ejecución satisfactoria y para probar la dependencia de módulos diferentes. Se busca obtener la aceptación de estas pruebas por parte de los usuarios. Finalmente se diseña el plan de pruebas.

Confirmación de los planes

Se confirma la entrega y la aceptación de los planes, el plan de entrenamiento a usuarios y operadores, el plan de recolección y/o conversión de datos, el de procedimientos de respaldo y recuperación de información, el plan de instalación y de procesos operacionales. A cada uno de los diferentes planes se le asigna un responsable, para que lleve el control de cada actividad y del avance obtenidos.

Revisión de los resultados

Es necesario revisar que todas las decisiones requeridas hayan sido tomadas en cuenta, que todos estén de acuerdo con las mismas, que el diseño se adapte a las necesidades del organismo, que sea flexible a cambios futuros y que sea posible su realización con técnicas y herramientas existentes.

Hay que analizar el impacto que tendrá sobre otros sistemas, los usuarios y sobre el organismo en general.

A estas alturas se tiene que garantizar que todos los aspectos del sistema se hayan tomado en cuenta antes de comenzar con la etapa de construcción.

Obtener aprobación del término de la etapa

Hay que asegurar que las áreas de la estrategia de transición sean adecuadamente planeadas y controladas fuera del contexto del proyecto principal. Es necesario revisar los costos requeridos. Se determina el momento oportuno del cambio de versión de la base de datos actual a la futura. Se producen planes detallados para las etapas de construcción y documentación del

usuario. Se verifica el plan global para el resto del proyecto y se publica para mantener informados a todos los involucrados.

Se produce un reporte o una presentación de fin de etapa en caso de ser requerido y se obtiene la aprobación para seguir con el plan estipulado.

2.4.3 Factores críticos para tener éxito

El factor crítico de esta etapa consiste en crear un diseño apropiado que se ajuste a las necesidades del organismo y que sea técnicamente realizable. Para lograr esto es necesario:

- Conocer las capacidades del hardware existente
- Entender a fondo las necesidades del organismo
- Tomar medidas balanceadas
- Identificar y resolver problemas potenciales.

2.4.4 Productos resultantes

- Arquitectura del sistema
- Esquemas lógicos y físicos.
- Diseño de la base de datos y de los archivos.
- Estimación detallada de volumen.
- Especificaciones de programas
- Especificaciones de procedimientos manuales.
- Planes de entrega, de aceptación, entrenamiento, cambio de datos e instalación.
- Plan de pruebas al sistema
- Documentación de operaciones
- Plan revisado de desarrollo del sistema

2.5 Etapa de Construcción

2.5.1 Objetivos y descripción

Durante la etapa de construcción se codifican y prueban los programas, utilizando las herramientas adecuadas. Estas dependen del ambiente técnico existente y del tipo de programas por realizar, y van desde un lenguaje de programación de bajo o alto nivel, hasta lenguajes de cuarta generación y paquetes que facilitan la creación de aplicaciones.

Sin importar cómo se realice la programación, el proceso de construcción involucra planeación, diseño de la estructura de los programas, validación, pruebas de *particular a lo general* (de unidades), pruebas de *general a lo particular* (del sistema) y una manera disciplinada para hacer el trabajo y controlar las interacciones sucesivas de los programas.

El equipo de trabajo desarrolla los programas sin la intervención del usuario, de tal manera que es importante que cualquier alteración a las especificaciones sea reconocida y revisada por el analista o usuario, posteriormente.

La creación de manuales del usuario, la documentación del entrenamiento y la de operaciones en general es tratada en paralelo con esta etapa.

2.5.2 Actividades principales

Administración y control del proyecto

Esta actividad compara el avance obtenido con respecto al plan y monitorea el progreso de la etapa. Se revisa la calidad de la codificación y de los detalles de la ejecución exacta y que todo el trabajo se realice bajo estándares. Se supervisa el desempeño de cada participante y se organizan juntas con los representantes de los usuarios. Se participa en la revisión de calidad y se provee de soporte administrativo en general.

Preparar la etapa de construcción

Se estudia el plan produciendo al final de la etapa de diseño. Se estima el tiempo que se empleará para la creación de los programas, en base a las especificaciones y a su complejidad.

Se asigna el trabajo a los diferentes programadores de acuerdo a la función de cada programa: si son transacciones en línea, si se trata de módulos a desarrollar por prototipo, si son procesos en lote, etc.

Las herramientas de desarrollo son seleccionadas, así como todo el ambiente de trabajo. Se publican las fechas de realización y entrega de programas.

Se proporcionan las diferentes versiones de información y programas para las pruebas. Finalmente hay que instalar el hardware y software requerido para comenzar a codificar.

Revisión del diseño y de estimaciones

El diseñador y programador deben de revisar las especificaciones y los estimados para determinar si es posible realizar el trabajo bajo los supuestos del plan a seguir. Se resuelven dudas y se documentan las decisiones y los cambios a realizar. Si los programadores no tienen la suficiente experiencia para realizar el programa, se les ayuda en la codificación y en las pruebas a realizar.

Se generan datos iniciales para las pruebas y se anotan los resultados esperados, utilizando métodos como el conocido bajo el nombre *prueba de escritorio*.

Producción de programas

Se codifica y prueba cada programa. En el caso del desarrollo incremental, las pruebas se realizan continuamente, pero habrá que hacer una prueba final completa. Las modificaciones al código se llevan a cabo conforme a los resultados de ejecución de cada programa. A continuación se realizan pruebas sobre diferentes conjuntos de programas, entre los cuales haya ligas lógicas.

Finalmente se revisan las matrices de programas con filas, arcos, columnas y funciones para comprobar que todo este completo.

Preparar, hacer y revisar pruebas del sistema

El ambiente de pruebas del sistema es preparado: datos, programas, comunicaciones, equipo y soporte en general. Se llevan a cabo las pruebas especificadas, se documentan los resultados y se registran los errores y las deficiencias existentes.

Se corrigen los errores encontrados, se actualizan los procedimientos y se vuelven a hacer las pruebas hasta eliminar todas las deficiencias. Como último paso se incluyen los programas libres de errores en las librerías de programas probados.

Revisión de resultados

Esta actividad está enfocada a determinar si todos los involucrados en el sistema están con

venidos de su buen funcionamiento. Se revisa que todo esté completo y que exista integridad total. También hay que cercionarse de que las correcciones hayan sido realizadas de manera adecuada y que no tengan impacto negativo sobre el resto de los programas.

Obtención aceptación de fin de etapa

Se busca formalizar el fin de la etapa y pasar a la siguiente. Se revisan nuevamente los planes de entrega, de conversión de datos, de transición del sistema y los de instalación. Se completa la documentación del sistema y se dan lineamientos sobre el grupo de personas involucradas durante la transición. Se confirma la fecha de comienzo de la siguiente etapa.

2.5.3 Factores críticos para tener éxito

Los factores importantes para lograr una etapa exitosa son:

- Asegurar la calidad de las pruebas realizadas en un periodo corto de tiempo
- Identificar y reaccionar rápidamente a errores y deficiencias de ejecución de comunicaciones y a cuellos de botella debidos a entrada/salida y/o procesador.
- Afinar los programas o la base de datos
- Comprobar las expectativas

2.5.4 Productos resultantes

- Diseño de programas
- Base de datos afinada
- Programas finales corregidos
- Revisión de la estrategia de instalación
- Resultados de las pruebas del sistema
- Indicadores preliminares de ejecución

2.6 Documentación

2.6.1 Objetivos y descripción

El objetivo de esta etapa es la entrega de manuales del usuario y la documentación de la operación del sistema. Esta documentación debe ser lo suficientemente explícita como para soportar las actividades de prueba del sistema en la etapa de construcción, que se realiza a la par con ésta. Es indispensable proveer la documentación completa antes de que se lleven a cabo las pruebas de aceptación de la etapa de transición.

La mejor manera de hacer la documentación de operaciones se logra incluyendo información de los antecedentes del sistema. También es de gran utilidad, si los mismos operadores describen en sus palabras el procedimiento, ayudados por personal técnico.

De manera similar se preparan los manuales del usuario, haciendo mención de las descripciones de funciones, de formas, reportes y de pantallas involucradas, de tablas de validación, mensajes de errores y de información general apropiada para entender el funcionamiento y manejo del sistema.

Es benéfico mencionar los diferentes eventos y las operaciones que se llevan a cabo, haciendo referencia de cuándo y qué tan seguido pasa cada evento.

2.6.2 Actividades principales

Manejo y administración del proyecto

Hay que comparar el progreso logrado con respecto al plan estipulado. Es necesario revisar la calidad del documento, su legibilidad y garantizar que éste sea completo. Esta actividad también se encarga de monitorear el rendimiento de los involucrados y de ayudarles si lo requieren. Hay que reportar constantemente el progreso. De vital importancia resulta ser la facilidad con la que se pueda entender el documento - hay que tener en mente, que los usuarios finales normalmente no son técnicos expertos en el área del cómputo.

Finalizar documentación del usuario

Se revisa el manual de referencia del usuario de la etapa de diseño. Hay que acordar el formato del manual, recopilar todo el material requerido y asignar un responsable para cada

sección del manual.

Es requisito incluir puntos relacionados con las entradas del sistema, mensajes de error y acciones a tomar, así como incluir referencias a la información presentada. Existe la posibilidad de usar parte de la documentación hecha durante las pruebas del sistema. Otra posibilidad es crear una ayuda en línea, utilizando el sistema de cómputo. Hay que organizar la producción y distribución de los manuales resultantes. Finalmente se llega a un acuerdo sobre qué persona será la responsable de actualizar la documentación del usuario y bajo qué procedimientos.

Proveer la documentación

Entregar la documentación a los usuarios. Recolectar información adicional de retroalimentación durante las pruebas del sistema. Proveer instrucción y ayudar a los operadores hasta que puedan operar solos el sistema.

Asociar el fin de la etapa

En esta actividad se revisará una vez más la documentación y se pedirá que los usuarios y operadores certifiquen que todo está en orden.

Hay que proveer datos necesarios para la revisión del plan de la etapa de construcción, según los requerimientos de la séptima actividad.

2.6.3 Factores críticos para tener éxito

Es importante involucrar a los usuarios y operadores del sistema lo más posible, para asegurar que:

- La documentación sea apropiada y efectiva.
- Los usuarios y operadores deben de tener la confianza suficiente para operar el sistema sin intervención ajena.

2.6.4 Productos resultantes

- Documentación del usuario
- Documentación de operación.

2.7 Etapa de Transición

2.7.1 Objetivos y descripción

La etapa de transición comprende todas aquellas actividades necesarias para implementar el sistema, además de ser un periodo inicial de soporte para el mismo. Es muy recomendable el llevar a cabo esta etapa en una manera gentil y tratar de disturbar en un mínimo todas las operaciones y funciones que suelen hacerse en el organismo, dejando a los usuarios con la confianza suficiente en el sistema nuevo, para que lo comiencen a utilizar.

Se provee la capacitación de los usuarios y se les ayuda a familiarizarse con la nueva forma de trabajar.

Se hace el cambio de toda la información nueva, se instala toda la infraestructura de hardware y software, y se revisa una vez mas cada detalle, ya que es la parte más crítica de todo el proyecto.

El cambiar toda la información nueva implica actividades como la carga de los datos nuevos, hacer la conversión de archivos y/o bases de datos y realizar una serie de pruebas para garantizar que todo está funcionando correctamente.

También se revisa la ejecución apropiada del sistema y se corrigen errores y deficiencias que pueda tener. Los planes para el entrenamiento de usuarios y para la aceptación del sistema son tomados de las especificaciones creadas durante la etapa de diseño.

Las pruebas de aceptación son utilizadas para probar diferentes aspectos operacionales del sistema, como pueden ser dependencias, tiempos, la capacidad de aislar fallas, ejecución de volúmenes grandes de información, etc. Esto tiene como finalidad el hacer un simulacro del funcionamiento real del sistema manejado por los usuarios y con todo el soporte del personal especializado.

2.7.2 Actividades principales

Administración y control del proyecto

Se compara el progreso con respecto al plan estipulado. Se hace énfasis en los problemas existentes y se les da solución rápida. Se revisa el desempeño de cada participante y se le apoya en cualquier dificultad que lo haga detenerse. Se controla el cambio del sistema en todos sus aspectos y se reporta el progreso obtenido.

Se organizan juntas formales para tratar el avance logrado y casi diario se realizan diversas juntas informales. Finalmente se proveen servicios generales de administración.

Entrenamiento de usuarios

Se proporciona el debido entrenamiento conforme al plan elaborado. Se organizan las sesiones, se crea la lista de alumnos e instructores y se fijan las horas.

Se publica el contenido de los cursos y se imparten normalmente a un grupo selecto de usuarios, los cuales seguirán entrenando a su gente internamente. Se preparan las sesiones incluyendo material de exposición, guías, ejercicios, demostraciones y cuestionarios. Idealmente se crea una base de datos pequeña con información real.

Se imparte el entrenamiento y se realizan pequeñas pruebas para ver si todos están entendiendo. En caso negativo se pueden organizar más sesiones.

Finalmente se provee el mecanismo por el cual los demás usuarios recibirán la instrucción adecuada.

Preparar las pruebas de aceptación

Se revisan los planes de entrega y de aceptación de la etapa de diseño. Se ayuda en la planeación y preparación de las pruebas de aceptación, esto incluye: la evaluación del criterio de aceptación de los usuarios, proveer cualquier información o equipo requeridos, dar la opinión sobre la manera de probar el sistema para evaluar la conversión correcta de datos y procedimientos existentes, la manera en que se van a corregir errores encontrados, etc.

Proveer soporte a las pruebas de aceptación

Esta actividad se encarga de que los usuarios prueben los aspectos operacionales del sistema. Hay que ayudar a los usuarios a que efectúen las pruebas correspondientes, a que detecten fallas o hagan preguntas en aspectos desconocidos a ellos, que se independicen y sean capaces

de resolver problemas y preguntas. Poco a poco se van resolviendo todos los puntos del plan de pruebas para lograr su aceptación.

Conversion de informacion

Se prepara todo para convertir los datos que se tienen disponibles al formato del sistema nuevo, esto comprende recolectar datos requeridos para introducirlos a la computadora, crear programas de conversión de datos, proveer utilerías para meter grandes volúmenes de datos.

Acto seguido se realiza la conversión de la información y se hacen pruebas, por ejemplo se generan reportes para determinar si los datos han sido almacenados correctamente.

Se comparan los volúmenes de datos con respecto a los producciones realizadas y en caso de haber grandes fluctuaciones, se ajusta el diseño y la implementación de los archivos. Se busca finalmente obtener la aprobación de los usuarios.

Instalacion de hardware y software

Esta actividad provee las facilidades y productos requeridos para el sistema. Se instalan máquinas, aplicaciones, muebles, cuartos y el sistema de comunicaciones necesarias para el funcionamiento adecuado del sistema nuevo. Se revisa la instalación y se prueban todos los componentes adicionados.

Hay que asegurar que se cuenta con los materiales necesarios para trabajar correctamente, esto es tener papel, discos flexibles, cintas, cartuchos de tinta, etc. Se consulta al usuario sobre la terminación satisfactoria de las instalaciones.

Ensayos diversos

Se planean y ejecutan diversos ensayos que contribuyen a completar la etapa de transición. Se revisan los resultados de estos ensayos y según estos se realizan ajustes pertinentes al sistema, para mejorar aún más la funcionalidad, la ejecución o la facilidad de uso. Todo esto debe ser llevado a cabo bajo el estricto control de la gente encargada de hacer pruebas al sistema.

Preparar el paso al sistema nuevo

Se revisan aspectos como disponibilidad de usuarios y sistemas, tiempos de producción y tiempos muertos, etc. Se organizan juntas para resolver estos puntos cruciales de la transición y se confirman fechas y horas disponibles.

Antes de comenzar el cambio del sistema hay que asegurar que todos los casos previos hayan quedado completamente cubiertos y que haya plena aceptación por parte de los usuarios.

Se revisan requisitos de seguridad, aspectos operacionales y ayuda de tipo técnica. Finalmente se cargan los programas y datos, asegurando que todo haya sido exitoso y finalmente se pide la aprobación para realizar el cambio al sistema nuevo.

Pasar al sistema nuevo

Esta actividad realiza el paso definitivo al sistema creado. Se realiza lo estipulado en el plan de transición elaborado en la etapa de diseño. Hay que controlar las actividades del plan de transición y finalmente dejarlo funcionando en forma definitiva. Se declara el nuevo sistema como el actual.

Soporte durante periodo crítico

Se provee de soporte a los usuarios para utilizar el sistema nuevo, se les ayuda a manejarlo, se les dan consejos para minimizar diversos irritantes.

Se identifican, registran y corrigen errores en el manejo del sistema. Hay que llevar un control estricto de los cambios del ambiente nuevo. Se contabilizan las caídas del sistema, el número de fallas y se registra el nivel de servicios.

Revisión posterior a la implementación

En esta actividad se lleva a cabo una sesión de retroalimentación, donde se mencionan diferentes puntos relacionados con todo el desarrollo del proyecto y se atienden recomendaciones provenientes de todos los grupos interesados.

Se colecciona y se agrupa información obtenida de encuestas hechas a los diferentes grupos interesados en el sistema, para producir material de retroalimentación, concerniente a los métodos y estándares empleados, a la intervención de los que desarrollaron el sistema, a la organización del proyecto en general y al éxito final.

Se genera un reporte con toda esta información y con las recomendaciones. Se atienden puntos necesarios que hayan surgido de esta actividad y se busca recibir la aprobación de fin de la etapa para entrar a la producción.

El sistema es liberado y con esto finaliza la intervención de programadores y desarrolladores.

2.7.3 Factores críticos para tener éxito

Es importante no perder de vista los siguientes puntos para lograr una etapa exitosa:

- Asegurar que el entrenamiento es apropiado y efectivo.
- Asegurar que los usuarios hagan suficientes pruebas para comprobar la operación correcta del sistema y para que se familiaricen adecuadamente.
- Coordinar la implementación con que es compleja y crítica en cuanto al tiempo que tarda.
- Asegurar que los operadores y los usuarios técnicos sepan detectar anomalías del sistema y corregirlas.

2.7.4 Productos resultantes

- Material de entrenamiento
- Usuarios y operadores entrenados.
- Sistema funcionando al 100%
- Datos convertidos
- Reporte de revisión posterior a la implementación.
- Soporte a usuarios y operadores
- Documentación completa del sistema.

2.8 Transición

2.8.1 Objetivos y descripción

La etapa de transición comprende todas aquellas actividades necesarias para implementar el sistema, además de ser un período inicial de soporte para el mismo. Es muy recomendable el llevar a cabo esta etapa en una manera gentil y tratar de disturbar en un mínimo todas las operaciones y funciones que suelen hacerse en el organismo, dejando a los usuarios con la confianza suficiente en el sistema nuevo, para que lo comiencen a utilizar.

Se provee la capacitación de los usuarios y se les ayuda a familiarizarse con la nueva forma de trabajar.

Se hace el cambio de toda la información nueva, se instala toda la infraestructura de hardware y software y se revisa una vez más cada detalle, ya que es la parte más crítica de todo el proyecto.

El cambiar toda la información nueva implica actividades como la carga de los datos nuevos, hacer la conversión de archivos y/o bases de datos y realizar una serie de pruebas para garantizar que todo está trabajando correctamente.

También se revisa la ejecución apropiada del sistema y se corrigen errores y deficiencias que pueda tener. Los planes para el entrenamiento de usuarios y para la aceptación del sistema son tomados de las especificaciones creadas durante la etapa de diseño.

Las pruebas de aceptación son utilizadas para probar diferentes aspectos operacionales del sistema, como pueden ser dependencias, horas pico, la facilidad de aislar fallas, ejecución de volúmenes grandes de información, etc. Uno debe como finalidad el hacer un simulacro del funcionamiento real del sistema manejado por los usuarios y con todo el soporte del personal especializado.

2.8.2 Actividades principales

Administración y control del proyecto

Se compara el progreso con respecto al plan estipulado. Se hace énfasis en los problemas existentes y se les da solución rápida. Se revisa el desempeño de cada participante y se le apoya en cualquier dificultad que lo haga detenerse. Se controla el cambio del sistema en todos sus aspectos y se reporta el progreso obtenido.

Se organizan juntas formales para tratar el avance logrado y casi diario se realizan diversas juntas informales. Finalmente se proveen servicios generales de administración.

Entrenamiento de usuarios

Se proporciona el debido entrenamiento conforme al plan elaborado. Se organizan las sesiones, se crea la lista de alumnos e instructores y se fijan fechas.

Se publica el contenido de los cursos y se imparten normalmente a un grupo selecto de usuarios, los cuales según convenga a su gente internamente. Se preparan las sesiones incluyendo material de exposición, guías, ejercicios, demostraciones y cuestionarios. Idealmente se crea una base de datos pequeña con información real.

Se imparte el entrenamiento y se realizan pequeñas pruebas para ver si todos están entendiendo. En caso negativo se pueden organizar más sesiones.

Finalmente se provee el mecanismo por el cual los demás usuarios recibirán la instrucción adecuada.

Preparar las pruebas de aceptación

Se revisan los planes de entrega y de aceptación de la etapa de diseño. Se ayuda en la planeación y preparación de las pruebas de aceptación, esto incluye la evaluación del criterio de aceptación de los usuarios, proveer cualquier información o equipo requeridos, dar la opinión sobre la manera de trabajar, el criterio para evaluar la conversión correcta de datos y procedimientos existentes, la manera en que se van a corregir errores en otros, etc.

Proveer soporte a las pruebas de aceptación

Esta actividad se encarga de que los usuarios prueben los aspectos operacionales del sistema. Hay que ayudar a los usuarios a que efectúen las pruebas correspondientes, a que detecten fallas o hagan preguntas en aspectos desconocidos a ellos, que se independicen y sean capaces de resolver problemas y preguntas. Poco a poco se van resolviendo todos los puntos del plan de pruebas para lograr su aceptación.

Conversión de información

Se prepara todo para convertir los datos que se tienen disponibles al formato del sistema nuevo, esto comprende recolectar datos requeridos para introducirlos a la computadora, crear programas de conversión de datos, proveer alfabetas para meter grandes volúmenes de datos.

Acto seguido se realiza la conversión de la información y se hacen pruebas, por ejemplo se generan reportes para determinar si los datos han sido almacenados correctamente.

Se compaginan los volúmenes de datos con respecto a las predicciones realizadas y en caso de haber grandes fluctuaciones se ajusta el diseño y la implementación de los archivos. Se busca finalmente obtener la aprobación de los usuarios.

Instalación de hardware y software

Esta actividad provee las facilidades y protocolos requeridos para el sistema. Se instalan máquinas, aplicaciones, muebles, cuartos y el sistema de comunicaciones necesarios para el funcionamiento adecuado del sistema nuevo. Se revisa la instalación y se prueban todos los componentes adicionales.

Hay que asegurar que se cuenta con los materiales necesarios para trabajar correctamente, esto es: tener papel, discos flexibles, cintas, cartuchos de tinta, etc. Se consulta al usuario sobre la terminación satisfactoria de las instalaciones.

Ensayos diversos

Se planean y ejecutan diversos ensayos que contribuyen a completar la etapa de transición. Se revisan los resultados de estos ensayos y según estos se realizan ajustes pendientes al sistema, para mejorar aún más la funcionalidad, la ejecución o la facilidad de uso. Todo esto debe ser llevado a cabo bajo el estricto control de la gente encargada de hacer pruebas al sistema.

Preparar el paso al sistema nuevo

Se revisan aspectos como disponibilidad de usuarios y sistemas, tiempos de producción y tiempos muertos, etc. Se organizan listas para resolver estas posibles eventualidades de la transición y se confirman fechas y horas disponibles.

Antes de comenzar el cambio del sistema hay que asegurar, que todos los pasos previos hayan quedado completamente cubiertos y que haya plena aceptación por parte de los usuarios.

Se revisan requisitos de seguridad, aspectos operacionales y ayuda de tipo técnica. Finalmente se cargan los programas y datos, asegurando que todo haya sido exitoso y finalmente se pide la aprobación para realizar el cambio al sistema nuevo.

Pasar al sistema nuevo

Esta actividad realiza el paso definitivo al sistema creado. Se realiza lo estipulado en el plan de transición elaborado en la etapa de diseño. Hay que controlar las actividades del plan de transición y finalmente dejarlo funcionando en forma definitiva. Se declara al nuevo sistema como el actual.

Soporte durante periodo crítico

Se provee de soporte a los usuarios para utilizar el sistema nuevo, se les ayuda a manejarlo, se les dan consejos para minimizar diversos irritantes.

Se identifican, registran y corrigen errores en el manejo del sistema. Hay que llevar un control estricto de los cambios del ambiente nuevo. Se contabilizan las caídas del sistema, el número de fallas y se registra el nivel de servicios.

Revisión posterior a la implementación

En esta actividad se lleva a cabo una sesión de retroalimentación, donde se mencionan diferentes puntos relacionados con todo el desarrollo del proyecto y se atienden recomendaciones provenientes de todos los grupos interesados.

Se colecciona y se agrupa información obtenida de encuestas hechas a los diferentes grupos interesados en el sistema, para producir material de retroalimentación, concretamente a los métodos y estándares empleados, a la intervención de los que desarrollaron el sistema, a la organización del proyecto en general y al éxito final.

Se genera un reporte con toda esa información y con las recomendaciones. Se atienden puntos necesarios que hayan surgido de esta actividad y se busca recibir la aprobación de fin de la etapa para entrar a la producción.

El sistema es liberado y con esto finaliza la intervención de programadores y desarrolladores.

2.3.3 Factores críticos para tener éxito

Es importante no perder de vista los siguientes puntos para lograr una etapa exitosa:

- Asegurar que el entrenamiento es apropiado y efectivo

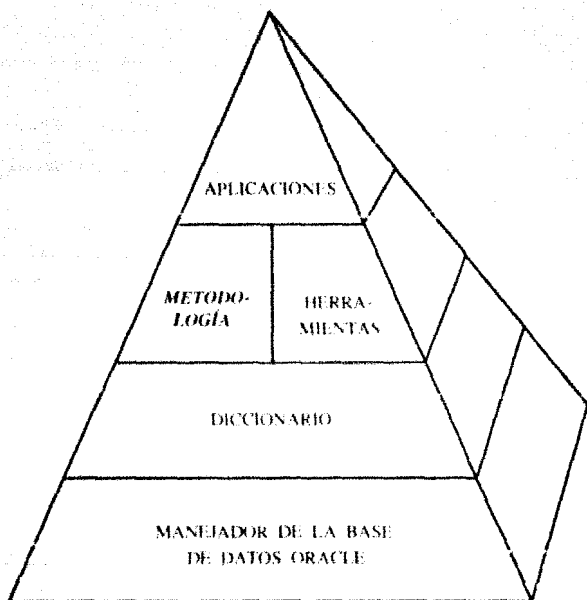
- Asegurar que los usuarios hagan suficientes pruebas para comprobar la operación correcta del sistema y para que se familiaricen adecuadamente.
- Coordinar la implementación que es compleja y crítica en cuanto al tiempo que tarda.
- Asegurar que los operadores y los usuarios técnicos sepan detectar anomalías del sistema y corregirlas.

2.3.4 Productos resultantes

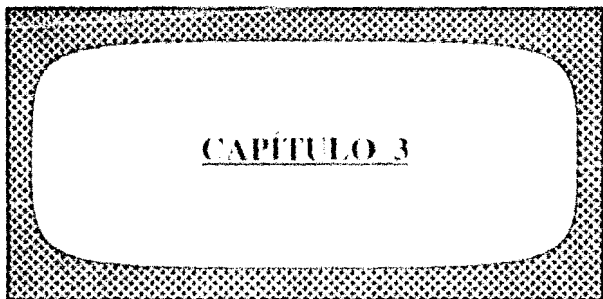
- Material de entrenamiento.
- Usuarios y operadores entrenados.
- Sistema funcionando al 100%.
- Datos convertidos.
- Reporte de revisión posterior a la implementación.
- Soporte a usuarios y operadores.
- Documentación completa del sistema.

2.9 CASE*Method dentro de la Tecnología CASE

El lugar lógico que ocupa el CASE *Method dentro de la tecnología CASE en los productos de la compañía ORACLE de México se muestra en la siguiente figura (ver próxima página). Es importante resaltar que la metodología va a la par con lo que son las herramientas CASE, pero se basa en el diccionario de datos (como ya se mencionó anteriormente) para ir guardando toda la información recopilada durante las diferentes etapas del desarrollo de un sistema. Como se puede apreciar en el diagrama, las aplicaciones se construyen sobre las herramientas y sobre la metodología, con el fin de facilitar lo más posible su desarrollo y elaboración.



<LUGAR QUE OCUPA LA METODOLOGÍA CASE >
<DENTRO DEL DESARROLLO DE APLICACIONES >



CAPÍTULO 3

HERRAMIENTAS CASE

A continuación describiremos algunas de las herramientas CASE utilizadas durante el desarrollo del trabajo realizado, que se presentará posteriormente (capítulo 4). Las herramientas CASE caen en tres categorías principales. La primera categoría se refiere a la metodología de desarrollo que constituye el conjunto de reglas y procedimientos a seguir por el equipo de desarrolladores para definir e implementar el sistema.

La segunda categoría la comprenden los productos del tipo diccionario, los cuales manejan toda la información acerca del sistema tal y como es implementado: los datos, las *funciones*, las reglas de la empresa, etc. También en esta categoría se encuentran las herramientas de diseño de diagramas (entidad-relación, flujo de datos, matrices de verificación, etc.).

La tercera categoría son las herramientas de trabajo para implementar los sistemas, es decir aquellas utilizadas para generar códigos, pantallas, reportes y otros elementos que forman una aplicación de software.

Un aspecto importante a aclarar es que con tanto que se menciona CASE como herramientas de software para realizar sistemas, es fácil encajar a cualquier producto de software, (como pueden ser los compiladores, procesadores de texto, lenguajes de programación y otros) como herramientas de CASE, ya que son utilizadas en el desarrollo de sistemas.

Pero cabe especificar que sólo consideramos como herramientas CASE al conjunto integral de herramientas que cubren todo el proceso de desarrollo de un sistema (ciclo de vida de los Sistemas).

Las herramientas que a continuación se mencionan son parte de un sistema CASE integral de la compañía ORACLE de México, que cubren todas las fases del ciclo de vida del software.

3.1 CASE*Method

CASE*Method es la metodología propia de ORACLE para el desarrollo de sistemas. Es el fundamento sobre el cual trabajan sus herramientas CASE. Para mayor referencia véase capítulo 2, donde se especifican todas las etapas que comprende.

3.2 CASE*Designer

Esta herramienta es una interfase gráfica de alto poder, que permite al usuario por medio de un sistema que utiliza un mouse, dibujar los diagramas de entidad relación, diagramas de flujo de datos, diagramas funcionales y matrices de análisis. CASE*Designer además de ser una eficaz herramienta para el diseño de diagramas, tiene como principal característica la de poder acceder directamente la información que se encuentra en el diccionario (CASE*Dictionary) y relacionarla con el diagrama que se está realizando. Cuenta también con la característica de editar textos que son automáticamente insertados en el diccionario.

3.3 CASE*Dictionary

Alrededor de este diccionario giran todas las demás actividades desarrolladas por el ingeniero de software, siendo éste el depósito de información. Este diccionario provee el lugar donde almacenar y analizar la información generada durante las primeras fases del desarrollo del sistema principalmente, la cual está a la disposición de todo el equipo de analistas, diseñadores, programadores, gerentes de sistemas, administradores del sistema, usuarios finales y para toda aquella persona que involucrada con el sistema, tenga necesidad de información propia del desarrollo de este.

La información almacenada en el diccionario es la que respecta a las diversas entidades, relaciones, atributos, sus descripciones, tamaños, reglas de validación, dominios, etc. Además de ser el lugar donde se almacena la información, CASE*Dictionary puede ser considerado como una herramienta inteligente, que nos permite hacer un análisis de la información que ahí se encuentra, por medio de diversos métodos que verifican la consistencia de la información.

Otra de las tantas cualidades que posee CASE*Dictionary es la de ser una herramienta que permite la elaboración de una buena parte de la documentación requerida en las diversas etapas del software, así también provee una potente interfase con las herramientas CASE, encargadas del diseño y construcción del sistema.

CASE*Dictionary trabaja en ambientes multiusuarios, soportando también ambientes distribuidos. Permite el diseño y la construcción de la base de datos, tomando en consideración toda la información que está almacenada. Para facilitar el funcionamiento y la coordinación de un proyecto, CASE*Dictionary soporta el manejo de versiones de una misma aplicación.

3.4 CASE*Generator

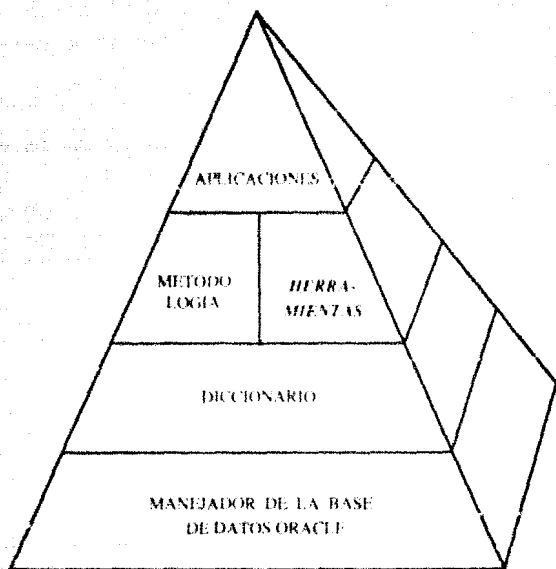
Los generadores son herramientas que son capaces de generar la aplicación en un primer corte, tomando en cuenta la información (especificaciones) recopilada durante las primeras fases del proceso de desarrollo y que se encuentra almacenada en el diccionario. Este modelo inicial creado por un generador la podríamos llamar un prototipo. De éste se partirá para después dar un pulimento y afinación a la aplicación. Al tener esta parte de la aplicación, la productividad se ve aumentada en una forma por demás considerable. Por ejemplo, el generador de formas crea formas que llevan ya programadas una gran cantidad de rutinas de validación, textos de ayuda, protección de la integridad de los datos, por mencionar las más importantes.

Todas estas tareas implican una gran cantidad de trabajo, por lo que al poder realizar esta programación de una forma automática, se disminuye el trabajo en forma considerable, además de que son utilizadas las especificaciones generadas en etapas anteriores del diseño. Es aquí donde podemos ver como se puede reutilizar la información que, en un proceso sin automatizar, es muy probable que se tenga que volver a recopilar parcialmente.

Existen una infinidad de herramientas CASE que pueden caer en estas categorías. Lo importante en un momento dado es la capacidad de integración con otras herramientas que permitan un ambiente de completa integración, llevado a través de todas las etapas del ciclo de vida del Software.

3.5 Las Herramientas dentro de la Tecnología CASE

El lugar lógico que ocupan las herramientas dentro de la tecnología CASE en los productos de la compañía ORACLE de México se muestra en la siguiente figura (ver próxima página). Nuevamente se puede apreciar, que las herramientas CASE van a la par con la metodología, aprovechando la información guardada en el diccionario de datos (como ya se mencionó anteriormente) para las diferentes etapas del desarrollo de un sistema.



< LUGAR QUE OCUPAN LAS HERRAMIENTAS CASE >
< DENTRO DEL DESARROLLO DE APLICACIONES >



CAPÍTULO 4

CASO PRACTICO

Elaboración de un Sistema de Computo para el Área Educacional de una Empresa

Introducción

En el capítulo siguiente se describe el proceso que se siguió para elaborar un sistema de cómputo basado en las etapas que propone el CASE*Method (ver capítulo 2). Para la elaboración de este sistema se utilizó lo mejor de la metodología descrita anteriormente como base fundamental de desarrollo.

Cómoda es saber que se tenga una metodología tan completa como CASE*Method, no significa que haya que seguir paso a paso las recomendaciones propuestas, ya que hay cosas que no van de acuerdo a una situación específica o a una forma de trabajar muy particular. Existen ciertas acciones o actividades que sirven para poder organizar mejor los datos y la información, sobre todo cuando se trata de sistemas sumamente grandes y/o de alta complejidad.

En general las metodologías se utilizan como base para desarrollar sistemas y difieren unas de otras en la forma de hacerlos y no tanto en el objetivo que se busca alcanzar. En unas se obtiene diferente información, en otras se elaboran diferentes diagramas o bien difieren unas de otras por el tipo de herramientas utilizadas como ayuda para llevar a cabo el desarrollo.

Además algunas metodologías están diseñadas para elaborar sistemas de grandes dimensiones y contemplan diversas dificultades que pueden presentarse en este tipo de sistemas, otras en cambio, suponen los mismos procedimientos sin importar el tamaño del sistema a desarrollar.

En nuestro caso particular, la metodología utilizada (CASE*Method) está contemplada para el desarrollo de sistemas grandes y hace uso de toda la estructura y herramientas que un sistema integral de CASE podrá llegar a utilizar.

Aunque esto no quiere decir que sólo se pueda utilizar esta metodología para sistemas muy grandes, usando parte de ella se pueden producir sistemas de tamaño medio con gran facilidad y eficacia. Se debe hacer notar, que para sistemas muy pequeños — aquellos que envuelven pocos párrafos e poca información — el uso de una metodología como esta puede resultar contraproducente.

A continuación se presenta el desarrollo de un sistema de computo que mantiene y procesa la información del área educacional de una empresa, cuyo fin es elaborar e impartir cursos de capacitación.

4.1 La Etapa de Estrategia del Sistema

Antes que nada, nos gustaría dar un panorama general de las actividades y funciones que se realizan en esta empresa, con el fin de ubicar al lector en el contexto de trabajo.

Esta empresa está dedicada a vender productos de software y una de las actividades que realiza es dar capacitación a aquellos clientes que han adquirido sus productos o piensan adquirirlos, además de capacitar al propio personal interno.

Dentro de las actividades que se realizan en el área educacional podemos mencionar las siguientes:

- Diseñar el contenido de los cursos
- Preparar y diseñar el contenido de los manuales que serán utilizados en los cursos
- Planear y organizar la impartición de los cursos
- Captar y evaluar a los instructores y asignarlos a la impartición de cursos
- Coordinar la asistencia de los alumnos a los cursos
- Controlar los ingresos obtenidos por concepto de impartición de cursos
- Mantener toda la información derivada de esta actividad, como puede ser: datos de clientes, asistentes, inventarios, costos, etc.

Para realizar esto, la empresa cuenta con personal calificado para la impartición de los cursos, llevando un control de los clientes que han tomado estos cursos y de la disposición de otros diversos, así como fechas en que se planean dar.

La problemática que se tenía en la empresa, es que no se poseía un control eficiente de los alumnos que habían tomado un curso, además se necesitaba llevar un control más preciso del calendario de cursos y de los ingresos generados. Otro de los problemas que se tenía, es que el manejo del cupo en los grupos se hacía de una manera desordenada. Adicionalmente nos encontramos con problemas organizacionales, ya que no se tenían bien definidos los servicios que se realizaban en la empresa y no había políticas claras y bien definidas.

En el momento de empezar el estudio, en esta empresa se contaba con un sistema que se había realizado sin previo estudio, lo que provocaba que hubiera mucha información redundante y no se mantenía todos los datos que eran generados en el departamento. Este sistema, ade-

más tenía que ser modificado cada vez que surgía un nuevo requerimiento, por lo que no funcionaba como se podría esperar.

El estudio del sistema anterior fue útil para nosotros, porque presenta un precedente de los elementos que se manejaban y de la terminología utilizada.

Posteriormente se hizo un análisis de las herramientas y los recursos con los que se contaban, para así poder elaborar un nuevo sistema. Se determinó que el equipo de cómputo que iba a ser utilizado estaba sobrando de capacidades, por lo que no se presta atención especial a este aspecto.

Las herramientas de desarrollo que se utilizaron son las propias que la empresa maneja: manejador de base de datos relacional, herramientas de desarrollo de cuarta generación, reporteadores, herramientas CASE para el desarrollo del sistema, así como un diccionario de datos.

Uno de los objetivos principales a seguir fue el determinar el comportamiento general del área educacional de la empresa y las actividades y funciones de cada uno de los integrantes. Para eso se establecieron entrevistas con las personas encargadas del área, involucrando también a personas de otras áreas y que trabajan en conjunto con el área educacional.

En estas entrevistas lo que se buscaba era determinar todas las funciones que se realizaban, para así poder localizar aquellas que era conveniente sistematizar y aquellas que no lo eran y lograr una comprensión del sistema sobre el cual se iba a trabajar.

El proceso de entrevistas consistió primeramente en obtener una lista de las actividades que se realizan en la empresa, para posteriormente clasificarlas jerárquicamente.

Como primer resultado se determinaron las principales funciones que se realizaban en esta área. Esto da en principio una base para que posteriormente se pueda analizar el funcionamiento entero del área en cuestión.

A continuación se muestran los grupos principales de funciones encontradas en una primera aproximación (pagina siguiente).

< FUNCIONES PRINCIPALES DEL AREA EDUCACIONAL >

FIGURA

Diseñar e impartir cursos e actividades con el fin de obtener el mejor aprovechamiento de los productos ORACLE.

RECURSOS

Controlar del tiempo así como obtener recursos físicos y humanos para la impartición de cursos.

PLANEACION

Planear y programar cursos e otras actividades referentes al área educacional, así como también a sentar políticas a seguir.

PROPAGANDA

Elaborar y entregar propaganda, publicidad e invitaciones, así como también otros cursos a impartir.

SEGUIMIENTO

Realizar el seguimiento y la verificación de recursos físicos y humanos necesarios para el funcionamiento óptimo del área educacional.

ADMINISTRACION

Administrar los recursos necesarios para los cursos y las actividades planeadas, como son las inscripciones de alumnos, requisiciones, facturación, etc.

DISEÑO

Diseñar y definir el contenido de los cursos, así como el material necesario para su impartición, así como también los procedimientos de los objetivos de entrenamiento.

IMPARTICION

Efectuar la impartición de los cursos y realizar las actividades planeadas.

Otro de los objetivos de las entrevistas era construir un modelo de datos lo suficientemente completo, que permitiera mantener y utilizar toda la información en una forma eficiente y consistente. Para esto se determinaron inicialmente las entidades de mayor importancia, lo que nos permitió tener un panorama general de los elementos que se iban a manejar y considerar la interrelación que existía entre éstas.

Para realizar esto, se determinaron las entidades que se manejaban en el sistema existente, obteniendo una descripción de cada una de éstas y considerando la forma en que se relacionan.

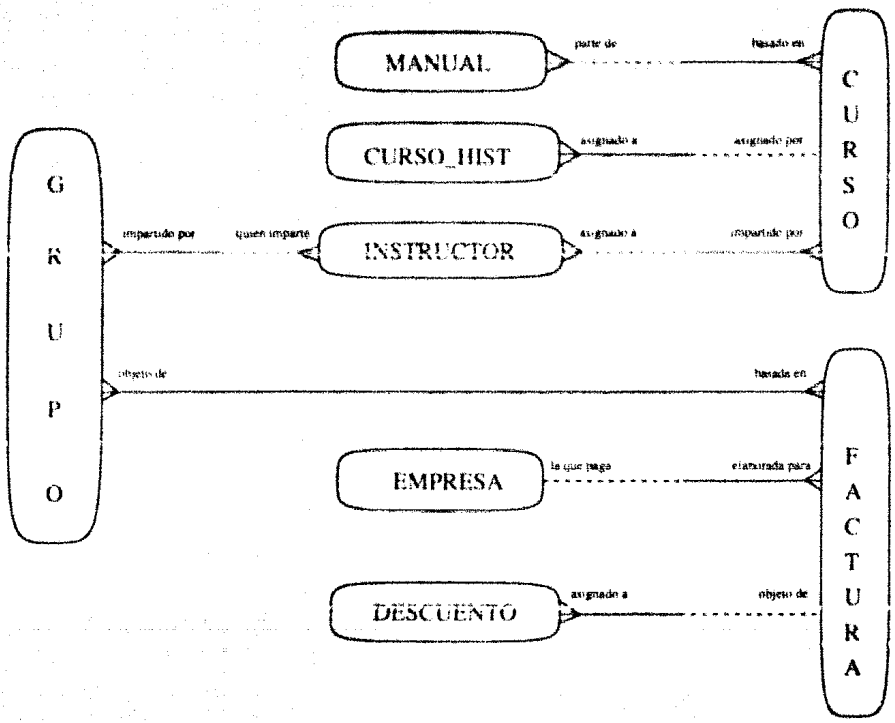
Esta información se obtuvo a lo largo de diversas entrevistas, donde se determinaban estas entidades, descripciones y relaciones de las mismas.

Después de obtener esta información, se construyó una primera aproximación del diagrama entidad-relación, donde se tomaron en cuenta sólo las principales entidades y relaciones.

El diagrama que a continuación se muestra, presenta estas entidades y sus relaciones encontradas en una primera aproximación (página siguiente):

Notese que el diagrama está compuesto de rectángulos con esquinas redondeadas, donde cada rectángulo representa una entidad. Las entidades están unidas por líneas que llevan nombres sobre las mismas, estas representan las relaciones entre entidades, que pueden estar dadas de uno a uno, de uno a muchos o bien de muchos a muchos, caso en el cual se deberá obtener una entidad adicional que las relacione.

< DIAGRAMA BASICO DE ENTIDAD - RELACION >



Toda la información que se recopiló fue a través de un proceso de retroalimentación con el usuario, donde se corregían posibles fallas o malas interpretaciones y donde se buscó obtener un modelo que representara lo más preciso posible las condiciones del mundo real.

Cabe hacer notar, que durante toda esta etapa de Estrategia, la información que se iba recopilando era introducida al CASE*Dictionary (ver herramientas CASE) capítulo V, en el cual se puede llevar un control muy eficiente de toda la información recopilada durante esta etapa. Todos estos datos serán reutilizados durante las siguientes etapas de desarrollo.

Durante esta etapa se obtuvo un panorama general del funcionamiento del área en estudio y de las condiciones en que este sistema trabaja y debería de trabajar en un futuro.

4.2 El Análisis del Sistema

Durante esta etapa se siguió el mismo proceso de entrevistas y juntas con los usuarios, con el fin de hacer un análisis detallado del sistema.

Durante este proceso nos dedicamos a obtener todas las funciones de una manera mucho más detallada de todo el mundo en estudio. Por ser tan detallado, este proceso requiere el más tiempo y dedicación que cualquier otro, ya que es esta etapa la que marca en forma tajante el éxito o fracaso de un proyecto.

Esto se puede apreciar en el capítulo 7, donde se indica el costo relativo de cualquier modificación realizada durante cada una de las etapas del desarrollo de un sistema. Si desde un principio queda bien analizado y diseñado todo el sistema, su buen funcionamiento queda asegurado para cuando el sistema se encuentre ya en producción.

La obtención del modelo jerárquico de funciones se obtiene también por el estudio detallado de la empresa, en coordinación con los usuarios de una manera interactiva y con un proceso constante de retroalimentación. Esto se hace con la finalidad de obtener un modelo lo más preciso y pegado posible al mundo real.

Al elaborar un estudio de las funciones que se realizan en una empresa, hay que considerar todas las funciones que existen en un momento dado. También hay que poner atención a aquellas funciones que no están contempladas, ya sea por desconocimiento del usuario o por que estas se encuentran en nuevos planes y políticas a futuro de la empresa, que solamente los directivos pueden conocer.

Para encontrar estas funciones hay que estar en estrecha coordinación con los jefes y gerentes, o con las personas que tengan poder e influencia en los cambios que puedan realizarse en un futuro.

Es interesante observar en el análisis funcional, que algunas de las funciones ya están sistematizadas otras en cambio son más casuales y factibles de ser sistematizadas. Por otro lado existen otras funciones, que por su naturaleza deben de ser manuales o en pocas ocasiones convertirse a manuales, en caso de que ya estén sistematizadas. Además se encontraron otras funciones que estaban fuera del alcance del sistema, por lo que fueron eliminadas.

Con esto se hace notar, que al principio hay que considerar todas las funciones que se presentan y luego se definirá la importancia y el alcance que tiene cada una dentro del sistema.

En el diagrama una función se muestra observando que las funciones de los niveles inferiores, regularmente arrojan una función a ser sistematizada, por lo tanto es aquí cuando podemos definir los procesos que se van a utilizar para llevar a cabo una función y los eventos, que harán que esta función sea ejecutada.

Para determinar cuando se debe de sistematizar o no una función dada, hay que hacer un estudio de cada una de ellas, considerando el tiempo y los problemas con los que nos podemos encontrar al hacer los procesos manualmente.

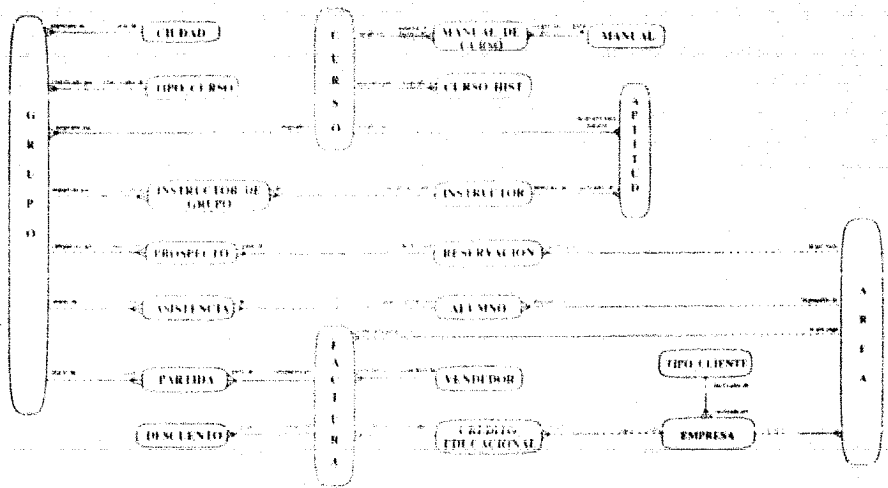
Un ejemplo de aquellas que no se sistematizaron, son todas las funciones en la que se establece una supervisión al nivel del personal, como por ejemplo, la revisión de los ejercicios de catequesis, la revisión de las cuentas que los usuarios utilizan en los cursos o aquellas funciones que implican una acción física de una persona, como una llamada telefónica o un envío de documentos.

Toda la información recopilada se almacena en el diccionario de datos, para poder posteriormente hacer revisiones o modificaciones de una manera fácil, de modo que a los analistas puedan ver reflejados estos cambios, además de poder utilizar esta información en cualquier etapa futura del desarrollo del sistema.

El análisis funcional que se obtiene a lo largo de este trabajo se muestra en el anexo A.

Durante esta etapa se realizó primero un análisis detallado de los cambios que se harán, en comparando algunas necesidades y condiciones actuales y futuras. Se detallaron y analizaron en forma detallada las relaciones entre estas, logrando obtener así un diagrama completo de esta relación que se muestra a continuación (pagina siguiente):

< DIAGRAMA DE ENTIDAD - RELACION >



Posteriormente se obtuvo una descripción detallada de cada entidad, con sus respectivos atributos descritos a detalle, así como una definición de dominios donde se encasillaban todos los atributos que corresponden a un mismo dominio.

Esta descripción detallada de las diversas entidades y sus respectivos atributos se puede apreciar en las siguientes páginas:

OBJECT
CASE Dictionary
Entity and Descriptions

Application Sys. Entity Name	Short name	Description	
EDUC_A2	W ALUMNO	ALU	persona que pasa a un curso
	APTUD	API	relación de los cursos que pueden ser impartidos por determinado profesor
	AREA	ARE	area o especialidad de trabajo de una empresa
	ASISTENCIA	ASI	relación de asistencia una clase a un grupo NOTA: se utiliza tambien en el proceso de marca y suma
	CIUDAD	CR	lugares geográficos
	CREDITO	CRU: CRU	unidad educativa que se da a los clientes en forma gratuita
	EDUCACIONAL		Se define unidad educativa como una medida de costo de los cursos por alumno. En el curso del curso de SQL:11 Se da 1 unidad educativa
	CURSO	CUR	curso impartido por el personal capacitado
	CURSO HIST	CUR HIST	curso impartido por personal capacitado historial de los cursos que se han dado de los cursos
	DESCUENTO	DIS	porcentaje que se rebaja el total de una factura
	EMPRESA	EMP	persona moral a la que se le ofrecen servicios
	PRESTERA	PRE	entidad financiera que presta dinero
	GRUPO	GRU	características particulares de un curso impartido a un grupo
	INSTRUCION	INS	persona capacitada para impartir algunos cursos determinados
	INSTRUCION DE GRUPO	INS GRU	relación de los cursos que han sido o serán impartidos por algún instructor
	MEDIDA	MEAS	medida de longitud, capacidad, peso, etc.

CASE Dictionary Report: EN11

ORACLE
CASE*Dictionary
Entity and Descriptions

Application Vn	Entity Name	Short name	Description
EDUSCA2	MANUAL DE CURSO	MAN_CUR	relacion de manuales que son utilizados en cada curso, ya que existen cursos que utilizan varios diferentes manuales
	PARTIDA	PART	relacion entre una factura y los grupos que interviene en ella
	PROSPECTO	PROSPE	persona a la que se le reserva un lugar a uno o grupos a impartir, sin necesidad de estar confirmada su asistencia
	RESERVACION	RES	relacion temporal de alumnos que han reservado, confirmados lugar a un curso o grupo
	TIPOCURSO	TIP_CUR	clasificacion de cursos con respecto a su contenido, como pueden ser: V, A, E, O, E, M, que tambien ORACLE vende por ORACLE, que no tienen ORACLE, o que tienen ORACLE, vendido por un O, E, M, o comprado en E, A
	TIPO CURSO	TIP_CUR	clasificador de cursos a impartir, en cuanto a su localizacion, esto es, indica si el curso tiene que impartirse en la empresa que solicita el curso, o en ORACLE mismo, en
	VENDEDOR	VEN	persona responsable de la venta de productos, la cual es autorizada para mostrar ciertos productos a sus clientes

CASE*Dictionary Reports: UN31

<ENTIDADES Y ATRIBUTOS>

Date: 05-JAN-91 Page: 1

ORACLE

CASE#Dictionary

Entity and Descriptions

Application: Vw Entity Name Short name Description

CASE#Dictionary Reports ENT1

PL REPORTS REPORTS

Al hacer el estudio de atributos y dominios, se determinaron los valores permitidos para cada atributo y las reglas de validación que afectarían a cada uno de éstos.

Al igual que toda la información recopilada con anterioridad, esta se fue depositando en el diccionario de datos.

Se hace continua mención al diccionario de datos, ya que este fue utilizado como base para el desarrollo de las diferentes etapas. Gracias a esta herramienta se lleva un control de toda la información recopilada y se puede analizar o cambiar con gran facilidad en cualquier momento, además de proveer una liga automática entre las diferentes etapas, proporcionando un camino suave y pautado entre éstas.

Otra de las ventajas que encontramos al utilizar estas herramientas de CASE fue el poder analizar la información en cuanto a las deficiencias que podamos tener, ya que cuenta con los propios procesos de verificación, que nos indican la falta de información básica.

Encontramos una gran diversidad de estos procesos, como pueden ser: la revisión de que todas las entidades tengan un identificador único, o que todas las entidades tengan relación al menos con otra, lo que nos permite determinar cuáles entidades y relaciones están dentro del alcance del sistema.

Otro proceso interesante a mencionar, es el uso de las matrices de análisis, con las que podemos hacer una referencia cruzada entre funciones y entidades, esto es, determinar en qué funciones se utiliza cada entidad y así poder definir cuándo alguna entidad no es involucrada en alguna de las funciones del sistema.

Adicionalmente las herramientas CASE, cuentan con una diversidad enorme de reportes útiles para agilizar el proceso del análisis, como son los reportes de identificadores únicos, reportes de funciones, de entidades y atributos, de descripción de campos, de relaciones entre entidades, de flujo de datos, etc.

Continuando con la información recopilada durante esta etapa, nos encontramos con el análisis de volúmenes de datos. Se determinó que la cantidad de información que maneja el sistema es de naturaleza estática, la información que tiene constante cambio crece en forma lenta y la capacidad de almacenamiento y procesamiento era superior a los requerimientos del sistema, por lo que no se prestó atención adicional a este análisis.

Una vez recopilada la información, esta fue revisada detalladamente en coordinación con los usuarios, para poder obtener la aprobación de los documentos que se presentaban en esta etapa y así poder proseguir con el diseño, que se describe a continuación.

4.3 El Diseño del Sistema

Una vez realizada la etapa de análisis y habiendo obtenido la aceptación del plan de desarrollo, fue posible comenzar el diseño del sistema, tomando como base primordial el modelo funcional propuesto, así como el diagrama final de entidad-relación.

Comenzamos por definir la arquitectura que tendría el sistema, que es la espina dorsal del mismo. Para lograr esta finalidad nos apoyamos en el esquema jerárquico de funciones y en el modelo entidad-relación y propusimos la arquitectura del sistema a desarrollar. Evaluamos todas las funciones que podían ser implementadas y manejadas desde el sistema, las reorganizamos y agrupamos de tal manera, que formarían grupos de funciones afines y que conformarían las diversas opciones de un menú principal. Este menú quedó formado por siete grupos de funciones, cuyas subdivisiones típicamente son las funciones implementadas en el sistema, o bien son a su vez nuevos menús.

Tuvimos mucho cuidado de no tener más de tres niveles en todos los casos, para facilitar al máximo la navegación por los menús y sus diferentes opciones.

Para hacer una selección adecuada de la arquitectura del sistema hay que tener siempre en mente, cuáles son las diez funciones más críticas, las diez funciones más urgentes, las diez más complejas y finalmente seleccionar veinte funciones adicionales importantes, más no críticas, para enfocar la funcionalidad de nuestro sistema hacia el campo de interés primordial y no perdernos en casos poco usuales e imprácticos.

La figura que a continuación se muestra (pagina siguiente) representa el menú principal y los diferentes menús que se diseñaron para tener acceso al sistema realizado en el equipo de cómputo. Toda esta parte fue realizada con un software para el control de formas llamado SQL*MENU.

ESTA TESTE NO DEBE
SALIR DE LA BIBLIOTECA

NIVEL 1

**** MENU PRINCIPAL ****

- 1) Inscripciones
- 2) Asistencias
- 3) Diseño de cursos
- 4) Planeacion e imparticion de grupos
- 5) Facturas y creditos educacionales
- 6) Registro e informes a clientes
- 7) Mantenimiento de catalogos e inventario
- 8) Salida

NIVEL 2

**** INSCRIPCIONES ****

- 1) Reservar lugar a alumnos
- 2) Confirmar una reservación
- 3) Cancelar lugar a alumnos
- 4) Cambiar alumnos a otro grupo
- 5) Menu anterior

NIVEL 2

**** ASISTENCIAS ****

- 1) Registrar lista final de asistentes
- 2) Registrar asistencia y derecho a diploma
- 3) Consultar vacantes en un grupo
- 4) Emitir lista provisional de alumnos inscritos a un grupo
- 5) Diplomas
- 6) Menu anterior

« OPCIONES DEL MENU »

NIVEL 2

**** DISEÑO DE CURSOS ****

- 1) Registrar y/o modificar características de un curso
- 2) Emitir lista de cursos existentes
- 3) Menú anterior

NIVEL 2

**** PLANEACION E IMPARTICION DE GRUPOS ****

- 1) Planeacion de grupos
- 2) Imparticion de grupos
- 3) Menú anterior

NIVEL 2

**** FACTURAS Y CRÉDITOS EDUCACIONALES ****

- 1) Registrar solicitud de factura
- 2) Cancelar solicitud de factura
- 3) Registrar y/o consultar facturas pagadas
- 4) Otorgar créditos educacionales a un cliente
- 5) Reportes de facturación
- 6) Menú anterior

< OPCIONES DEL MENU >

NIVEL 2

**** REGISTRO E INFORMES A CLIENTES ****

- 1) Registrar y/o consultar los datos del cliente
- 2) Emitir reporte del calendario de cursos
- 3) Elaborar cartas para envío de información a clientes
- 4) Menú anterior

NIVEL 2

**** MANTENIMIENTO DE CATALOGOS E INVENTARIO ****

- 1) Registrar inventario de manuales
- 2) Registrar instructores y/o sus niveles de experiencia
- 3) Modificar el valor de un crédito educacional
- 4) Emitir reporte del inventario de manuales
- 5) Menú anterior

NIVEL 3

**** REPORTE DE FACTURACION ****

- 1) Emitir reporte de solicitud de factura
- 2) Emitir reporte de ingresos teóricos
- 3) Emitir reporte de ingresos reales
- 4) Menú anterior

< OPCIONES DEL MENU >

NIVEL 3

**** DIPLOMAS ****

- 1) Emitir lista de alumnos con derecho a diploma de un grupo
- 2) Emitir lista de alumnos con derecho a diploma
- 3) Emitir carta de entrega de diplomas al cliente
- 4) Menú anterior

NIVEL 3

**** PLANEACION DE GRUPOS ****

- 1) Registrar instructores y/o su nivel de experiencia
- 2) Registrar calendario semestral de cursos
- 3) Consultar instructores que pueden impartir un curso
- 4) Registrar el inventario de manuales por curso
- 5) Consultar estadísticas de cursos impartidos
- 6) Menú anterior

NIVEL 3

**** IMPARTICIÓN DE GRUPOS ****

- 1) Registrar un grupo
- 2) Modificar un grupo
- 3) Posponer un grupo
- 4) Cancelar un grupo
- 5) Asignar instructores a un grupo
- 6) Emitir carta recordatoria de impartición de grupo
- 7) Menú anterior

< OPCIONES DEL MENÚ >

El siguiente punto importante a realizar fue el diseño de la base de datos. Para esto se utilizó una de las múltiples funciones soportada por el diccionario de datos, que consiste en generar a partir de las entidades registradas previamente, las tablas que se crearán para formar la base de datos propia del sistema, así como utilizar los atributos asociados a estas entidades para decidir selectivamente si será parte de una tabla en forma de un campo. Así es como el modelo de entidad relación pudo durante la etapa del análisis, fue utilizado para crear las tablas y sus respectivos atributos, que son una parte fundamental de cualquier sistema.

De nuevo se puede observar, como el diccionario de datos es en todos los aspectos muy útil, ya que para nosotros fue muy sencillo crear la base de datos, dado que todo ya había sido definido en las etapas previas. Otro punto importante que hay que hacer notar, es que existe una independencia total entre lo que es la etapa de estrategia y análisis, con respecto a las subsecuentes etapas de la metodología empleada, ya que nosotros decidimos utilizar las facilidades de una base de datos relacional, pero bien se hubiera podido llevar a cabo el diseño sobre archivos convencionales o por medio de una base de datos jerárquica.

La apéndice B muestra la definición de las tablas y sus respectivos campos empleados para la creación de la base de datos relacional. En esta se puede apreciar el nombre y la versión de la aplicación, así como el nombre, tamaño, tipo, la capacidad de aceptar valores nulos y una breve descripción de todos los atributos.

Para poder decidir el tamaño de cada atributo fue muy importante hacer una estimación de los volúmenes de datos que serían manejados, por lo menos de aquí a un par de años. Un ejemplo de esto sería ver lo que pasa con el campo de una clave que es fijo y por otro lado con un campo numérico, que puede llegar a crecer considerablemente con el tiempo. Si tomamos el campo de una clave de un instructor, se tiene la necesidad de definir el tamaño que tendrá siempre este campo, que en nuestro caso específico lo creamos de tres caracteres, pero si pensamos en el campo secuencial numérico que llevará el identificador único de grupo, debemos hacer un estimado de cuántos cursos se programan por mes, extrapolar este dato a un año y multiplicar por los n años de vida útil prevista para el sistema. No hay que olvidar dar una holgura por cualquier tipo de imprevisto que se pueda llegar a presentar.

Nuevamente todo esta integración de los distintos módulos para cada campo fue hecha con la ayuda del diccionario de datos. Al principio nos pareció muy tedioso el tener que registrar todos estos datos para cada una de las entidades por separado, pero a la hora de diseñar todo, nos dimos cuenta de que se simplifica cuantiosamente el trabajo, dado que todo ha sido analizado y definido previamente. Prácticamente toda la información registrada y almacenada con anterioridad puede ser reutilizada, gracias a la poderosa herramienta que representa el CASE*DICTIONARY.

Como lo indicamos anteriormente, todas las funciones que quedaron comprendidas dentro del sistema computarizado, son funciones que pudieron ser implementadas para ser hechas con ayuda del sistema. Sin embargo, existe otro grupo de funciones del área educacional, que no pudo ser integrado dentro del sistema, ya que son funciones manuales que tienen que ser realizada físicamente por una persona. Por supuesto que quedaron contempladas dentro del análisis funcional del área y ya depende del responsable de la misma, asignar una o varias personas que se ocupen de atender esas funciones complementarias manuales. Algunas de estas funciones pueden ser realizadas por los mismos instructores de cada grupo, pero en el caso de otras será recomendable tener a alguien adicional encargado del cumplimiento de dichas necesidades, para poder dar el mejor servicio posible.

Las funciones principales a las que nos referimos en esta sección son:

- Revisar el salón de clases antes de cada sesión (limpieza y orden)
- Revisar la cafetería (limpieza, existencia de bebidas y alimentos)
- Conseguir el material y los manuales del curso a impartir
- Crear el ambiente de cómputo propicio para impartir el curso
- Crear la lista de asistentes y tomar asistencia de alumnos.
- Hacer las cartas de entrega y las diplomas de curso
- Generar facturas

Una vez terminada la arquitectura del sistema y el diseño de la base de datos, proseguimos a crear las especificaciones de los módulos. Para cada módulo se tuvo que especificar el procesamiento a detalle, el uso de las entidades involucradas, el despliegue de las pantallas, los mensajes que aparecerían al término de cada acción, así como mensajes de ayuda o de situación inválida, y finalmente los procedimientos de validación de datos y el control para la ejecución óptima de cada módulo.

Para hacer este tipo de especificaciones hay que tener siempre en mente, que un buen sistema debe estar pensado bajo una única filosofía de trabajo, lo cual quiere decir que el usuario siempre utilizará las mismas teclas para hacer acciones parecidas, que los mensajes sean los mismos para situaciones equivalentes, que siempre haya una indicación en caso de error, etc. Hay que estandarizar las formas, las funciones de las diferentes teclas, los títulos, las acciones

automáticas del sistema, y en general cualquier aspecto que atañe al comportamiento o presentación del mismo.

Todas las especificaciones de las acciones a realizar por el sistema en cada forma y el control dentro de cada módulo, fueron registradas a detalle, para que en la fase de construcción sólo tuviésemos que ver las especificaciones de funcionamiento y pudiéramos crear cada módulo en un mínimo de tiempo.

Se creó una matriz cruzada entre módulos y atributos, que es la que indica cuáles módulos tienen acción sobre los diversos atributos de todas las entidades existentes. Hicimos una distinción para diferenciar si la acción sobre determinado atributo era de tipo lectura, escritura, creación, borrado o una combinación de éstas, mediante una nomenclatura que nosotros ideamos.

Esto fue de gran utilidad para revisar la creación de cada uno de los atributos involucrados en el sistema. Nos dimos cuenta que dos de ellos nunca eran creados, sin embargo sí poseíamos mecanismos con los cuales podíamos modificarlos. Esto fue obviamente un error de diseño, que gracias a la matriz entre módulos y atributos, pudo ser corregido a tiempo.

También se tuvo que incorporar un esquema de seguridad para ciertos módulos. Este tipo de acceso controlado es importante cuando queremos restringir a ciertos usuarios en cuanto a las posibilidades que tienen de ver cierta información confidencial, o si se quiere evitar que modifiquen información crítica del sistema.

Los módulos que fueron restringidos para usuarios sin privilegios amplios, son los relacionados con las facturas de los cursos. Consideramos que esta es una función que no cualquier usuario debe poder hacer, ya que errores en la elaboración de las facturas pueden provocar muchas disputas entre los clientes y la empresa. Sin embargo, es muy fácil cambiar el tipo de acceso a los diversos usuarios, a través de una utilidad de seguridad incluida en el manejador de menús, que solamente pueda acceder una persona con amplios privilegios.

Esta utilidad hace uso del concepto de clases de trabajo, que básicamente trata de agrupar diversos usuarios con el mismo tipo de privilegios bajo un conjunto único, que tiene permiso de hacer ciertas funciones o de acceder ciertos módulos dentro del sistema.

De esta manera es muy fácil asignar usuarios, que por ejemplo solo tengan acceso a los reportes generados, pero que no se les permita modificar nada, o bien, tener un grupo de usuarios que puedan acceder la mayoría de los módulos con excepción de los más críticos, y finalmente se puede tener un último grupo de usuarios que tenga permiso de hacer absolutamente todo lo que quiera con el sistema, como sería el caso del responsable del sistema.

SERVICIOS EDUCACIONALES
 « Cambio de alumnos de un grupo a otro »

Curso que se quiere cambiar:
 Num. de grupo Fecha de inicio Fecha de terminación Cancelado

.....
.....
.....
.....

Curso al que se quiere cambiar:
 Num. de grupo Fecha de inicio Fecha de terminación Cancelado

.....
.....
.....
.....

Los alumnos del grupo _____ se cambiarán al grupo _____ (S/N): _____

Introduzca el código de curso del cual se quieran cambiar los alumnos _____
 Char Mode: Replace Page 1 Count: *0

« CAMBIO DE ALUMNOS DE UN GRUPO A OTRO »

SERVICIOS EDUCACIONALES
< Emitir reporte de existencia de manuales >

- 1.- Desplegar reporte en la pantalla.
- 2.- Imprimir reporte.
- 3.- Generar reporte (archivo).
- 4.- Generar e imprimir reporte.
- 5.- Regresar al menu anterior.

Elige opción ----

Nombre del Archivo

de copias

1

Char Mode: Replace Page 1

00001 42

< REPORTE DE EXISTENCIA DE MANUALES >

SERVICIOS EDUCACIONALES
< Emitir reporte de alumnos con derecho a diploma por grupo >

- 1.- Desplegar reporte en la pantalla.
- 2.- Imprimir reporte.
- 3.- Generar reporte (archivo).
- 4.- Generar e imprimir reporte.
- 5.- Regresar al menu anterior.

Elija opcion ---

Num grupo

Nombre del Archivo

de copias

|

|

|

Char Mode: Replace Page 1

< REPORTE DE ALUMNOS CON DERECHO A DIPLOMA >

Los usuarios que pretendan utilizar el sistema deberán cumplir con los siguientes requisitos:

- Estar familiarizados con la teoría de bases de datos relacionales
- Saber utilizar SQL*PLUS
- Saber utilizar SQL*FORMS
- Saber utilizar SQL*MENU
- Saber utilizar SQL*REVIEW
- Saber utilizar CASE*DICTIONARY

El requisito más importante de los anteriores es el conocimiento de SQL*Forma, ya que a través de éste se hacen la mayoría de las pantallas de consulta y registro. De hecho no es indispensable el que los usuarios sean unos expertos en estos productos, pero el conocimiento de las teclas de función de los terminales facilitan el uso del sistema. Para aprender a usar estos productos, la empresa Citic le ofrece la impartición de estos paquetes durante todo el año.

Una vez concluido el diseño del sistema, se presentó todo lo elaborado durante esta etapa a los responsables del mismo. Este plan de aceptación tuvo como finalidad el mostrar la arquitectura del sistema, la estructura del menú con su esquema de seguridad, la funcionalidad de cada módulo, tests e las respuestas y acciones esperadas del sistema ante una entrada determinada y los niveles de ayuda al usuario, así como el contenido y la materia de acceder los diferentes reportes que se generarán.

Los responsables del sistema estuvieron conformes con lo propuesto, con excepción de pequeños detalles, mismos que fueron modificados en el acta. El total de los módulos a crear fue de 34, sin contar los diferentes reportes soportados por el sistema.

4.4.1 a Construcción del Sistema

La etapa de construcción es una etapa en donde hay que tomar todas las especificaciones señaladas durante el diseño y convertirlas a bases de datos reales en una computadora y crear todos los programas correspondientes a cada módulo.

Nuevamente utilizamos los recursos ofrecidos por el diccionario de datos, creado durante las primeras etapas del desarrollo, para convertir todas las especificaciones de las tablas por crear, a las tablas que utilizaríamos como base del sistema. Estas tablas quedaron dentro de una base de datos que había sido creada con fines de desarrollo. La referencia de las tablas creadas se puede ver en la sección anterior (diseño), ya que son exactamente las mismas.

Esta parte de la creación de la base de datos fue muy fácil, solo revisamos que todas las tablas quedarán bien creadas, esto es, que las especificaciones de los atributos estuviesen correctas en cuanto a tipo, longitud y su aceptación de valores nulos o no.

Posteriormente tuvimos que introducir información de prueba en cada una de las diferentes tablas, para poder comenzar a hacer los diversos programas y tener una manera de ir elaborando pruebas preliminares sobre cada uno de los módulos especificados. Los datos de prueba no correspondían con datos reales, pero sí tomamos información equivalente a lo que podrían ser los datos reales.

Terminada esta carga de datos, proseguimos a crear inicialmente la estructura del menú, mismo que nos ayudó a crear los diversos módulos de una manera organizada y a tener una visión más clara de los programas que ya habían sido creados y de cuales nos faltaba hacer. Los nombres de los programas que se realizarán se eligieron según la función que llevarían a cabo cada uno de estos, esto es, la opción número uno del submenú de facturas lo denominamos "Factur", la opción número cuatro del submenú de mantenimiento lleva el nombre de "Manten-I" y así subsecuentemente. Esta medida contribuye a poder llevar un control y orden durante la creación de los programas y facilita enormemente las necesidades de mantenimiento futuro al sistema.

El siguiente paso a dar fue crear los diversos programas según las especificaciones de la etapa de diseño. Los primeros módulos que fueron elaborados correspondieron a los programas más sencillos, y son aquellos que básicamente constituyen un catálogo de mantenimiento, que hace referencia a una o máximo dos tablas dentro de la base de datos y cuya lógica no es compleja.

Proseguimos con los programas un poco más complejos y dejamos al final aquellos, cuya lógica es muy compleja porque accesan muchas tablas a la vez y porque hay que garantizar la integridad de los datos para cada acción diferente que se le pueda ocurrir a una usuario. Aquí una de las formas más complejas fue la del registro de facturas, ya que involucra muchas tablas simultáneamente y porque la información es crítica (se trata de hacer un desglose de todo lo que tiene que pagar un cliente).

Cada una de las diferentes formas creadas fue revisada varias veces por el programador,

para ver si cumplía con los requisitos de las especificaciones de funcionamiento. Se revisó que cada tecla tuviese una función bien definida y que teclas diversas fueran deshabilitadas para asegurar el buen funcionamiento del módulo. Se revisaron los mensajes de ayuda y error que manda el sistema, para ver si son inteligibles y si de verdad indican lo que está pasando, para que sea una ayuda genuina al usuario y no como pasa a menudo con los sistemas de cómputo, que representan una fuente de confusión, ya que los mensajes que se obtienen no corresponden con la situación actual o bien son vagos o ambiguos.

También se revisó minuciosamente la actualización correcta de las tablas involucradas en cada programa, a fin de que la información fuese modificada adecuadamente y que mantuviese la integridad de la base de datos. En este respecto pasamos mucho tiempo, ya que una base de datos no sirve de nada, si la información obtenida no es la correcta, aun teniendo un diseño óptimo y formas muy dinámicas que accesen estos datos.

Conforme íbamos avanzando en la creación de los programas, fuimos elaborando a la par la documentación del usuario, para que quedaran asentadas todas las características, funciones y la utilización de cada módulo en específico.

Los programas se optimizaron lo más posible a través del uso de pequeños trucos existentes para acceder más rápidamente las tablas de la base de datos. Un ejemplo de las medidas tomadas para optimizar la ejecución de los programas fue el indicar en cada campo de un *query* (selección de datos) el nombre de la tabla, cosa que por comodidad no se hace en la práctica, pero que produce un incremento de velocidad para ejecutar el *query*.

Otra buena costumbre en la programación con SQL *Forms, es realizar la validación de campos a través del mismo SQL *Forms y no realizarla por rutinas especiales, a menos que la situación la precise.

Una vez terminados todos los programas, nos dedicamos a crear los diversos reportes. Pusimos mucho énfasis en crear las mismas interfaces con el usuario, independientemente de que cada reporte muestre información diferente. Para esto creamos formas con diversas opciones a escoger, para que el usuario tuviese la oportunidad de ver la información del reporte en pantalla, para mandarla a imprimir o bien para hacer ambas cosas a la vez.

Consideramos que un sistema debe de dar la opción de ver previamente la información generada en un reporte, porque constituye también una manera adicional de ver la información contenida en la base de datos y tal vez ese usuario ni siquiera le interesa obtener todo un reporte cuantioso de hojas, sino solo quiera consultar el total de alguna factura en específico. Esta es una medida que ahorra mucho tiempo e invita al usuario a no gastar tanto papel, que finalmente es tirado a la basura.

Otra de las características de los reportes, es la posibilidad de generarlos en un lapso de tiempo definido por el usuario a través de indicar una fecha inicial y una final, para que así no siempre salgan reportes kilométricos e incómodos en su manejo.

Cuando fueron finalizados todos los programas y los diversos reportes, proseguimos con las pruebas definitivas del sistema. Para lograr el mejor resultado es necesario probar cada uno de los módulos de manera independiente y por supuesto también en conjunto con los otros módulos, ya que lo que hacen ciertos programas sobre la base de datos, puede tener repercusión sobre el funcionamiento de otros programas.

Para revisar que toda la información fuera creada, modificada o borrada correctamente, tuvimos que crear formas temporales que nos permitiesen ver los datos de cada tabla por separado de manera rápida y revisar para cada módulo en la matriz cruzada de atributos función, para determinar si la información era actualizada como se había previsto y especificado durante el análisis.

También revisamos que la información generada por los diferentes reportes fuera correcta. Dado que cada programa y reporte ya había sido revisado y depurado previamente, fue una tarea sencilla el realizar todas las pruebas del sistema completo.

Se hicieron pruebas adicionales para verificar la ejecución rápida del sistema. Se cargó mucha información de prueba para simular una base de datos con una cantidad de registros aproximada a la que en realidad se tendrá. Esta es la única manera de saber si el sistema trabajará con un tiempo de respuesta aceptable o no.

En todos los casos de prueba se tuvo una respuesta del sistema menor a cuatro segundos, tiempo que se considera todavía por debajo del límite aceptable de espera. Esto nos permitió liberar el sistema y considerarlo como un sistema terminado.

4.5 La Documentación del Sistema

La documentación del sistema se puede dividir en forma general en dos partes: la documentación proporcionada automáticamente por medio del diccionario de datos y la generada por nosotros para el uso y mantenimiento del sistema.

En cuanto al primer tipo de documentación se tiene una serie muy completa de reportes, que pueden ser obtenidos con gran facilidad por medio del diccionario de datos. Este diccionario fue utilizado en las primeras etapas del desarrollo del sistema para registrar todo tipo de información útil. Esta información incluye las funciones del sistema, las entidades primarias y

secundarias, atributos con sus respectivas descripciones, tipos y extensiones, los diferentes dominios de los datos, frecuencias y volúmenes, etc. Esta es una fuente valiosa de información para documentar las diferentes partes de un sistema.

Para obtener cierta información por medio del diccionario de datos, lo único que hay que hacer es invocar el diccionario desde una terminal conectada al sistema desde la cual se tenga acceso al mismo y pasar a la sección de reportes de cada etapa en particular, o bien a la sección general de reportes e introducir el número correspondiente de la opción deseada. Acto seguido el diccionario de datos preguntará si la información se quiere ver en la pantalla, si se desea producir un reporte o análisis.

A través del diccionario de datos se tendrá acceso a los reportes más variados, como pueden ser reportes de descripción de entidades, de sus relaciones con respecto a otras entidades, a las diversas llaves primarias y/o secundarias, a los atributos de las entidades, etc.

También se pueden generar reportes relacionados con los volúmenes de la información manejada, las frecuencias y los tipos de cada evento, así como matrices de función/entidad, módulos/funciones y muchos más.

La ventaja de mantener la documentación creada con el diccionario de datos es la capacidad de poder generar diversos tipos de reportes con los mismos datos base, esto es, con la información recopilada y guardada con esta herramienta durante diversas etapas del desarrollo de un sistema.

Es muy fácil el actualizar cualquier tipo de información registrada ahí. Es además muy confiable, ya que la información se encuentra centralizada y no duplicada, siendo muy fácil mantenerla actualizada para consultas posteriores.

Toda la información generada de esta manera es sobre todo útil para quienes quieran ver la estructura interna del sistema (probablemente algún programador o analista que requiera ciertas referencias para hacer modificaciones al sistema, para el mantenimiento o para crear nuevos módulos y reportes necesarios para el crecimiento del mismo). El usuario final no encontrará el manual de operación a través de esta herramienta. Ejemplos de este tipo de documentación pueden ser vistos en la descripción que hacemos del desarrollo del sistema en las etapas de análisis, diseño y construcción.

Existe otro tipo de documentación creada por nosotros que nos proporciona la descripción del funcionamiento y el manejo de las formas que físicamente conforman todo el sistema. Se trata de un documento que nos indica el objetivo de cada una de las diferentes pantallas del sistema y las indicaciones necesarias para obtener y crear la información deseada. Los usua-

rios del sistema pueden consultar este manual, en caso de tener duda del funcionamiento de alguna de las formas creadas

Es importante resaltar, que el sistema fue creado con características que proveen un ambiente de trabajo amigable. Esto implica el que haya siempre una ayuda e indicadores de la acción a tomar en la mayoría de los campos que conforman los diversos módulos del sistema, por ejemplo, si en un campo se tiene la opción de introducir un nombre completo, parcial o bien sólo un pseudónimo o alias, se indicará de manera automática la información que se espera recibir en un renglón dedicado a guiando al usuario por toda la forma.

Esta característica de poder indicar sólo una parte de un nombre también pertenece al ambiente amigable creado, ya que permite hacer búsquedas de la información sin tener que teclear el nombre exacto y completo, sino sólo una parte mínima, que diferencie a esa cadena de caracteres como única dentro de la base de datos.

Una característica adicional de este ambiente amigable es la estandarización del despliegado de las formas y de la función de diversas teclas de control utilizadas en el manejo de SQL Forms. Se trata de dar la mayor funcionalidad a la tecla de *Retorno* y cuando es indispensable el uso de una tecla como *Próxima Pantalla*, se indica en el renglón de ayuda al usuario.

Para lograr un mejor funcionamiento global y asegurar la consistencia de los datos, se deshabilitaron varias teclas, cosa que no queda documentada en el manual del usuario, pero que el ambiente amigable indica en caso de que el usuario oprima una tecla con función restringida en ciertos campos o bloques.

También se cuenta con una serie muy completa de indicadores de validación por error o no factible, como por ejemplo podría ser, que erróneamente se haya introducido una fecha final menor que una fecha inicial. Esta situación será detectada automáticamente y el usuario no podrá seguir, hasta introducir una información congruente. En el caso de que sólo se puedan introducir datos fijos como una respuesta de "Sí/No" o claves existentes para brindar la búsqueda de cierta información, se tienen mecanismos que proveen al usuario con la clave necesaria o con el tipo de respuesta que se espera obtener.

Este tipo de revisión de consistencia no siempre es posible darla, ya que en el caso de caracteres alfabéticos (como en un nombre propio o el de una empresa o personal), no es posible revisar la ortografía y/o el contenido de los datos registrados.

Siempre es factible, que aun existiendo esta ayuda en línea para el funcionamiento apropiado de cada forma bajo un ambiente de trabajo amigable, haya situaciones especiales en las cuales un usuario requiera lineamientos adicionales para el manejo óptimo de alguna pantalla.

Consecuentemente se creó el manual del usuario, el cual describe cada módulo del sistema, indicando desde dónde se puede acceder cada forma a partir del menú principal. Contiene anotaciones adicionales de qué programa es ejecutado desde cada opción, con el fin de facilitar un poco el mantenimiento del sistema.

En caso de que el usuario encuentre un funcionamiento anormal desde alguna de las opciones del menú, podrá recurrir al manual y determinar cuál de los módulos no está trabajando correctamente, para que pueda ser corregido lo antes posible por la gente encargada del mantenimiento del sistema.

El sistema desarrollado se diseñó de tal manera, que no es necesaria la intervención de un operador dedicado a mantener en operación el sistema, de ahí que no se haya elaborado un manual de operador. La única acción que pudiese ser considerada como perteneciente a las labores de un operador es el respaldo de la información contenida en la base de datos. Como para efectuar este procedimiento basta hacer un *export* de la información y un respaldo a cinta desde el sistema operativo, no fue necesario incluir la explicación de procedimientos especiales para llevar a cabo esta función.

4.6 La Transición del Sistema

Es importante aclarar que es el momento de realizar esta tesis el sistema se encontraba en la etapa de transición, por lo que se describirán las acciones tomadas hasta el punto en el que actualmente se encuentra. Consideramos que el estado en que se encuentra el sistema es suficiente para podermos dar cuenta de las ventajas que tuvimos al utilizar la tecnología CASE.

Como mencionamos con anterioridad, ya se contaba con un sistema que cumplía con sólo algunas de las funciones necesarias para el correcto funcionamiento de esta área de trabajo. Nuestra primera misión fue la de informar a los usuarios que el nuevo sistema iba a empezar a ser utilizado.

Una vez informados los usuarios, comenzamos con una demostración del sistema nuevo, en donde les explicamos el funcionamiento general y las ventajas que tendría su utilización. Se elaboraron diversas prácticas de prueba con el usuario en el transcurso de la introducción al

sistema, éste tuvo oportunidad de involucrarse de tal manera que finalmente pudo formular las dudas que le surgieron acerca del sistema.

El plan de trabajo consistió en migrar los datos del sistema anterior al nuevo. Para este proceso se utilizó la información que se había recopilado y se elaboró un estudio de los datos que se encontraban en el sistema anterior. Dentro de este estudio de datos se determinó cuáles eran de utilidad para el nuevo sistema y cuáles se necesitaban recolectar para tener una base de información que permitiera comenzar a utilizar el sistema; esto es, la información de las tablas base.

La primera actividad en que se involucró a los usuarios fue la captura de la información básica, como son catálogos, entidades menores y toda aquella información que se conociera en el momento y que fuera de naturaleza estática (códigos, claves, etc.), todo esto con el fin de involucrar al usuario en los procesos más sencillos del sistema.

Una vez instalado el sistema en el equipo en el que se utilizara, migrados los datos y capturada la información faltante, comenzó la utilización del nuevo sistema en paralelo con el sistema anterior, todo esto con el fin de detectar posibles fallas y no afectar en la producción del sistema anterior.

El sistema se encuentra actualmente en esta etapa de desarrollo; el usuario está haciendo pruebas y nosotros estamos al tanto para corregir las posibles deficiencias o pequeños nuevos requerimientos que puedan surgir.

Todas estas actividades se están llevando a cabo en una forma interactiva con el usuario, ya que continuamente se realizan pruebas dando se analizan las fallas encontradas.

Se cuenta además con el apoyo de un manual de usuario, en donde se explica el funcionamiento del sistema en forma detallada. Esta etapa terminará cuando consideremos que están resueltas todas las fallas que puedan surgir del uso del sistema.

4.7 La Etapa de Producción del Sistema

En cuanto a esta etapa se tiene contemplado que el sistema seguirá corriendo en paralelo hasta el momento en que se corrijan las deficiencias que se puedan presentar en la etapa anterior, una vez comprobada la funcionalidad del sistema, se liberará y se pondrá en manos de los usuarios completamente.

Continuamente estaremos al tanto del funcionamiento del sistema y se planea estar como apoyo al mismo, hasta que el usuario lo domine completamente y se realicen las correcciones, cambios o ajustes que se presenten durante la producción del sistema.

Finalmente se le entregará al usuario toda la documentación requerida para el manejo del sistema, además de toda la recopilada durante las diversas etapas, la cual se encuentra en su totalidad en el CASE*Dictionary, todo esto con el fin de que se puedan hacer modificaciones o extensiones futuras sin la necesidad de contar con nuestra ayuda.



CONCLUSIONES

CONCLUSIONES

Esta tesis es el resultado de una investigación realizada sobre historia, metodología y herramientas de CASE, así como de la realización de un sistema creado bajo esta tecnología.

Se obtuvo un sistema eficiente que funciona con un ambiente amigable al usuario y sumamente apegado al funcionamiento del área educacional de una empresa de cómputo, que es fácil de utilizar y de mantener. El sistema tiene como objetivo el simplificar y automatizar todas las funciones particulares del área anteriormente mencionada. El sistema está en la etapa de transición, ya que se está pasando a manos de los usuarios para que ellos comiencen a utilizarlo y se puedan efectuar las últimas afinaciones sobre los diversos procesos, razón por la cual todavía no se posee información sobre las posibles deficiencias del sistema.

Los beneficios del sistema creado son cuantiosos, ya que se espera que la implantación de esta nueva herramienta de trabajo en el área educacional aplique todas las funciones que se realizan actualmente y haga que el área se pueda organizar adecuadamente. También provee la capacidad de reutilizar toda la información recopilada para posibles cambios en un futuro, los cuales podrán ser efectuados de manera rápida.

La elaboración de esta tesis incrementó considerablemente nuestros conocimientos sobre la tecnología CASE, los antecedentes históricos, las herramientas existentes y el conocimiento de una metodología más completa.

Las herramientas CASE son muy útiles en la elaboración de un sistema, ya que automatizan muchas de las funciones del desarrollo de sistemas, a la vez que permiten reutilizar el trabajo invertido durante etapas iniciales del desarrollo en las etapas posteriores y finales. Por otro lado es muy conveniente seguir los lineamientos de una metodología completa y organizada, para lograr un desarrollo por etapas y diversas actividades de manera ordenada. Es muy fácil olvidar pequeños detalles, sobre todo si se trata de crear un sistema grande en el cual haya mucha gente involucrada, que a la larga llegan a ser significativos y arreglarlos en etapas finales del desarrollo puede resultar muy costoso.

Nos gustaría agregar que no hay que ver a la metodología CASE como una panacea. La metodología CASE está en armonía a cubrir todas las necesidades para el desarrollo de un sistema grande, por lo que realizar todas las propuestas propuestas en el desarrollo de un sistema chico sería impráctico. Para un sistema mediano como el realizado por nosotros es recomendable utilizar la metodología parcialmente.

Es evidente que seguir una metodología completa es conveniente, pero más bien hay que tomarla como una guía de pasos a seguir, en donde interviene el criterio y la experiencia de cada líder de proyecto al adecuar sus necesidades de desarrollo a la metodología propuesta y no seguir indiscriminadamente todas las actividades propuestas, ya que esto podría generar mucho trabajo adicional e inútil.

Finalmente llegamos a la conclusión, de que en la actualidad es necesario hacer uso de herramientas CASE y de una metodología completa para incrementar la producción de sistemas de cómputo.

Creemos que debe de haber un cambio en la mentalidad de los programadores, analistas, diseñadores y líderes de proyectos, ya que la tecnología evoluciona, los sistemas son cada vez más complejos y existe una necesidad imperiosa por ser cada vez más productivos. Si los creadores de nuevos sistemas no cambian sus costumbres de diseño y programación, cada vez será más grande la diferencia que exista entre el desarrollo de hardware y de software. Pensamos que los creadores de software deben de utilizar todas las ventajas que ofrece la tecnología CASE.

Sabemos que es difícil lograr esta meta, ya que representa inicialmente una inversión para adquirir las herramientas y el conocimiento de la tecnología (que seguramente será recuperada por un incremento en su producción), pero tal vez lo más difícil de lograr es el cambio de mentalidad que se requiere por parte de los desarrolladores.



APENDICE A

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY REPORT

Report Date

24-NOV-90 16:58:49

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : EDUCA2

EDUCA2

Diseñar e impartir cursos a usuarios con el fin de obtener el mejor aprovechamiento de los productos ORACLE.

REC

Obtener defensas y/o obtener recursos físicos y humanos para la impartición de cursos.

PLAN

Planear y programar cursos y otras actividades referentes al área educacional, así como también asentar políticas a seguir.

PROPAG

Emisar y entregar propaganda, publicidad e información a los clientes sobre cursos a impartir.

SEGU

Realizar el seguimiento y la verificación de recursos físicos y humanos necesarios para el funcionamiento óptimo del área educacional.

ADMIN

Administrar los recursos necesarios para los cursos y las actividades planeadas, como son las inscripciones de alumnos, requisiciones, facturación, etc.

CONTINUED

FUNCTION HIERARCHY
(CONTINUATION)

APPLICATION : EDUCAJ

VERSION : 9

REFERENCE : EDUCAJ

EDUCA2

DISEÑO

Diseñar y definir el contenido de los cursos, así como el material necesario para su impartición y actividades que conlleven al cumplimiento de los objetivos de entrenamiento.

IMPARTI

Efectuar la impartición de los cursos y realizar las actividades planeadas.

END

Date : 24-NOV-90

ORACLE CASE Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : REC

REC

Corregir deficiencias y/o obtener recursos físicos y humanos para la impartición de cursos.

REC1

Obtener el material necesario para la impartición de cursos (carpetas, manuales, copias, proyectores, acetatos, etc.) y registrar éstos en el inventario.

REC2

Obtener recursos de cómputo (soporte técnico, generación de claves, etc.).

REC3

Pedir servicios de cafetería.

REC4

Hacer la requisición de personal nuevo.

REC5

Registrar instructores y/o sus niveles de experiencia.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : REC5

REC5

Registrar instructores y/o sus niveles de experiencia.

REC5.1

Registrar instructores.

REC5.2

Registrar el nivel de experiencia de un instructor.

END

FUNCTION HIERARCHY

APPLICATION : EDUCAZ

VERSION : 9

FUNCTION : PLAN

PLAN

Planear y programar cursos y otras actividades referentes al Area educacional, así como también asentar políticas a seguir.

PLAN1

Determinar el equipo de instructores disponibles y su nivel de experiencia para impartir los diferentes cursos.

PLAN3

Determinar los recursos materiales existentes necesarios para la impartición de cursos (como manuales, copias, claves de computadora).

PLAN4

Estimar la demanda de cursos a impartir y planear y programar el calendario semestral de cursos ordinarios (asentar características de grupo: fechas, duración, etc.).

PLAN6

Planear y programar cursos especiales (no contemplados en el calendario semestral); (asentar características de grupos: fechas, duración, etc.).

PLAN5

Discutir, planear y dar a conocer políticas a seguir en el Area educacional.

CONTINUED

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY
(CONTINUATION)

APPLICATION : EDUCAS

VERSION : 9

REFERENCE : PLAN

PLAN

PLAN8

Registrar un cambio de planes en la programación de un grupo (por cambio a fechas, duración, prácticas, asignar o cambiar instructores, etc.), así como la cancelación de un grupo a impartir.

PLAN5

Registrar el calendario semestral de cursos ordinarios.

PLAN7

Registrar cursos especiales (no registrados en el calendario semestral).

PLAN2

Registrar instructores y/o sus niveles de experiencia.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAO

VERSION : 9

FUNCTION : PLAN5

PLAN5

Registrar el calendario semestral de cursos ordinarios.

PLAN5.1

Registrar los diferentes grupos a impartir.

PLAN5.2

Registrar el lugar de impartición de los grupos.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : PLAN7

PLAN7

| Registrar cursos especiales (no registrados en el |
calendario semestral).

PLAN7.1

Registrar el grupo del curso a impartir.

PLAN7.2

Registrar el lugar de impartición del curso.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDGCAP

VERSION : 9

FUNCTION : PLAN2

PLAN2

Registrar instructores y/o sus niveles de experiencia.

PLAN2.1

Registrar instructores.

PLAN2.2

Registrar el nivel de experiencia de un instructor.

END

Date : 24-NOV-90

ORACLE : CASE*Dic1989y

FUNCTION HIERARCHY

APPLICATION : EDUCAZ

VERSION : 9

FUNCTION : PROPAG

PROPAG

Emitir y entregar propaganda, publicidad e
información a los clientes sobre cursos a impartir.

PROPAG01

Emitir y entregar el calendario de cursos a los
clientes y vendedores.

PROPAG02

Preparar, emitir y/o entregar material publicitario
de los cursos a impartir.

PROPAG03

Proporcionar informes a clientes.

END

Date : 24-NOV-90

ORACLE : CASE-Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAT

VERSION : 9

FUNCTION : PROPAG1

PROPAG1

Emitir y entregar el calendario de cursos a los
clientes y vendedores

PROPAG1.1

Determinar el cliente(s) al cual se le enviara el
calendario de cursos.

PROPAG1.2

Determinar el calendario de cursos que sera
proporcionado al cliente(s).

PROPAG1.3

Enviar al cliente(s) el calendario de cursos.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : PROPAG1.1

PROPAG1.1

Determinar el cliente(s) al cual se le enviara el calendario de cursos.

PROPAG1.1A

Obtener la información de un grupo de clientes (como por ejemplo si se trata de un V.A.A., de un D.E.M., de alguien que no tiene ORACLE, etc.) y elaborar la carta anexa.

PROPAG1.1B

Obtener la información de un cliente en específico y elaborar la carta anexa.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HYPARCHY

APPLICATION : EDUCAZ

VERSION : 9

FUNCTION : PROPAG2

PROPAG2

Preparar, emitir y/o entregar material publicitario de los cursos a impartir.

PROPAG2.1

Determinar el cliente(s) al cual se le enviará el material publicitario u otros.

PROPAG2.2

Determinar el material publicitario que será proporcionado al cliente(s).

PROPAG2.3

Enviar al cliente(s) el material preparado.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAP

VERSION : 9

FUNCTION : PROPAG2.1

PROPAG2.1

Determinar el cliente(s) al cual se le enviará el material publicitario u otros.

PROPAG2.1A

Obtener la información de un grupo de clientes (como por ejemplo si se trata de un V.A.R., de un O.E.M., de alguien que no tiene ORACLE, etc.) y elaborar carta anexa.

PROPAG2.1B

Obtener la información de un cliente en específico y elaborar carta anexa.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : PROPAG3

PROPAG3

Proporcionar informes a clientes.

PROPAG3.1

Informar verbalmente al cliente lo referente a los
cursos a impartir (como fecha, proceso, etc.), así
como los requisitos necesarios para asistir a uno de
ellos.

PROPAG3.2

Registrar los datos del cliente.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : PROPAG3.2

PROPAG3.2

Registrar los datos del cliente.

PROPAG3.2A

Si existen datos del cliente, se pueden dejar como están o bien modificarlos.

PROPAG3.2B

Si no existen datos del cliente se registran sus datos.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAS

VERSION : 3

FUNCTION : SEGUI

SEGUI

Realizar el seguimiento y la verificación de recursos físicos y humanos necesarios para el funcionamiento óptimo del Área educacional.

SEGUI1

Evaluar la calidad del curso (es. LECTA e instructores, contenido del curso, servicios y facilidades).

SEGUI2

Verificar la existencia de recursos (manuales, carpetas, acetatos, etc.).

SEGUI3

Verificar el buen funcionamiento de los servicios de cómputo.

SEGUI4

Verificar los servicios de intendencia y cafetería.

SEGUI5

Llevar el seguimiento de créditos educacionales utilizados por el cliente.

CONTINUED

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY
(CONTINUATION)

APPLICATION : EDUCA7

VERSION : 9

REFERENCE : SEQUI

SEQUI

SEQUI6

Llevar el seguimiento de los ingresos recaudados por los servicios educacionales.

END

Date : 24-NOV-90

ORACLE CASE*Dictionary

FUNCION HIEARCHY

APPLICATION : KBUCA2

VERSION : 9

FUNCTION : ADMON

ADMON

Administrar los recursos necesarios para los cursos y las actividades planeadas, como son las inscripciones de alumnos, requisiciones, facturación, etc.

ADMON1

Inscribir alumnos en los grupos por impartir.

ADMON2

Verificar y decidir la impartición de cursos.

ADMON3

Elaborar y entregar la solicitud de factura.

ADMON4

Emitir créditos educacionales.

END

Date : 24-MOV-90

ORACLE : CASE-Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : ADMON1

ADMON1

Inscribir alumnos en los grupos por impartir.

ADMON1.2

Reservar el lugar a un alumno en un grupo.

ADMON1.3

Obtener y registrar la confirmación de asistencia del alumno.

ADMON1.1

Verificar y registrar los datos del cliente.

ADMON1.4

Cancelar una reservación o confirmación de un alumno a un grupo.

ADMON1.5

Recibir petición de impartición de un curso especial.

CONTINUED

Date : 24-NOV-90

ORACLE CASE*Dictionary

FUNCTION HIERARCHY
(CONTINUATION)

APPLICATION : EDUCA2

VERSION : 9

REFERENCE :ADMON1

ADMON1

ADMON1.6

Cambiar a todos los alumnos de un grupo a otro.

END

Date : 24-NOV-90

ORACLE CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : ADMON1.2

ADMON1.2

Reservar el lugar a un alumno en un grupo.

ADMON1.2.1

Determinar el grupo al cual asistirá el alumno.

ADMON1.2.2

Verificar si hay cupo disponible para ese grupo.

ADMON1.2.3

Reservar lugar a uno o varios alumnos y registrar sus nombres.

END

Date : 24-NOV-90

ORACLE CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDWA2

VERSION : 9

FUNCTION : ADMIN1.1

ADMIN1.1

Verificar y registrar los datos del cliente.

ADMIN1.1.1

Si existen datos del cliente, se pueden dejar como están o bien modificados.

ADMIN1.1.2

Si no existen datos del cliente, se registran sus datos

END

Date : 24-NOV-90

ORACLE : PL/SQL DICTONARY

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : ADMON2

ADMON2

Verificar y decidir la impartición de cursos.

ADMON2.1

Verificar el número de alumnos inscritos y confirmados a un grupo determinado.

ADMON2.2

Cancelar grupo y dar aviso de cancelación al cliente.

ADMON2.3

Posponer grupo y dar aviso al cliente (ver PLANB); cambiar la fecha de impartición.

ADMON2.4

Recordar al cliente la impartición del curso.

ADMON2.5

Cambiar alumnos de un grupo a otro y dar aviso al cliente (ver ADMON1.6).

END

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : ADMON3

ADMON3

Elaborar y entregar la solicitud de factura.

ADMON3.1

Realizar la solicitud de factura.

ADMON3.2

Cancelar la solicitud de factura.

ADMON3.3

Verificar y registrar el que las facturas hayan sido pagadas.

END

Date : 24-NOV-90

ORACLE CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAE

VERSION : 9

FUNCTION : ADMON4

ADMON4

Emitir créditos educacionales.

ADMON4.1

Recebir la petición de otorgar créditos educacionales por un vendedor a una empresa.

ADMON4.2

Registrar los datos de la empresa.

ADMON4.3

Otorgar y hacer válidos los créditos educacionales a una empresa.

ADMON4.4

Emitir carta de entrega y dar los créditos educacionales al cliente.

END

Date : 24-NOV-90

ORACLE : CASE Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCA2

VERSION : 9

FUNCTION : DISENO

DISENO

Diseñar y definir el contenido de los cursos, así como el material necesario para su impartición y actividades que conllevan al cumplimiento de los objetivos de entrenamiento

DISENO1

Determinar el objetivo del curso, así como los temas a tratar, profundidad, duración etc.

DISENO2

Determinar la forma de impartir el curso (prácticas, teoría, evaluaciones, etc.) y determinar los recursos necesarios (materiales de apoyo, manuales, etc.).

DISENO4

Revisar y/o redefinir la estructura y efectividad de cursos existentes. (p. ej; modificar duración, temas, precio, etc.)

DISENO3

Registrar el curso diseñado.

END

Date : 14-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDU0A2

VERSION : 9

FUNCTION : IMPART

IMPART

Efectuar la impartición de los cursos y realizar las actividades planeadas.

IMPART2

Impartir curso.

IMPART7

Aplicar las encuestas a los alumnos.

IMPART6

Registrar la asistencia.

IMPART8

Emittir y entregar diplomas.

IMPART3

Obtener lista provisional de asistentes.

CONTINUED

Date : 24-NOV-90

ORACLE CASE DICTIONARY

FUNCTION HIERARCHY
(CONTINUATION)

APPLICATION : EDUCA2

VERSION : 9

REFERENCE : IMPART

IMPART

IMPART4

Validar lista provisional de asistentes.

IMPART5

Registrar lista final de asistentes.

IMPART1

Verificar y reunir los recursos necesarios, antes de dar comienzo el curso (como seria la cafeteria, el salón, manuales y materiales para el curso).

END

Date : 14-NOV-98

ORACLE CASE*DICTIONARY

FUNCTION HIERARCHY

APPLICATION : EDUCAR

VERSION : 9

FUNCTION : IMPART6

IMPART6

Registrar la asistencia.

IMPART6.1

Registrar la asistencia diaria de los alumnos.

IMPART6.2

Registrar la asistencia total el último día de curso.

IMPART6.4

Determinar el derecho a diploma para cada asistente.

IMPART6.3

Actualizar la cantidad de manuales.

IMPART6.5

Actualizar el nivel de experiencia de los instructores que impartieron el curso.

END

Date : 24-NOV-90

ORACLE : CASE*Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAJ

VERSION : 9

FUNCTION : IMPART8

IMPART8

Emitir y entregar diplomas.

IMPART8.1

Elaborar lista de asistentes con derecho a diploma y entregarla a la persona correspondiente.

IMPART8.2

Obtener y entregar diplomas.

END

Date : 24-NOV-80

ORACLE CASE Dictionary

FUNCTION HIERARCHY

APPLICATION : EDUCAZ

VERSION : 9

FUNCTION : IMPART8.2

IMPART8.2

Obtener y entregar diplomas.

IMPART8.2A

Obtener y entregar los diplomas directamente al asistente, en caso de tener estos el ultimo dia de clases.

IMPART8.2B

Obtener diploma, elaborar carta y enviarla junto con el diploma al cliente en caso de no haber sido entregado el diploma el ultimo dia de clases (ver PROFAS2.1).

END

ORACLE CASE Dictionary

END OF FUNCTION HIERARCHY REPORT

PRECOMP.RPT



APÉNDICE B

ORACLE
 CASE Dictionary
 Entities and their Attributes

Application	Version	Entity	Attribute Name	Opt	DT	Size	Dec	Attribute Type
EDUCAD	8	ALUMNO	NUM_ALUMNO	N	IN	5		Identificador único de alumno (asignado por el sistema)
			NOMBRE_ALUMNO	N	CH	30		Nombre del alumno
		APITUB	NIVEL_EXPERIENCIA	Y	CH	1		nivel de dominio del curso que reparte ese instructor (A,B,C)
			CURSON_DADOS	Y	NU	3		cantidad de veces que el instructor repartió el curso
		AREA	CODIGO_AREA	N	IN	6		Identificador único de área (asignado por el sistema)
			CODIGO_DISTRICTO	Y	CH	3		Código postal
			DELEGACION	Y	CH	15		Delegación en la que se encuentra el área
			TELEFONO	Y	CH	40		Teléfono(s), extensiones y fax del área
			RESPONSABLE	Y	CH	50		persona responsable del área
			COMISE_AREA	N	CH	10		nombre del área o departamento
			CALLE_NUM	Y	CH	30		numero de la calle y número
			COLONIA	Y	CH	30		colonia en la que se encuentra el área
			ESTADO	Y	CH	15		Estado

CASE Dictionary Page 14 8/27/91

ORACLE
CASEP Dictionary
Entities and their Attributes

Report/Entity	Version/Attr	Attribute Name	Req	PK	Supp	Default Value
EDUCAD	9 AREA	Ciudad	1	CK	20	Ciudad
		Fax	Y	CV	20	fax del area (con telefono, extension y fax)
		AREAS_RESPONSABLE	Y	CA	20	suena del responsable del Area
ASISTENCIA		INDICADOR_CURSO	N	CK	3	Indica si el alumno que se registra se presento al curso
		DIPLOMA_ENTREGADO	Y	CK	1	Indica si el diploma ha sido entregado (S/N)
		DERECHO_DIPLOMA	Y	CK	1	Indica si el alumno merece diploma del curso o no (S/N)
CIUDAD		CLAVE_CURSO	N	CK	3	Indica clave de curso y ciudad
		NUMERO_CIUDAD	N	CK	35	numero de la ciudad
CREDITO_EDUCACIONAL		CREDITOS_OTORGADOS	N	IN	3	numero de creditos que otorgo el verificador a una empresa
		CLAVE_CREDITOS CASEP Dictionary Reports ATTRIBUTE	N	IN	4	numero de contrato secuencial

ABACUS
CAIMS Dictionary
Entities and their Attributes

Application	Version	Entity	Attribute Name	Opt	PK	Size	Desc	Attribute Domain
								asignado por el sistema
ABACUS	0	CREDITO_EDUCACIONAL	SUB_CONTRATO	Y	10	4		numero de contrato real
			SEDES_POR_UTILIZADO	Y	10	3		numero de créditos hechos efectivos por una empresa
		CURSO	CODIGO_CURSO	4	09	5		referencia al curso
			horas_dias	Y	10	2		numero de horas al día recomendadas para repetir el curso
			DURACION_DIA	1	10	2		numero de días en los que habitualmente se debe de dar el curso
			UNIDADES_EDUCACIONALES	Y	04	2		cantidad de unidades de costo de un curso por persona
			NUMERO_CURSO	N	04	39		numero del curso
		CURSO_EST	COSTO	N	00	12	2	costo del curso
			FECHA_INICIAL_CONTO	N	04			fecha desde la cual se cobra el costo por el curso

CAIMS Dictionary Reports : ATTEN!%

DBASE II
 Database Dictionary
 Entities and their Attributes

Application	Version	Entity	Attribute Name	Opt #1	Size	Opt.	Attribute Description
FOLCLAF	5	DECUENTO	DESCUENTO	N	ML	3 2	Descuento que se le aplica al total final de la factura
			IMPRES_APLIC	N	ML	7	Indica el orden en que se debe de aplicar el descuento
			CONCEPTO_DESC	N	CA	50	razón por la cual se aplica el descuento
	EMPRESA	EMPRESA	EMPRESA_EMPRESA	N	12	4	Identificador único de una empresa (se genera por el sistema)
			DIRIG	N	CA	20	razón corto de una empresa
			EMPRESA_EMPRESA	N	CA	50	razón o razón social de la empresa
FACTURA		FACTURA	FACTURA_FACTURA	N	12	3	Identificador único de factura (asignado por el sistema)
			CANTIDAD_CREDITOS	N	ML	3	numero de créditos utilizados en el pago de la factura
			FECHA_FACTURACION	N	CA		Fecha en la que se hizo la factura

Database Dictionary Report : ATTEC114

ORACLE
CASEPOTENTIAL:
Entities and Their Attributes

Application	Version	Entity	Attribute Name	Opt	SI	Size	Dec	Attribute Notes
ESCANZ		Y EFECTIVA	TOTAL	0	NO	12	2	cantidad total a pagar por el cliente
			CANCELADO	1	CH	1		indica si la factura fue cancelada (Y/N)
			NUM_FACT_TOTAL	1	IN	5		numero total de la factura otorgada al fact. ha sido pagada o no?
		GRUPO	NUM_GRUPO	0	IN	5		numero secuencial asignado por el sistema para cada grupo
			FECHA_INICIO	0	DA			fecha real de inicio de curso
			DURACION_TOTAL	1	IN	3		total de horas repartidas entre ese grupo
			CANCELADO	1	CH	1		indica si el grupo ha sido cancelado o no (Y/N)
			FECHA_FIN	0	DA			fecha real de terminacion de curso
		INSTRUCTOR	CLAVE_INSTRUCTOR	0	CH	5		identificacion unica de instructores (alfabeticos)
			NOMBRE_INSTRUCTOR	0	CH	35		nombre del instructor

ORACLE
SAS Dictionary
Entities and their Attributes

Application	Version	Entity	Attribute name	Opt	PK	Size	Per	DEF/Null/Notes
SASCAF	9	INSTRUCION_DE_GRUPO						
		MANUAL	MAN_NUMERAL	N	18	5		numero unico de manual designado por el sistema
			EXIST_MAN_1987	Y	08	1		indica si existe un manual especifico para el instructor (S/N)
			EXIST_CATEDOR	Y	08	1		indica si existen materiales para impartir el curso (S/N)
			EXIST_ORIGINAL_Man	Y	08	1		indica si existe el manual original (S/N)
			TITULO_MANUAL	N	08	25		titulo asociado al manual
			CANT_COPIAS_MANUAL	N	18	5		cantidad de copias de manual que hay de existencia en el curso

MANUAL_DE_CURSO

SAS Dictionary Reports - ATTRIB74

ORACLE
 CDB*Dictionary
 Entries and their Attributes

Application	Version	Entity	Attribute Name	Opt	Size	Dist	Attribute Notes
EDUCAD	9	PABTIDA	DESCO_PABTIDA	N	60	1	descuento aplicado por curso (%)
			COSTO_TOTAL	N	80	12	costo total de curso aplicando descuentos parciales
			CURSOS_APLICADOS	N	18	2	numero de cursos aplicados a la misma (sección) que formaron curso
			CONCEPTO_MES	N	20	30	meses por los cuales se aplica el descuento
PROSPECTO		HOMBRE_PROSPECTO	N	28	35	nombre del alumno al cual se le reserva lugar en un grupo	
		NUM_PROSPECTO	N	18	6	identificador unico de prospecto (asignado por el sistema)	
RESERVACION			NUM_RESERVACION	N	18	5	numero de solicitud de reservacion (identificador unico)

CDB*Dictionary Supplied Attribute

DBACLE
 CASE Dictionary
 Entities and their Attributes

Application	Entity	Attribute name	Opt	SI	Size	Sec	Attribute notes
COORAF	RESERVA	CARTA_COMPRA	1	CO	1		Indicador de si se entregó la carta compra (S/N).
		QUITO_RESERVA	1	CA	35		Nombre de la persona que pide una reservación para un sistema.
		FECHA_RESERVACION	1	CA			Fecha en la que se hace la reservación de un sistema.
TIPO_CLIENTE		CODIGO_TIPO_CLIENTE	1	CA	3		Código de tipo de cliente, conforme a la clasificación dada.
		DESCRIPCION	1	CA	35		Descripción del tipo de cliente.
TIPO_CURSO		CODIGO_TIPO_CURSO	1	CA	1		Indicador de tipo de curso (en DBACLE, en empresa externa, etc.).
		DESCRIPCION	1	CA	35		Descripción del tipo de curso.

Last Modified Date: 3/27/97

ORACLE
 CASE# Dictionary
 Entities and their Attributes

Application	Version	Entity	Attribute Name	Col ID	Col Len	Attribute Desc
ORACLE	V	VERSION	CLAVE_VERSION	4	08	Identificador unico del version
			NOMBRE_VERSION	4	24	nombre del version

CASE# Dictionary Report: ATR016

BIBLIOGRAFIA

CASE*Dictionary User's Guide and Reference

Version 4.0

Mary Lomas, Sue Lavesey

Oracle Corporation UK Limited Chertsey, Surrey

England - Nov 1988

SQL*Design Dictionary

Version 3.0

Anna Head, Roderick C Low

Oracle Corporation, Belmont, California

U.S.A. - 1989

SQL*Plus Class Notes

Nimish Mehta, Chris Schuck, Mark Rosen

Oracle Corporation, Belmont, California

U.S.A. - June 1988

Introduction to CASE*Dictionary Class Notes

Leta Davis, Leslie Stern, Renee Taylor

Oracle Corporation, Belmont, California

U.S.A. - 1989

SQL*Forms Class Notes

Mark Rosen

Oracle Corporation, Belmont, California

U.S.A. - 1987

The Relational Database Management System

Detailed Systems Analysis

Oracle Corporation UK Limited

England - 1988

Application Performance Tuning Class Notes

Rudy Corsi, Donald Feinberg, Tom O'Connor

Oracle Corporation, Belmont, California

U.S.A. - May 1988

Database Administration Class Notes

Version 5.1

Rudy Corsi, Donald Feinberg, Tom O'Connor

Oracle Corporation, Belmont, California

U.S.A. - June 1988

Relational Database Design

Extended Relational Analysis Class Notes

Relational Systems Corporation 22809 Shagbark, Birmingham, Mi 48010

U.S.A.

Module Design Methodology

Training for Information Processing

Okinawa International Centre

Japan International Cooperation Agency

Japan - 1989

CASE*Method Tasks and Deliverables

The Relational Database Management System

Version 1.0

Richard Barker, Don Clegg, Jeremy Davis

Oracle Corporation UK Limited, Chertsey, Surrey

England - Dec 1988

Oracle CASE Technology

Computer Aided Systems Engineering Products

U.S.A.

Introduction to CASE*Method

Oracle Corporation UK LIMITED

England - 1988

CASE Questions and Answers

Oracle Corporation, Belmont, California

U.S.A. - 1988

Oracle Magazine

The SQL Database Journal

Volume III / Number 2

U.S.A. - Fall 1989

BYTE

The CASE Philosophy

Michael L. Gibson

pp.209 - Apr 1989

BYTE

Methodology: The Experts Speak

Keynote: Chris Crane, Edward Yourdon, Peter P. Chen, Larry L. Constantine

pp.221 - Apr 1989

BYTE

The CASE Experience

Carma McClure

pp.235 - Apr 1989

BYTE

A CASE Workshop

pp.246 - Apr 1989

TECNOLOGIA - CII

Técnicas Estructuradas para el Desarrollo del Sistema de Información

Rodrigo Martínez

México - Ago 1985

Sales Terminology Guide to CAM: Technology

Oracle Corporation, Belmont, California

U.S.A.

CASE Market Analysis

Competitive Bulletin

Alex Mollen

Oracle Corporation, Belmont, California

U.S.A. - Dec 1987