



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

13  
ZES

FACULTAD DE CONTADURIA Y ADMINISTRACION

**APLICACION DE LA FILOSOFIA CASE EN  
EL DESARROLLO DE UN SISTEMA PARA  
EL CONTROL DE EMBARCACIONES EN  
UN CLUB NAUTICO.**

**SEMINARIO DE INVESTIGACION  
I N F O R M A T I C A  
QUE PARA OBTENER EL TITULO DE  
LICENCIADO EN INFORMATICA  
P R E S E N T A N:**

**ALEJANDRO GARCIA CASTRO  
MARIANA PEREZ-VARGAS OBREGON  
OSCAR GERARDO SANCHEZ FLORES  
ARACELI SANCHEZ MARTINEZ**

**DIRECTOR DEL SEMINARIO:  
ACT. FRANCISCO DAVID MEJIA RODRIGUEZ**

**PERIODO 94-2**

**FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

---

***A la Universidad Nacional Autónoma de México:***  
Por brindarnos la oportunidad de superarnos.

***A la Facultad de Contaduría y Administración:***  
Por lo que ahora somos.

***A la Licenciatura en Informática:***  
Por nuestra formación.

***Al Act. David Mejía:***  
Por contar siempre con su paciencia y apoyo incondicionales.

***A todos nuestros profesores:***  
Por haber compartido con nosotros sus conocimientos y sabiduría;  
especialmente al Dr. Ricardo Rivera Soler.

---

***A mis padres:***  
Por su cariño, apoyo, dedicación y confianza.

***A mis hermanos:***  
Por su apoyo incondicional.

***A mis amigos:***  
Mariana, Araceli y Oscar, por cinco años de esfuerzos compartidos.

**Alejandro.**

---

***A mis padres:***  
Con infinito cariño y admiración, por enseñarme siempre a valorar la vida y  
por apoyarme en cada instante de la misma.

***A Alejandro:***  
Con todo mi amor, por su comprensión y apoyo incondicionales, por su  
cariño y ternura.

***A mi hermana Jimena:***  
Con mi cariño por estar y compartir siempre todos los momentos.

***A mi familia materna:***  
Especialmente a mis abuelos Luis Felipe y María Elena; y a la pequeña Laura.

***A todos mis amigos:***  
Que contribuyeron a lograr este proyecto, en particular a Oscar, Araceli y  
Alejandro.

***Al Colegio Madrid.***

**Mariana**

---

***A mis padres:***  
Mil gracias por brindarme la hermosa oportunidad de ser alguien en la vida,  
por ofrecerme su total apoyo aún en las situaciones más adversas, compartir  
mis sueños y ayudarme a cumplirlos.

***A mi hermana:***  
Por su gran ejemplo de superación y perseverancia.

***A mi hermano:***  
Por su gran apoyo en la consecución de mis metas.

***A mi familia:***  
Por estar siempre junto a mí apoyándome; principalmente a mis tíos Juan,  
Manuel y Hugo que han contribuido a mi formación.

***A mis amigos:***  
Por permitir compartir mi vida y mis sueños. En especial a Mariana,  
Alejandro y Araceli, por brindarme su gran amistad a lo largo de estos últimos  
años.

**Oscar**

---

***A mis queridos padres: Eloisa y Raúl***  
Por ser mi ejemplo de amor, rectitud, responsabilidad y superación en la vida.

***A mis hermanas: Maribel y Olivia***  
Por su confianza y apoyo incondicional.

***A Mariana, Oscar y Alejandro:***  
Por brindarme la oportunidad de formar parte de un excelente equipo de trabajo, además ser mis amigos.

**Araceli.**

---

## INDICE GENERAL

<b>Introducción .....</b>	<b>I</b>
<b>Planteamiento del problema .....</b>	<b>IV</b>
Definición del problema	
Justificación ¿Por qué CASE?	
Alcances del proyecto	
<b>Antecedentes de CASE .....</b>	<b>7</b>
Filosofía	
Evolución	
<b>¿Qué es CASE? .....</b>	<b>36</b>
Definición de CASE	
Ciclo de Vida de sistemas	
Herramientas CASE	
Repositorio, Reingeniería y Relacional	
Consideraciones organizacionales para la implantación de productos CASE	
<b>Ambiente CASE para el desarrollo del sistema .....</b>	<b>63</b>
LSA: LINC System Approach	
LDA III: LINC Design Assistant	
<b>Investigación del Sistema .....</b>	<b>83</b>
Plan de Soluciones	
Investigación del Sistema	
<b>Definición del Sistema .....</b>	<b>106</b>
<b>Desarrollo del Sistema .....</b>	<b>143</b>
<b>Conclusiones .....</b>	<b>151</b>
<b>Bibliografía .....</b>	<b>154</b>



---

## **CAPITULO I: ANTECEDENTES DE CASE**

### **1.1 Filosofía**

### **1.2 Evolución**

#### **1.2.1 Programación Estructurada**

##### **1.2.1.1 Refinamiento paso a paso**

##### **1.2.1.2 Top-Down**

##### **1.2.1.3 Bottom-Up**

#### **1.2.2 Análisis Estructurado**

##### **1.2.2.1 Técnicas de diagramación**

###### **1.2.2.1.1 Diagramas de Flujo de Datos**

###### **1.2.2.1.2 Descomposición Funcional**

###### **1.2.2.1.3 Láminas Estructuradas**

###### **1.2.2.1.4 Diagramas HIPO**

###### **1.2.2.1.5 Diagramas de Warnier-Orr**

###### **1.2.2.1.6 Diagramas de Michael Jackson**

###### **1.2.2.1.7 Diagramas de Actividades**

###### **1.2.2.1.8 Árboles y Tablas de Decisión**

###### **1.2.2.1.9 Diagramas de Entidad-Relación**

###### **1.2.2.1.10 Diagramas de Navegación de Datos**

###### **1.2.2.1.11 Diagramas H.O.S.**

##### **1.2.2.2 Análisis Estructurado de De Marco**

##### **1.2.2.3 Metodología de Análisis Estructurado Yourdon**

#### **1.2.3 Diseño Estructurado**

##### **1.2.3.1 Metodología de Diseño Yourdon**

##### **1.2.3.2 Metodología de Diseño Jackson**

##### **1.2.3.3 Metodología de Diseño Warnier/Orr**

##### **1.2.3.4 Metodología H.O.S.**

#### **1.2.4 Modelo de Datos**

##### **1.2.4.1 Modelo de Datos Relacional**

###### **1.2.4.1.1 Antecedentes**

###### **1.2.4.1.2 Componentes**

###### **1.2.4.1.3 Algebra Relacional**

##### **1.2.4.2 Normalización**

###### **1.2.4.2.1 Formas Normales**

##### **1.2.4.3 Reglas de Codd**

---

## **CAPITULO II: ¿QUE ES CASE?**

- 2.1 Definición de CASE
- 2.2 Ciclo de vida de sistemas
- 2.3 Herramientas CASE
  - 2.3.1 Definición y Evolución
  - 2.3.2 Clasificación de los productos CASE
    - 2.3.2.1 Clasificación por función
    - 2.3.2.2 Clasificación por su uso en el ciclo de vida de desarrollo
  - 2.3.4 Categorías de las herramientas CASE
- 2.4 Repositorio, Reingeniería y Relacional
  - 2.4.1 Repositorio central de información
  - 2.4.2 Reingeniería
  - 2.4.3 Relacional
- 2.5 Consideraciones organizacionales para la implantación de productos CASE
  - 2.5.1 Puntos básicos que se sugieren cubrir para implantar CASE
  - 2.5.2 Consideraciones técnicas para seleccionar herramientas CASE

## **CAPITULO III: AMBIENTE CASE PARA EL DESARROLLO DEL SISTEMA**

- 3.1 Enfoque de Sistemas LINC (LSA)
  - 3.1.1 Investigación del sistema
  - 3.1.2 Definición del sistema
    - 3.1.2.1 Modelo basado en objetos
  - 3.1.3 Desarrollo del sistema
  - 3.1.4 Implantación del sistema
  - 3.1.5 Revisión del sistema

- 3.2 Asistente de Diseño LINC (LDA)
  - 3.2.1 Auditoría de Diseño
  - 3.2.2 Análisis de Puntos de Función
  - 3.2.3 Construcción de Prototipos

## **CAPITULO IV: INVESTIGACION DEL SISTEMA**

- 4.1 Plan de Soluciones
- 4.2 Investigación del Sistema
  - 4.2.1 Metas, objetivos y restricciones del sistema
  - 4.2.2 Personal y aptitudes
  - 4.2.3 Estándares del proyecto
  - 4.2.4 Actividades, áreas funcionales y límites de implantación
  - 4.2.5 Requerimientos del usuario
  - 4.2.6 Otros recursos requeridos para la definición del sistema
  - 4.2.7 Resumen del sistema de información
  - 4.2.8 Presentación del proyecto

## **CAPITULO V: DEFINICION DEL SISTEMA**

- 5.1 Estándares de Diseño
- 5.2 Modelo de Objetos LINC
  - 5.2.1 Specs
  - 5.2.2 Reportes
  - 5.2.3 Perfiles y Lógicas Globales
  - 5.2.4 Reglas del Negocio
  - 5.2.5 Diccionario de Datos
  - 5.2.6 Relaciones entre Objetos
  - 5.2.7 Revisión del Diseño
- 5.3 Atributos Técnicos
- 5.4 Tamaño y Complejidad del Sistema

## **CAPITULO VI: DESARROLLO DEL SISTEMA**

### **6.1 Conversión del Modelo de Objetos a especificaciones de LINC**

## INTRODUCCION

Durante los últimos veinticinco años el portafolio de aplicaciones se ha visto afectado por una aguda crisis: la crisis del software.

La causa de esta crisis es provocada por la urgencia de que los sistemas estén funcionando y arrojando resultados lo antes posible, esto ocasiona que la gente que se dedica a desarrollarlos deje a un lado las primeras etapas del desarrollo del software: planeación, análisis y diseño para enfocarse principalmente en su construcción y mantenimiento; dedicando a esta última fase el mayor tiempo ya que los sistemas resultantes no son los deseados, lo que requiere que se realicen cambios constantemente con el fin de que funcionen de manera correcta.

Esta crisis ha ocasionado las siguientes consecuencias:

- Mayor crecimiento del hardware ante el software
- No se aprovechan al máximo los recursos de la máquina
- La fase de mantenimiento es excesiva
- Costos elevados en el desarrollo y mantenimiento de sistemas
- Flujos de información erróneos
- Sistemas ineficientes y costosos

Los desarrolladores de software han tratado de encontrar solución a ésta crisis, para ello han propuesto varias soluciones:

Un primer intento fué la introducción de las técnicas estructuradas, las cuales representan un cambio en la actitud para el desarrollo de sistemas. Estas técnicas proponen que la productividad y la calidad del software se pueden controlar mejor a través de la disciplina, formalidad y estandarización.

A finales de los años 70's surgió otro intento para dar solución a ésta crisis, que era proporcionar herramientas dirigidas al usuario final con el propósito de que desarrollaran sus propias aplicaciones. Esto pretendía incrementar la

productividad del software, utilizando herramientas que simplificaran el desarrollo de sistemas.

Actualmente existe un enfoque diferente para el desarrollo de sistemas: la filosofía CASE (Computer Aided Software Engineering), la cual representa un nuevo intento que se está realizando para dar solución a la crisis del software; ya que los intentos anteriores no han sido una solución completa.

El presente trabajo muestra la aplicación de la filosofía CASE en el desarrollo de sistemas administrativos; para lo cual se tomó como ejemplo el desarrollo de un sistema de control de embarcaciones para una marina turística.

El primer capítulo nos permite conocer de manera general el conjunto de técnicas y herramientas que ayudan al desarrollo de sistemas, estas técnicas han evolucionando a través del tiempo comenzando desde la programación estructurada, pasando por el Diseño Estructurado con las metodologías presentadas por Edward Yourdon, Michael Jackson, Jean Dominique Warnier y Ken Orr; para llegar al Análisis Estructurado con las técnicas de diagramación como herramientas fundamentales de comunicación junto con las metodologías de Análisis Estructurado de Edward Yourdon, y Tom DeMarco. Para finalizar este capítulo, se hace referencia al modelo de datos relacional, así como sus componentes que en conjunto representan una parte fundamental de la filosofía CASE.

En el capítulo dos explicamos CASE y su aplicación en el ciclo de vida de sistemas; se definen las herramientas CASE como la columna vertebral de esta filosofía, su evolución y las categorías existentes, así como su clasificación ya sea por el área funcional en la que se utilizan o por su uso en el ciclo de vida del software (Upper y Lower CASE). Se presenta una parte fundamental de CASE: la Triple "R" (Repositorio, Reingeniería y Relacional); y para concluir el capítulo mencionamos las consideraciones necesarias para la implantación de CASE en el ámbito organizacional.

El siguiente capítulo se divide esencialmente en dos partes, la primera de ellas define específicamente la metodología de sistemas LINC (Language Interface Network Compiler) que se utilizará para realizar la investigación, definición y desarrollo del sistema para el control de embarcaciones, explicando sus etapas, objetivos y tareas. La segunda parte muestra la herramienta de diseño LDA

(LINC Design Assistant), así como sus componentes, los cuales en conjunto sustentan el diseño y parte del desarrollo del sistema que permitirá mostrar la aplicación de la filosofía CASE.

Los tres primeros capítulos proporcionan el fundamento que nos permite dar paso a los capítulos cuatro, cinco y seis que corresponden a la investigación, definición y desarrollo del sistema respectivamente. En el capítulo cuatro mostramos el Plan de Soluciones que define el alcance y los objetivos del negocio que se está analizando, y posteriormente mencionamos los pasos del análisis que utiliza la metodología LSA (LINC System Approach) y que son necesarios para determinar las entradas del diseño para el capítulo cinco.

En el capítulo cinco presentamos la definición del Sistema propuesto, utilizando la herramienta LDA (LINC Design Assistant) como Upper CASE; el objetivo principal de esta etapa es modelar todos los objetos involucrados en la aplicación: reglas del negocio, pantallas, componentes, eventos, relaciones, etc., para obtener como resultado el diseño del sistema.

En el capítulo seis presentamos el desarrollo del sistema; para dar inicio a esta fase se toma como entrada el modelo de objetos definido en la etapa anterior y se continúa su desarrollo hasta donde LDA permite trabajar en una plataforma de PC. En este punto se cuenta con un prototipo perfectamente depurado, que cumple con las necesidades y requerimientos que se plantearon junto con el usuario.

Para finalizar, presentamos las conclusiones en las cuales hacemos una evaluación de los resultados que obtuvimos al utilizar la filosofía CASE, resaltando las ventajas de utilizar esta filosofía y lo que esperamos que en un futuro suceda con CASE.

## **PLANTEAMIENTO DEL PROBLEMA**

### **DEFINICION DEL PROBLEMA**

El objetivo principal que perseguimos en este trabajo, es demostrar que la filosofía CASE se puede aplicar para dar solución a un problema real, administrativo y de negocios. Pretendemos mostrar de forma práctica y aplicativa, como se puede llevar a cabo cumpliendo con los objetivos de CASE (calidad y productividad), la investigación, la definición y el desarrollo de un sistema.

La calidad esta referida en el contexto de que tan eficientes son las herramientas CASE para utilizarse como soporte en el cambio de requerimientos de los usuarios; y la productividad se refiere a los resultados de comparar el construir un sistema utilizando la tecnología CASE, con el equivalente a desarrollarlo utilizando los enfoques y lenguajes de tercera generación.

Para materializar esta idea, tomamos como ejemplo la situación de una Marina Turística que actualmente enfrenta problemas para controlar sus embarcaciones y los servicios que proporciona a las mismas, esto la lleva a no poder brindar un servicio de calidad a sus clientes.

Operadora Náutica La Marina de Acapulco, S.A. de C.V. desde el año de 1989, es una empresa dedicada a brindar servicios exclusivos de marina turística, club náutico y condominios de lujo.

Actualmente, debido al incremento de sus clientes y a la falta de información oportuna, confiable y de calidad; está enfrentando severos problemas que le impiden el cumplimiento de sus objetivos.



## **JUSTIFICACION**

Elegimos la filosofía CASE porque representa un nuevo enfoque para que el desarrollo de sistemas deje de ser una especie de arte oculto y se transforme en un proceso más metódico y dinámico que permita realizar la simulación de actividades del negocio para aumentar la calidad y productividad del mismo.

Como se podrá constatar en capítulos posteriores, el incorporar CASE como filosofía en la construcción de sistemas, proporciona las siguientes ventajas:

- Favorece la creación de prototipos
- Disminuye el equipo de desarrollo
- Maneja un repositorio
- Genera documentación automática
- Permite diagramación automática
- Maneja lenguajes de cuarta generación
- Permite el manejo de reingeniería
- Permite la reutilización del diccionario, documentos, programación, diseños
- Está basado en un modelo de base de datos orientado a objetos
- Reduce el tiempo dedicado a la programación
- Es flexible a los cambios ya que permite que los sistemas crezcan modularmente
- Involucra a toda la organización: dirección, alta gerencia, usuarios, analistas, programadores, etc.
- Realiza reducciones drásticas en tiempo y costo del software
- Mejora la productividad y calidad de los sistemas
- Brinda apoyo durante todas las etapas del ciclo de vida
- Las fases de desarrollo y mantenimiento son más rápidas
- Brinda soluciones con mínimos defectos

Son muchas y muy amplias las facilidades que brinda CASE, y éstas representan las principales razones por las cuales decidimos enfocarnos a esta filosofía.

## **ALCANCES DEL PROYECTO**

Buscando solucionar los problemas de Operadora Náutica La Marina, descritos anteriormente, decidimos enfocar la solución propuesta al análisis (investigación), diseño (definición) y desarrollo de un sistema para el control de embarcaciones.

Es importante mencionar que las dos primeras etapas: investigación y definición, son las etapas a las que se le deben de dedicar más tiempo, ya que del éxito de éstas depende que la fase de desarrollo sea rápida y eficiente; y así lograr que el resultado del sistema sea satisfactorio.

## **CAPITULO I: ANTECEDENTES DE CASE**

Las metodologías de análisis estructurado se pueden definir como un conjunto de técnicas y herramientas cuyo propósito es desarrollar sistemas y software de alta calidad y bajo costo de mantenimiento; éstas, han ido evolucionando a través del tiempo con el fin de permitir un desarrollo automático y rápido de sistemas; por este motivo, las técnicas estructuradas se consideran como antecedentes de la filosofía CASE. Sus características son: ser amigables, rigurosas pero permitiendo flexibilidad, estar orientadas a bases de datos, utilizar lenguajes de cuarta generación y generadores de código, y operar con herramientas de diseño automatizadas.

Los objetivos que persiguen las técnicas estructuradas son:

1. Proporcionar programas de alta calidad con un comportamiento predecible
2. Desarrollar programas fáciles de mantener
3. Simplificar los programas y su proceso de desarrollo
4. Lograr mayor control y predictibilidad en el proceso de desarrollo
5. Acelerar el proceso de desarrollo de los sistemas
6. Disminuir los costos del desarrollo de sistemas

### **1.1 FILOSOFIA DE LAS TECNICAS ESTRUCTURADAS**

Las primeras filosofías sólo se aplicaron a la programación; pero con el tiempo se han extendido hacia el diseño y análisis.

En seguida se mencionan los cuatro principios básicos de la filosofía de las técnicas estructuradas:

Principio de abstracción:

- Consiste en separar u obtener una vista de aspectos que son comunes a cierta realidad y representarlos de forma simplificada, esto permite ver un programa en capas o niveles.

Principio de formalidad:

- Este principio indica la necesidad que existe de utilizar una metodología o bien, un método riguroso en el desarrollo de sistemas.

Principio de divide y vencerás:

- Consiste en dividir los problemas complejos, en un conjunto de problemas pequeños, sencillos, fáciles de entender y solucionar.

Principio de orden jerárquico:

- Este principio muestra que los componentes de una solución deben de estar organizados, y una forma de hacerlo es por medio de árboles jerárquicos, ya que la solución queda entendida y la construcción se realiza de nivel por nivel.

## **1.2 EVOLUCION DE LAS TECNICAS ESTRUCTURADAS**

Principios de los 70's:

Programación Estructurada

Convenciones de codificación estructurada

Programación Top-Down

\* Parnas: Ocultamiento de información

\* Dijkstra: Niveles de abstracción

\* Wirth: Refinamiento paso a paso

Mediados de los 70's:

Diseño Estructurado

\* Yourdon/Constantine: Diseño estructurado

\* Jackson: Metodología de diseño

\* Warnier/Orr: Metodología de diseño

Finales de los 70's:

**Análisis Estructurado**

- \* De Marco: Análisis estructurado
- \* Gane & Sarson: Análisis estructurado (SADT)
- \* Requerimientos de diseño de lenguaje

**Técnicas de Base de Datos**

- \* Modelo de datos relacional

### **1.2.1 PROGRAMACION ESTRUCTURADA**

Nació en 1969 en la conferencia NATO, y su autor es Dijkstra. Está considerada como una metodología que proporciona estructura y disciplina a la forma de los programas; permite construcciones en forma jerárquica, y programas modulares, su objetivo principal es eliminar la sentencia GOTO, sustituyéndola por estructuras de control de alto nivel que hagan mas confiable la programación y disminuyan el número de instrucciones de código.

#### **Objetivos de la Programación Estructurada**

1. Aumentar la confianza en el programa
2. Mejorar la lectura del programa
3. Reducir la complejidad
4. Simplificar el mantenimiento
5. Incrementar la productividad del programador
6. Proporcionar una disciplina por medio de una metodología de programación

### **1.2.1.1 REFINAMIENTO PASO A PASO**

Es un procedimiento básico de nivel general, a partir del cual se descomponen tareas mas detalladas, de esta manera el programa queda dividido en capas o niveles; el nivel mas alto representa la forma mas abstracta y las capas sucesivas sirven para definir los componentes de su propio nivel.

#### **1.2.1.2 TOP-DOWN**

Top-Down es un proceso de descomposición, y como su nombre lo indica empieza con el nivel mas alto (mayor abstracción). El primer paso es crear toda la estructura y definir los componentes pero sin construirlos; conforme avanza el proceso de diseño se van realizando los detalles que constituyen las capas mas bajas.

Con este método, el crecimiento del programa es paulatino, lo que permite que los problemas puedan ser detectados y corregidos en fases tempranas.

#### **1.2.1.3 BOTTON-UP**

Botton-Up es un proceso de concatenación; un programa se construye combinando componentes simples con el fin de obtener componentes mas abstractos y de mayor nivel. Por lo tanto el proceso de construcción comienza del detalle y va hacia el nivel de abstracción.

Las metodologías de programación estructurada pretenden establecer orden y rigor en el proceso de programación, pero desafortunadamente las metodologías existentes no son lo suficientemente rigurosas y exactas, es por ello que no son una herramienta que permita controlar la complejidad ya que carece de programación automática y rigor matemático.

## **1.2.2 ANALISIS ESTRUCTURADO**

El análisis es un proceso para definir los requerimientos de una solución. En esta etapa se analizan las necesidades de los usuarios y las propiedades que debe de tener el nuevo sistema. Se identifican los requerimientos de eficiencia

y las funciones que deben ejecutarse; la salida principal del análisis es la especificación funcional, la cual incluye la definición de reportes, estructura de datos, base de datos, archivos, tablas, etc.

El análisis es un punto crítico en el desarrollo de sistemas; ya que el resultado de éste afecta el desarrollo de todos los pasos que le siguen. El análisis estructurado proporciona un proceso para que paso a paso se ejecute el análisis y se produzca una especificación funcional nueva y mejorada.

Entre las versiones del análisis estructurado están las de los siguientes autores:

1. Tom De Marco
2. Edward Yourdon

Ambas versiones representa una disciplina estructurada basada en los siguientes conceptos:

- Organización jerárquica Top-Down
- Concepto "Divide y Vencerás"
- Comunicación Gráfica
- Herramientas de Documentación

Tom De Marco quien es uno de los autores del análisis estructurado, lo define como: "Proceso de transformación de una cadena de información que contiene operaciones concurrentes y nuevos requerimientos en una descripción rigurosa y ordenada que representa el sistema a construir"<sup>1</sup>; explica que la diferencia básica que existe entre el análisis clásico y el estructurado es la nueva especificación del sistema, la cual es más rigurosa y amigable que los métodos clásicos.

Características de la especificación estructurada:

- Es un modelo gráfico
- Esta particionada
- Es un modelo jerárquico Top-Down

La especificación estructurada se compone de tres elementos:

1. *Diagrama de flujo de datos.*

Representa gráficamente que hace el sistema más no como lo hace.

2. *Diccionario de datos.*

Es la descripción formal de los datos del sistema.

3. *Especificación de los procesos (Minispec).*

Describe que es lo que sucede dentro de la caja del proceso del DFD, explica como se transforma el dato de entrada en dato de salida, se escribe en pseudocódigo utilizando tablas o árboles de decisión.

A continuación se mencionarán las técnicas estructuradas, las cuales constituyen un punto muy importante dentro del proceso de análisis, ya que su objetivo es facilitarlos para lograr la presentación de soluciones gráficas.

### 1.2.2.1 TECNICAS DE DIAGRAMACION

Los diagramas, constituyen una parte importante en la filosofía CASE; ya que ayudan a pensar mas claramente la solución, además son una herramienta esencial de comunicación, involucran a los usuarios finales y constituyen una parte importante de la documentación.

Algunas técnicas de diagramación son: diagramas de flujo de datos, HIPO, de Warnier-Orr, de Jackson, de actividades, de entidad relación, de navegación de datos, así como otro tipo de técnicas como son la descomposición funcional, láminas estructuradas, árboles y tablas de decisión, etc..

Funciones de las técnicas estructuradas:

- Precisan la comunicación entre los desarrolladores
- Estandarizan las interfases entre módulos
- Ayudan en el debugging
- Auxilian en el mantenimiento
- Permite la revisión del diseño
- Contribuyen al ligado automático y a la generación de código



### ***1.2.2.1.1 DIAGRAMAS DE FLUJO DE DATOS (Data Flow Diagrams)***

Los diagramas de flujo de datos (DFD's) sirven para mostrar el flujo del proceso, así como los datos. Son una herramienta de análisis top-down que permiten llegar hasta el detalle.

El DFD es una representación del sistema, muestra los procesos y las interfases entre los datos. Se forma por cuatro componentes básicos: flujo de datos, proceso, almacenamiento y terminador.

*Flujo de datos:* Se representa por medio de una flecha y sirve para indicar como se conectan los procesos.

*Proceso:* Opera o transforma el dato, se dibuja en un rectángulo con las puntas redondeadas.

*Almacenamiento de datos:* Representa un archivo lógico, cada almacenamiento se conecta al proceso por medio del flujo de datos.

*Terminador:* Muestra el origen del dato usado en el sistema y el último recipiente del dato producido por el sistema, este queda fuera del DFD.

### ***1.2.2.1.2 DESCOMPOSICION FUNCIONAL***

Este, es un método que se aplica sobre las funciones que conforman a una organización, se pueden mencionar tres tipos de descomposición funcional, las cuales varían en el grado de verificación que utiliza cada una de ellas:

*Especie 1:*

Este tipo de descomposición funcional sirve para mostrar la estructura de una corporación, se pueden mostrar las áreas funcionales, las cuales se pueden sub-dividir en procesos, y éstos se sub-dividen en actividades.

*Especie 2:*

Una función recibe ciertas entradas y produce salidas, éstas se representan como las funciones matemáticas:  $y = F(x)$ , este tipo de representación permite que se pueda checar que tanto las entradas como las salidas, estén correctas.

*Especie 3:*

Sólo permite hacer cierto tipo de descomposiciones las cuales están definidas por axiomas matemáticos, la metodología HOS, la cual se explica más adelante, concluye que la descomposición funcional se puede dividir en descomposición binaria: el padre proporciona los datos de entrada para los hijos, y también, recibe la salida.

Se requieren tres tipos de descomposición binaria:

1. Se ejecuta la primera función, pasa a la segunda función, y esta se ejecuta.
2. Se ejecuta la primera o la segunda función.
3. Se ejecutan las dos funciones de manera independiente.

Utilizando éstos tres tipos de descomposición funcional, se obtiene automáticamente la generación del código.

**1.2.2.1.3 LAMINAS ESTRUCTURADAS (Structure Charts)**

Las láminas estructuradas son una forma de descomposición funcional y junto con los diagramas de flujo de datos forman una metodología de diseño estructurado; sus componentes son rectángulos y flechas los cuales sirven para representar módulos y sus conexiones. En este tipo de diagramas, existen ciertas reglas:

- Sólo puede haber un módulo raíz, que es el que origina el control.
- El control se pasa del módulo raíz hacia abajo y se regresa al módulo que invocó, para que cuando termine la ejecución se regrese el control a la raíz.
- La relación entre los módulos es unilateral (A puede invocar a B, pero B no puede invocar a A) y un módulo no se puede invocar a sí mismo.

#### **1.2.2.1.4 DIAGRAMAS HIPO (*Hierarchical Input Process Output*)**

Es una técnica que utiliza un conjunto de diagramas para mostrar la entrada, la salida y las funciones de un programa o sistema; el diagrama puede ser a nivel general o detallado.

Tipos básicos de HIPO:

*Tabla visual de contenido:* Es una lista de componentes funcionales.

*Diagramas generales:* Muestran la entrada, la salida y el proceso del componente funcional principal.

*Diagramas detallados:* Presentan la información necesaria para poder entender la función especificada en el diagrama general.

Los diagramas HIPO son herramientas de análisis y diseño de sistemas; no se considera como una metodología completa, ya que no tiene procedimientos que guíen al analista en la construcción de una especificación funcional, o en su defecto, al programador en el diseño del programa.

#### **1.2.2.1.5 DIAGRAMAS DE WARNIER-ORR (*Warnier-Orr Diagrams*)**

Son herramientas que ayudan para lograr programas estructurados, representan gráficamente la estructura jerárquica de un programa o sistema, se dibuja utilizando la hoja de manera horizontal, y en lugar de bloques como la mayoría de los diagramas, utiliza llaves; se leen de izquierda a derecha y de arriba hacia abajo. Los diagramas de Warnier-Orr están formados por tres partes: Entrada, proceso y salida (Begin, process y end).

Una de sus ventajas es que ayudan a describir un programa o sistema en diferentes niveles de detalle, lo que hace que el diseño sea más claro. Uno de los principales problemas de esta técnica de diagramación, es que no está orientada a base de datos.

### **1.2.2.1.6 DIAGRAMAS DE MICHAEL JACKSON (*Michael Jackson Diagrams*)**

Esta técnica, utiliza los diagramas de árbol para representar la estructura de los programas y las estructuras de datos como estructuras jerárquicas, hace énfasis en derivar los programas estructurados de las estructuras de datos y están formados por cuatro elementos: secuencia, selección, iteración y elementación.

Una ventaja de los diagramas de Jackson, es que si se realizó una representación correcta de entradas y salidas de datos se puedan precisar las estructuras de los programas; esto ayuda a la codificación sin problemas, pero es una técnica que no se recomienda para lógica complicada. Su desventaja es que trata a los sistemas de bases de datos como sistemas de archivos.

### **1.2.2.1.7 DIAGRAMAS DE ACTIVIDADES (*Action Diagrams*)**

Fueron diseñados para enseñar fácilmente a los usuarios finales, así como para asistir a éstos en la aplicación de lenguajes de cuarta generación (4GL) como FOCUS, IDEL, MANTIS, RAMIS Y NOMAD.

La construcción básica de los diagramas de actividades, son las ramas o brackets que pueden ser de cualquier longitud. Dentro de una rama puede haber una o varias ramas y estas a su vez, pueden estar anidadas, pero sólo se ejecuta una división de la rama a la vez. Los procedimientos se representan en un rectángulo con puntas redondeadas, y en otro diagrama se hace el detalle del procedimiento. Estos diagramas empiezan desde el nivel general y pueden irse explotando hasta llegar al nivel de detalle.

Las herramientas CASE utilizan estos diagramas de actividades; ya que ayudan a complementar otras técnicas de diagramación. Son fáciles y rápidos de dibujar ya se manualmente o por computadora, y están diseñados para que puedan interactuar con un modelo de datos.

### ***1.2.2.1.8 ARBOLES Y TABLAS DE DECISION***

Los árboles y tablas de decisión, no se diseñaron como técnicas de diagramación de computadoras, ya que su aplicación es amplia y se extienden a varias áreas.

Los árboles de decisión son un modelo de función discreta en la cual se determina el valor de una variable, y dependiendo del valor que tome se ejecuta una acción. Estos diagramas sirven para proporcionar una visión gráfica de la decisión que se hizo.

Cada vez que se ejecuta un árbol de decisión, se sigue un camino que va desde la raíz hasta el final del árbol, pasando por las ramas de éste.

Las tablas de decisión son un modelo alternativo para una función, la cual se muestra por medio de una matriz, éstas pueden ser tan extensas como sea necesario ya que no tienen ninguna restricción. Esta técnica se utiliza cuando existen muchas condiciones ya que es más fácil de entender, si existen pocas condiciones se recomienda el uso de los árboles de decisión, ya que ambas técnicas sirven para el análisis y diseño de sistemas que involucran lógicas complejas.

Estas dos técnicas (árboles y tablas de decisión) se utilizan como soporte para complementar otras herramientas de diseño como los diagramas estructurados, los Warnier-Orr y los de actividades.

### ***1.2.2.1.9 DIAGRAMAS DE ENTIDAD-RELACION (Entity-Relationship Diagrams)***

Los diagramas de entidad-relación (E-R) tuvieron su origen en los trabajos desarrollados por Peter Chen en 1976. Estos diagramas sirven para mostrar un conjunto de entidades que intervienen en un proceso, sus relaciones entre sí y como éstas relaciones se pueden afectar con el proceso; se representan por medio de rectángulos interconectados a través de ligas las cuales muestran las asociaciones que existen entre las entidades que conforman un modelo.

### **1.2.2.1.10 DIAGRAMAS DE NAVEGACION DE DATOS (Data Navigation Diagrams)**

Estos diagramas muestran como es la navegación a través de una base de datos tradicional y la secuencia en que se accesan los registros de un procedimiento dentro de un modelo de datos. El uso de estos diagramas permite que el diseño de los procedimientos sea fácil y que después se traduzcan a código estructurado.

Los diagramas de actividades y los de navegación se usan en conjunto; los últimos establecen las entidades principales de un procedimiento, examinan el entorno de las entidades y establecen la secuencia de acceso por medio de un sub-conjunto del modelo de datos; posteriormente los diagramas de navegación se convierten en diagramas de actividades.

### **1.2.2.1.11 DIAGRAMAS HOS (Higher-Order Software)**

Esta técnica utiliza la descomposición funcional rigurosa, la cual está basada en reglas matemáticas. Los diagramas de HOS fueron creados por M. Hamilton y S. Zeldin, y se implementan con herramientas CASE.

Utiliza árboles binarios, cada nodo del árbol representa una función que tiene uno o más objetos en la entrada como en la salida. La entrada se escribe del lado derecho de la función y la salida del lado izquierdo. Si las funciones resultan complejas, se puede dividir en sub-funciones, esta descomposición se realiza por medio de estructuras de control como el JOIN, INCLUDE y OR, las cuales se pueden combinar para formar mapas de control.

Esta técnica tiene varias ventajas: tiene bases matemáticas lo que permite concluir que las estructuras de control son correctas, proporciona gráficas por medio de la computadora con el fin de crear rápidamente estructuras complejas, automatiza la generación del código ejecutable, elimina la necesidad de realizar pruebas dinámicas y al tener verificación automática y cruzada puede crear especificaciones de nivel complejo.

### 1.2.2.2 ANALISIS ESTRUCTURADO DE DE MARCO

El análisis estructurado de De Marco, consta de siete pasos al final de los cuales se consigue la especificación estructurada.

*PASO 1: Construir el modelo físico actual*

Documentar la forma en que se hacen actualmente las cosas, esto se representa a través de un DFD físico, y sirve para mostrar ubicación, personal, nombres, procedimientos manuales, automáticos, etc..

*PASO 2: Construir el modelo lógico a partir del modelo físico*

Utilizar el DFD del paso uno para crear el modelo lógico del sistema actual. La diferencia es que el modelo físico muestra los detalles físicos de como se hacen las cosas, y el lógico muestra que se hace en los niveles abstractos.

*PASO 3: Construir el modelo lógico del sistema nuevo*

Desarrollar un nuevo modelo lógico para el nuevo sistema, se utiliza como base el modelo del paso dos y a partir de éste se realizan los cambios. Se crea una especificación estructurada que contenga DFD's, diccionario de datos y la especificación de los procesos.

*PASO 4: Crear una familia de los nuevos modelos físicos.*

Crear un nuevo modelo físico para el sistema que se va a construir, identificando las interfases humanas y con la máquina. Se realizan varios DFDs con el fin de representar varios grados de automatización.

*PASO 5: Realizar una estimación*

Hacer una estimación costo-tiempo para cada alternativa del paso cuatro.

*PASO 6: Seleccionar un modelo*

Seleccionar el modelo físico más adecuado para determinar las funciones y requerimientos del sistema.

*PASO 7: Empaquetar la especificación en subsistemas.*

Afinar la especificación estructurada hecha hasta el paso seis, con el fin de conjuntar DFDs, diccionario de datos y Minispees.

### **1.2.2.3 METODOLOGIA DE ANALISIS ESTRUCTURADO YOURDON**

Edward Yourdon es el autor de esta metodología, en el año de 1974 fundó Yourdon Inc. la cual es una organización que se dedica a impartir seminarios de capacitación en Ingeniería de Software y consultoría en desarrollo de sistemas así como en ambientes CASE.

La Metodología de Análisis Yourdon (Modelo Esencial), describe lo que tiene que hacer el sistema mostrando la lógica de las actividades y la descripción de los movimientos de los datos dentro del sistema. Este modelo incluye las siguientes partes (Ver figura 1.1):

**Propósito del Sistema.**

Declara la razón por la cual existirá el sistema.

**Aspecto del Medio Ambiente.**

Describe dónde se ubica el sistema dentro del negocio.

**Aspecto de Información.**

Describe la información utilizada por el sistema y cómo cambia a través del tiempo.

**Aspecto del Comportamiento.**

Describe los procesos que forman parte del sistema.



## PARTES DEL MODELO ESENCIAL (ANALISIS YOURDON)

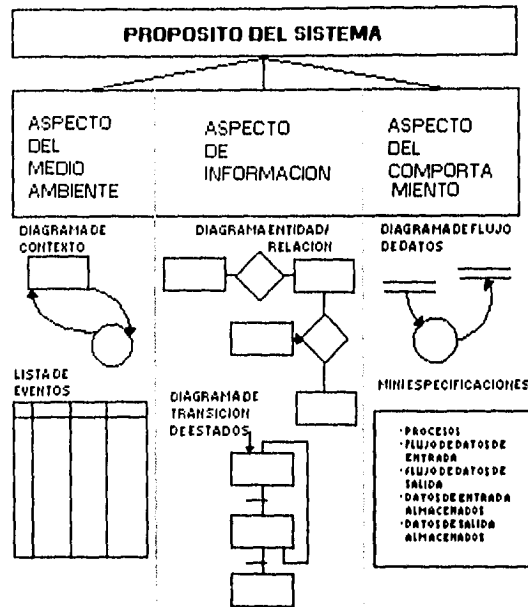


FIGURA 1.1

### 1.2.3 DISEÑO ESTRUCTURADO

Como se pudo observar, en el marco evolutivo de las técnicas estructuradas primero nacen las técnicas relacionadas con el diseño de sistemas; esto es porque lo último que se desarrollo fueron la fase del análisis y las técnicas de base de datos, es decir, el proceso evolutivo se inició al revés de como funciona el ciclo de desarrollo de sistemas.

El diseño es un proceso que consiste en planear la construcción de un sistema, se determinan los componentes de los datos y los procedimientos, la forma en que estos se ensamblan, y se describen los algoritmos. Las entradas del diseño son las salidas del análisis: especificación funcional, y los requerimientos del problema.

Según la definición de Stevens, Myers, Constantine y Yourdon, el diseño es "un conjunto de técnicas, estrategias y métodos para diseñar sistemas de software y programas."<sup>2</sup>

Hay dos tipos de diseño:

1. *Diseño de sistemas*: Se realiza un diseño general en el cual se definen los componentes procedurales básicos y sus interrelaciones, así como las principales representaciones de datos. Se diseña la arquitectura del programa.

2. *Diseño detallado*: Se especifican los algoritmos y se eligen las estructuras de datos. Se refina el diseño de sistemas para crear un diseño más detallado, para lograr esto se requieren varios niveles de refinamiento.

Las metodologías de diseño estructurado utilizan la descomposición funcional como un mecanismo primario de diseño; generalmente las metodologías más utilizadas como Yourdon, Warnier/Orr y Jackson, utilizan el diseño top-down. El producto del diseño estructurado es un diagrama estructurado que muestra los componentes del programa, su jerarquía, su posición dentro del programa y la conexión entre los componentes.

### 1.2.3.1 METODOLOGIA DE DISEÑO YOURDON

La metodología de Diseño Yourdon (también llamado Modelo de Implantación), hace énfasis en la implantación de tecnología específica para el funcionamiento del sistema; muestra la gente y máquinas que desempeñarán las actividades, así como también la forma y secuencia en tiempo del desplazamiento de los datos dentro del sistema. Este modelo incluye las siguientes partes:

**Modelo de Implantación del Usuario:** Describe el funcionamiento del modelo de implantación desde el punto de vista del usuario final, se presenta también información adicional como definición de interfases, diseño de pantallas, explicación de mensajes de error, etc.

**Modelo de Implantación del Sistema:** Permite delinear por cada procesador las unidades de ejecución (conjunto de instrucciones), unidades de almacenamiento, así como las interfases, para contar con una adecuación conforme a la arquitectura del Software.

**Modelo de Implantación de Programas:** Describe para cada unidad de ejecución, la localización de funciones y módulos, así como las interfases entre todos los módulos.

### 1.2.3.2 METODOLOGIA DE DISEÑO JACKSON

La metodología Jackson así como el diseño estructurado son un refinamiento del diseño top-down.

Elementos que proporciona esta metodología:

- Pasos de diseño específicos
- Técnicas gráficas de diagramación
- Métodos para evaluar que el diseño esté correcto

Esta metodología separa la fase de implantación de la fase de diseño, está orientada a los datos (data-oriented), deriva la estructura del programa de la estructura de datos, asume que el problema está bien especificado y que el programa se va a implantar en lenguaje de segunda o tercera generación.

Pasos para el diseño:

#### *1. Paso de dato:*

Describir cada entrada y salida de los datos en forma jerárquica a través de un diagrama de árbol.

#### *2. Paso de programa:*

Tomar todas las estructuras de datos del paso uno y colocarlas dentro de un programa estructurado.

#### *3. Paso de operaciones:*

Hacer una lista de las operaciones que se deben ejecutar para que con los datos de entrada se obtengan salidas, después, ubicar cada operación de la lista en un componente del programa estructurado.

#### *4. Paso de texto:*

Transcribir el programa estructurado en texto estructurado, añadiendo las condiciones lógicas que gobiernan la ejecución de ciclos y la selección de estructuras.

La Metodología Jackson utiliza tres técnicas de diagramación para el diseño de programas:

1. *Diagrama de red del sistema:* Flujo de los datos en los programas.
2. *Diagrama de árbol:* Representación jerárquica del programa y de las estructuras de datos.
3. *Texto estructurado:* Pseudocódigo.

La mayor limitación de la metodología Jackson es que no se puede aplicar directamente a problemas reales, primero porque el proceso de diseño asume la existencia de especificaciones correctas lo cual es poco frecuente para la mayoría de las aplicaciones de datos, en segundo lugar éste proceso solo se limita a programas sencillos y en tercer lugar el diseño está orientado a sistemas de procesamiento en batch, por lo que resulta ineficiente para sistemas en línea o de base de datos.

### **1.2.3.3 METODOLOGIA DE DISEÑO WARNIER-ORR**

Esta metodología tiene su origen en los años 50's en París, su autor fué Jean-Dominique Warnier quien creó la parte de la metodología para la construcción lógica de programas (Logical Construction Program LCP); a mediados de los 70's Ken Orr, basándose en la metodología de diseño Jackson, modificó la metodología LCP de Warnier la cual se denominó diseño estructurado de programas (Structured Program Design SPD); así pues la metodología de diseño Warnier-Orr es un híbrido que resultó de la combinación de la metodología LCP y SPD.

Esta metodología utiliza la notación de la teoría de conjuntos para representar los diagramas, sólo que estos se representan verticalmente.

Es un método top-down ya que utiliza la descomposición funcional para derivar el diseño del programa. La metodología Warnier-Orr es similar a la de Jackson ya que ambas derivan los programas de las estructuras de datos, utilizan estructuras de datos jerárquicas y separan el diseño lógico del físico.

Pasos del diseño:

1. *Definir los procesos de salida:* Representar cada programa de salida como una estructura de datos jerárquica.
2. *Definir la base de datos lógica:* Definir todos los elementos necesarios para producir un programa de salida.
3. *Ejecutar el análisis de eventos:* Definir todos los eventos que puedan afectar a los datos en la base de datos lógica.
4. *Desarrollar la base de datos física:* Definir los archivos físicos para la entrada de datos.
5. *Diseñar el proceso lógico:* Diseñar el proceso lógico del programa que es necesario para producir la salida deseada.
6. *Diseñar el proceso físico:* Añadir los procedimientos de control lógico y manejo de archivos para completar el diseño del programa.

La metodología de Warnier-Orr se basa en el modelo de entrada-proceso-salida; siendo el punto mas importante la salida, es por ello que hace una descripción detallada de las salidas que debe producir el sistema, haciendo que el proceso de diseño se realice de atrás hacia adelante (backward).

Los diagramas de esta metodología son una ventaja, ya que representan secuencias de acciones organizadas jerárquicamente, pero al mismo tiempo constituyen una limitación ya que sólo permiten la organización jerárquica, siendo que no todas las bases de datos presentan este tipo de organización; otro problema de los diagramas es que no funcionan para problemas muy grandes y con mucho anidamiento. El punto más crítico de esta metodología es la orientación a las salidas ya que funciona para problemas pequeños que

solo manejan reportes. Así pues; esta metodología se recomienda para sistemas pequeños orientados a reportes, ya que no reconoce ningún tipo de estructura más que la jerárquica, no esta orientada a base de datos y no proporciona elementos para corregir y verificar el diseño.

### **1.2.3.4 METODOLOGIA H.O.S. (Higher-Order Software Methodology)**

Esta metodología es la base de la filosofía CASE para crear aplicaciones basadas en un conjunto de axiomas matemáticos que garantizan que el resultado del diseño va a ser lógicamente completo y consistente.

Esta metodología utiliza el software denominado USE.IT que proporciona:

- Un lenguaje para expresar funciones y su descomposición
- Una pantalla interactiva para construir y manipular los mapas de control
- Una biblioteca que contiene tipos de datos, funciones primitivas y módulos definidos
- Una rutina analizadora para la verificación automática
- Un generador de código

Consta de tres elementos:

*1. AXES:* Lenguaje para describir las funciones jerárquicas de H.O.S.

*2. Analizador:* Verifica que las reglas matemáticas cumplan con las especificaciones para garantizar que la estructura lógica de la aplicación sea correcta.

*3.R.A.T.:* Convierte la lógica que ha verificado el analizador en código listo para ejecutarse.

## 1.2.4 MODELO DE DATOS

La evolución de las técnicas estructuradas, se han visto beneficiadas con la utilización de los Modelos de Datos.

Para comprender su significado, es necesario definir que un modelo es una representación de un fenómeno, este se obtiene por medio de abstracción, en la cual se omiten detalles y se concentra en lo general. Abstrayendo se puede reducir la complejidad y ayuda a la comprensión de los elementos del modelo, dándole mayor estabilidad, además de permitir ver el mismo objeto pero de diferentes formas.

Existen dos tipos de abstracción:

**Generalización:** se usa para abstraer conceptos y agrupa un conjunto de ocurrencias; se utiliza al manejar archivos de tipo genérico de objetos, que tienen propiedades comunes en muchos registros.

**Agregación:** se usa para asignar las propiedades que nos interesan de un objeto, para distinguirlo y describirlo; se utiliza en archivos que agrupan campos.

Un modelo de datos es una herramienta que sirve para visualizar las interrelaciones de los datos y pretende capturar el significado parcial de los mismos (ya que uno total no es posible); está compuesto por Reglas Genéricas y Operaciones.

Las Reglas Genéricas o propiedades estáticas se componen por estructuras de datos, que se emplean para representar en el modelo los objetos y sus propiedades; y por restricciones de integridad que se utilizan para tratar de asegurar que cualquier transformación a las estructuras de datos produzca un resultado consistente.

Las Operaciones o propiedades dinámicas representan las transformaciones que sufren los objetos (estructuras de datos), las cuales están sujetas a las restricciones de integridad definidas en el modelo: lectura, eliminación, modificación, inserción.

## **1.2.4.1 MODELO DE DATOS RELACIONAL**

### **1.2.4.1.1 ANTECEDENTES**

Dado el crecimiento en tamaño y complejidad de los sistemas, fue necesario estudiar la optimización de técnicas que facilitaran su desarrollo a costos razonables. Uno de los estudios más importantes fueron los trabajos de N. Dijkstra sobre teorías de lenguajes de programación y programación estructurada, gracias a la aportación de ésta última se avanzó gran terreno en el ámbito de las computadoras.

En el año de 1968, Edgar F. Codd inició sus investigaciones en computadoras dedicadas al almacenamiento de datos para posteriormente crear un modelo abstracto de información. En 1970, Codd publicó un artículo que habla sobre el Modelo Relacional el cual revolucionó las teorías de modelos de datos y para finales de los años 80' se convirtió en el estándar de la industria.

Se propusieron ciertos objetivos del Modelo de Datos Relacional que son:

- 1. Independencia de Datos:* Diferenciar entre lo físico y lo lógico de los datos.
- 2. Comunicación:* Permitir un canal abierto para que los programadores y usuarios finales tengan una buena comprensión y comunicación acerca de los datos.
- 3. Procesamiento de Conjuntos:* Poder manipular grandes volúmenes de datos en una sola operación.

### **1.2.4.1.2 COMPONENTES**

Los componentes del Modelo de Datos son:

#### **Estructuras**

La estructura básica del modelo relacional es precisamente la relación, la cual denota una conexión entre un par de objetos, pudiendo haber relaciones formadas por más de dos de ellos.



El nombre con el que se puede conocer a una relación es el de entidad, el cual es un conjunto finito de atributos. Dichos atributos son etiquetas que enriquecen o complementan la información que se tiene o requiere de esa entidad. A cada atributo en una entidad corresponde un dominio. Un dominio es un conjunto de valores que puede tomar un atributo.

Ahora bien, una relación es la extensión de la entidad, un conjunto finito de mapeos del producto cartesiano de los dominios.

Un tuplo es un elemento de una relación (se le puede considerar como un registro).

La llave primaria es un atributo que permite identificar de manera única a una entidad en un sistema específico, así como a cada una de las ocurrencias. Para definir la llave primaria se deben considerar los atributos mínimos para identificar a la entidad.

#### **Restricciones de Integridad.**

- **Valores Nulos**

Es la ausencia de valor para un atributo, porque no se conoce o no aplica. Es necesario recalcar que cualquier atributo puede tener valor nulo excepto la llave primaria.

*Razones de existencia de los nulos:*

Se crea un tuplo nuevo y en ese momento no se conoce el valor del atributo.

Se agrega un nuevo atributo a una tabla ya existente.

Se desea obtener un agregado del valor de un atributo.

- **Integridad de Entidades**

Una entidad se distingue por medio de una llave primaria, debido a que la identifica de manera única, dicha llave no puede aceptar un valor nulo, ya que al permitirlo, no se garantizaría la unicidad del valor.

- **Integridad Referencial**

Es el medio por el cual, la llave primaria de una entidad se conecta con otra entidad (para ella llave foránea). La llave foránea es el mecanismo mediante el cual podemos conectar, ligar o complementar dos o más entidades.

Una llave foránea siempre hace referencia a una llave primaria en la misma entidad o alguna otra. En una relación, pueden existir  $n$  llaves foráneas, éstas pueden formarse por uno o más dominios, siempre que la correspondencia con cada dominio de la llave primaria al que hace referencia sea UNO a UNO.

Los valores no nulos de la llave foránea se deben contener en el dominio correspondiente de la llave primaria.

Reglas de integridad referencial:

- Controlar los cambios que se hacen a los datos en las entidades.
- Lo cambios deben ser controlados debido a que existen conexiones entre entidades.
- Las reglas determinan que acción se debe tomar al agregar, borrar y modificar tuplos en una entidad.

#### *Reglas para Agregar*

El valor de la llave foránea debe ser nulo o deberá ser igual a algún valor del dominio de la llave primaria a la que hacer referencia.

#### *Reglas para Borrar*

Si se desea borrar un tuplo de una entidad, la cual contiene la llave primaria y existe una relación con otra entidad (que utilice esa llave primaria como foránea) se tienen las siguientes opciones:

**Borrado restringido:** Solo podrá ser eliminado el tuplo si es que no existe en la otra entidad con la que se relaciona un valor igual.

**Borrado en cascada:** Al eliminar el tuplo, de igual forma deberán eliminarse en la otra entidad todos los valores iguales al que se eliminó.

**Borrado con nulificación:** Al eliminar el tuplo, los valores iguales a este de la llave foránea en la otra entidad deberán cambiarse a nulos.

#### *Reglas para modificar*

Existen dos casos, cuando se desea modificar el valor de la llave primaria y cuando se desea modificar el valor de la llave foránea.

Primer caso:

1. Modificación restringida.
2. Modificación en cascada.
3. Modificación con nulificación.

Segundo caso:

El nuevo valor asignado a la llave foránea debe ser igual a alguno de la llave primaria o bien nulo.

### **1.2.4.1.3 ALGEBRA RELACIONAL**

Una gran ventaja del modelo relacional es que cualquier entidad es manejada como un conjunto. Cuando dos entidades se han definido sobre los mismos dominios, se pueden considerar como conjuntos del mismo universo: el conjunto de todos los posibles tuplos en la relación resultante del producto cartesiano de esos dominios. A dicho conjunto se le pueden aplicar las siguientes operaciones:

- **Unión:** El total de los valores de los dominios que están tanto en una entidad como en otra en una relación.
- **Intersección:** Solo los valores de los dominios que se encuentran en una entidad y que también están incluidos en la otra entidad.
- **Diferencia:** Solo los valores de los dominios que se encuentran en una entidad y que no se incluyan en la otra entidad.
- **Producto:** El producto cartesiano de dos relaciones, es decir, el conjunto de todos los posibles tuplos.
- **Permutación:** Esta operación es solo aplicable a una relación. Sirve para modificar el orden de los dominios. Cuando se cambia este orden, obtenemos como resultado una nueva relación, que a su vez es la permutación de la relación original.
- **Proyección:** Operación que es también aplicable a una sola relación. Con ella podemos obtener una nueva relación, la cual es un subconjunto de los dominios de la relación original.
- **Restricción:** Es la operación que se genera al aplicar una restricción a una relación, mediante la cual obtenemos como resultado un subconjunto de tuplos.

- **Join:** Es una operación binaria que combina dos relaciones usando los dominios que tienen en común, dando como resultado otra relación. La única condición que deben cumplir las relaciones para poder realizar un *join* es que exista una intersección no vacía entre cualesquiera de los dominios en ambas relaciones. La nueva relación resultante, tendrá concatenados todos los dominios de la primera y segunda relación.

## **1.2.4.2 NORMALIZACION**

Es una técnica que se utiliza para ejemplificar la estructura lógica de los datos. El proceso de normalización identifica los datos redundantes que puedan existir en la estructura lógica, determina las llaves únicas necesarias para el acceso a los datos y ayuda a establecer las relaciones entre los datos.

### **1.2.4.2.1 FORMAS NORMALES**

Son restricciones que se han definido para el manejo de estructuras relacionales con el propósito de evitar anomalías al agregar, borrar o modificar tuplos en una base de datos. A continuación se mencionan solo las tres primeras de ellas, aunque en la actualidad se manejan algunas más.

#### **Primera Forma Normal. (1FN)**

Una entidad estará en *1FN* si los valores de cada atributo de ésta son atómicos.

#### **Segunda Forma Normal. (2FN)**

Una entidad estará en *2FN* si está en *1FN* si no existe un atributo no primario de la entidad que dependa parcialmente de la llave primaria de esta. Es decir, cada columna que no es llave depende enteramente de la llave primaria.

#### **Tercera Forma Normal. (3FN)**

Una entidad estará en *3FN* si está en *2NF* y no existe un atributo no primario que dependa transitivamente de la llave primaria de la entidad. Una columna que no es llave puede depender de otra que tampoco sea llave

### 1.2.4.3 REGLAS DE CODD

Edgar F. Codd propone doce reglas aplicables a un *Manejador de Bases de Datos Relacional (DBMS)*; teóricamente no podemos hablar de que exista un *DBMS*, ya que ninguno cumple estrictamente con ellas. A continuación, mencionamos las reglas:

0.

“Cualquier *DBMS* que proclame ser relacional, deberá manejar, completamente, las estructuras de datos por medio de sus capacidades relacionales.

#### 1. REGLA DE INFORMACION (*The Information Rule*)

Toda la información dentro de una estructura de datos relacional se representa de manera explícita, a nivel lógico, y exactamente de una manera, como valores de una tabla.

#### 2. REGLA DE ACCESO GARANTIZADO (*Guaranteed Access Rule*)

Se garantiza que todos y cada uno de los datos pueden ser leídos recurriendo a una combinación de nombre de la tabla, valor de la llave primaria y nombre de la columna.

#### 3. TRATAMIENTO SISTEMATICO DE VALORES NULOS. (*Systematic Treatment Of Null Values*)

En un *DBMS* relacional se soportan los valores nulos, para representar información faltante o no aplicable de una forma consistente, independientemente del tipo de dato.

#### 4. CATALOGO DINAMICO EN LINEA BASADO EN EL MODELO RELACIONAL (*Dynamic Online Catalog Based On The Relational Model*)

La descripción de las estructuras de datos se representa a nivel lógico de la misma forma que los datos ordinarios, de modo que los usuarios puedan aplicar el mismo lenguaje relacional para consultarla.

#### 5. REGLA DE SUBLenguAJE PARA LA COMPRESION DE DATOS (*Comprehensive Data Sublenguaje Rule*)

Un *DBMS* relacional deberá contar con un lenguaje completo que contemple definición de datos, manipulación de datos y control de acceso.

**6. REGLA DE ACTUALIZACION DE VISTAS (View Updating Rule)**

Todas las vistas que teóricamente sean actualizables, deberán poder ser actualizadas por medio del sistema.

**7. INSERCIÓN, ACTUALIZACIÓN Y BORRADO DE ALTO NIVEL (High-Level Insert, Update And Delete)**

La posibilidad de manejar una relación base o una relación derivada como un solo operador se aplica a la lectura, inserción, modificación y eliminación de tuplos.

**8. INDEPENDENCIA FISICA DE DATOS (Physical Data Independence)**

Los programas aplicativos y la actividad en terminales no deberán ser afectados por cambios en el almacenamiento físico de los datos o en los métodos de acceso.

**9. INDEPENDENCIA LOGICA DE DATOS (Logical Data Independence)**

Los programas aplicativos y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo, en las estructuras de datos, que preserven la información y que teóricamente permitan la no afectación.

**10. INDEPENDENCIA DE INTEGRIDAD (Integrity Independence)**

Las restricciones de integridad deberán poder definirse con el mismo lenguaje de definición de datos y deberán almacenarse en el catálogo y no en los programas aplicativos.

**11. INDEPENDENCIA DE DISTRIBUCION (Distribution Independence)**

Un DBMS relacional deberá tener independencia de distribución.

**12. REGLA DE NO SUBVERSION (Non Subversion Rule)**

Si un DBMS relacional tiene un lenguaje de bajo nivel (que opere un registro a la vez), ese lenguaje no deberá poder emplearse para subvertir las reglas de integridad y las restricciones expresadas en el lenguaje relacional.”<sup>3</sup>

- <sup>1</sup> T. de Marco, Structured Analysis and System Specification. (Guide 47). Chicago: IBM Guide, 1978.
- <sup>2</sup> W. Stevens, G. Myers and L. Constantine, "Structured Design", IBM System Journal, 13 no. 2, 1974.
- <sup>3</sup> Adad, Rubén; Medina, Miguel Angel; Careaga, Alfredo. "Fundamentos de las Estructuras de Datos Relacionales". Grupo Noriega Editores. México. 1993.

## CAPITULO II: ¿QUE ES CASE?

El presente capítulo tiene como objetivo mostrar qué es y qué engloba la filosofía CASE; el marco evolutivo a partir de su surgimiento hasta principios de la década de los noventas y las diferentes herramientas existentes.

En esta sección también se muestran dos de las clasificaciones que consideramos más importantes para las herramientas CASE, el criterio de la primera es en base a las tareas que realizan y el de la segunda, por su ubicación en el ciclo de vida de sistemas.

Por último, hacemos algunos comentarios sobre el impacto que puede provocar la implantación de CASE en el ámbito organizacional.

### 2.1 DEFINICION DE CASE

“C.A.S.E.: Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering). CASE es una palabra que surge de la combinación de dos conceptos:

C.A.S.A.: Computer Aided System Analysis.

C.A.P.: Computer Aided Programing.”<sup>1</sup>

“CASE puede ser definido como un enfoque de ingeniería disciplinado y estructurado para el desarrollo de software y sistemas, confiables y de bajo costo.”<sup>2</sup>

Podemos considerar a CASE como una filosofía debido a que estudia las causas y razones para lograr un nuevo enfoque en la creación y desarrollo de sistemas; su propósito no es simplemente incrementar la productividad del programador con un número determinado de herramientas, sino establecer un ambiente donde todos los esfuerzos estén coordinados con el objetivo de alcanzar un incremento global en la productividad entre programadores, analistas, usuarios y gerentes.



“CASE realiza la transición de un mundo de enfoques únicos e individuales en el desarrollo de software, a un mundo disciplinado, estructurado, estandarizado de ingeniería de software.”<sup>3</sup> Es decir realiza un cambio en el sentido de que las personas que desarrollaban sistemas lo efectuaban de una manera personalizada e individualista, sin importar si su trabajo sería comprendido por otra persona o si los resultados que él está obteniendo son los deseados por el grupo de desarrollo; el nuevo enfoque CASE se basa en un marco de trabajo, el cual consta de una participación constante y disciplinada de todas las partes involucradas en el proyecto y de una estandarización total de los conceptos, buscando obtener una mayor automatización del proceso de desarrollo de sistemas.

CASE como una filosofía de desarrollo de sistemas requiere del establecimiento de un ambiente donde todos los esfuerzos están coordinados y todos sepan lo que está pasando. Este concepto está muy enfocado al área gerencial, que es la que controla el aspecto administrativo; es importante mencionar que CASE involucra tanto al personal de desarrollo de sistemas, al área gerencial y al usuario de sistemas. La idea es que mientras más personas participen de todas las áreas involucradas con el desarrollo de un sistema, el resultado que se obtenga será mejor para todos.

Como podemos ver CASE no es una revolución en la ingeniería de sistemas, es más bien una nueva propuesta con un enfoque diferente al concepto de ciclo de vida de software que se basa en la automatización.

CASE surge como una parte importante de la solución a la crisis del software, pero no puede considerarse como la solución total. Contribuye agilizando el proceso de desarrollo de sistemas de información haciéndolos más eficientes y productivos para que cumplan con las necesidades y requerimientos de los usuarios, objetivos planteados, tiempos estimados, etc., aspectos que los viejos métodos de desarrollo no cubrían.

## 2.2 CICLO DE VIDA DE SISTEMAS

“Por lo general, siempre ha existido controversia sobre los pasos que se requieren para desarrollar aplicaciones basadas en computadoras, sin importar la metodología que se utilice, cualquier modelo generalizado para resolver problemas debe incluir las siguientes actividades:

- Definir el problema y sus requerimientos
- Analizar el proceso y soluciones alternativas
- Diseñar la mejor alternativa
- Construir el sistema
- Probar la operación del sistema y los resultados
- Implantar la solución
- Mantener el sistema operando adecuadamente”<sup>4</sup>

Tomando en cuenta este marco, la construcción de sistemas por métodos diferentes, el método tradicional y el método CASE, puede ser ilustrada comparando los modelos de “caída de agua” (waterfall) de ambos enfoques. En estos enfoques, el trabajo fluye de una etapa a otra y la subsecuente etapa recibe la salida exacta de la etapa precedente.

En el ciclo de desarrollo de sistemas tradicional, la parte teórica se señala en la parte superior de la figura 2.1. La realidad se muestra en la parte baja de la lámina. En cada etapa se realizan cambios, se encuentran errores, se tienen nuevos requerimientos, y se pierde mucha energía cuando todo el trabajo realizado es llevado a etapas superiores para reiniciar el flujo. El sistema se vuelve un verdadero remolino mientras el trabajo se cicla y se recicla. Y para continuar más adelante con la “caída de agua”, la turbulencia y la confusión se incrementan y el status de cualquier elemento particular del sistema siempre es incierto.

## WATER FALL. "CICLO TRADICIONAL"

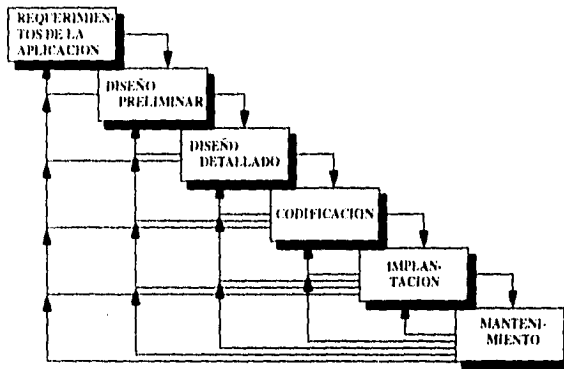


FIGURA 2.1

La situación es totalmente diferente en la "caída de agua" de CASE. Se maneja el mismo ciclo de desarrollo de sistemas, pero el flujo del trabajo cambia radicalmente y se vuelve más tranquilo, con la ventaja de que se gasta una cantidad mínima de energía en términos de recursos. En la "caída de agua" de CASE no existen cambios, parches, ni modificaciones que no encajen con los requerimientos. El proceso fluye casi automáticamente. Cuando algún cambio sea necesario en cualquier etapa de la "caída de agua", el trabajo es llevado hasta el inicio, a la definición de los requerimientos, permitiéndosele que fluya nuevamente a través de todo el proceso prácticamente de manera automática, resultando un flujo constante de trabajo. En la figura 2.2 se muestra un diagrama con la caída de agua de CASE.

## WATER FALL "CASE"

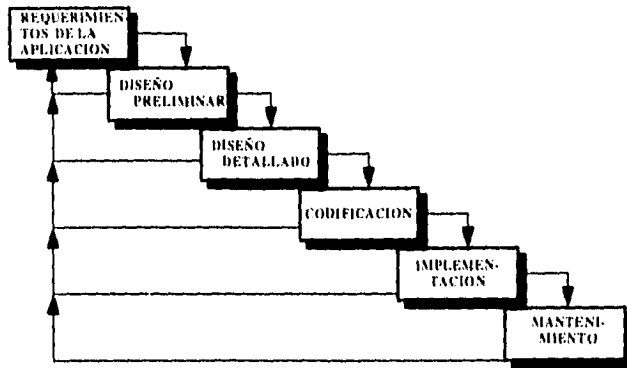


FIGURA 2.2

Si comparamos ambos métodos basándonos en lo anterior tenemos que, en el ciclo de desarrollo de Sistemas Tradicional un mínimo de esfuerzo es puesto en la definición de requerimientos y en el análisis; y la mayor parte se invierte en la construcción, pruebas e implantación. El resultado, invariablemente, es que el trabajo nunca es finalizado; los proyectos terminados son reciclados para que sean nuevamente analizados y desarrollados desde su fase inicial, como si fueran un nuevo sistema. El mantenimiento es desarrollado indefinidamente, y entre más tiempo pasa se vuelve más complejo y complicado, perdiéndose el concepto original del sistema. Todo esto colabora en gran proporción a la Crisis del Software.

Los nuevos métodos de desarrollo CASE tienen una marcada elevación en las fases iniciales, cuando conjuntamente trabajan los usuarios y el personal de desarrollo. Esto tiene una razón de ser, ya que si más tiempo es invertido en la definición de requerimientos, en discusiones con el usuario, al final estos usuarios estarán más satisfechos con el producto final. Similarmente, si se invierte bastante tiempo durante el análisis y el diseño para expresar los detalles de la aplicación en una forma estandarizada, estos detalles pueden ser automáticamente tomados por sistemas CASE y transformados en una serie de pasos hasta el producto final deseado. Aplicando un mayor esfuerzo y utilizando herramientas CASE automatizadas en las fases iniciales, la

aplicación será finalizada en la forma deseada en menos tiempo y con menos esfuerzo requerido para el posterior mantenimiento. Los viejos métodos de desarrollo usualmente tenían prisa por comenzar a producir cualquier resultado lo más pronto posible, en cambio CASE utiliza métodos de ingeniería ya probados, de esta forma el proyecto es finalizado tal y como fué planeado aumentando de esta forma la productividad.

### COMPARACION DE METODOS DE DESARROLLO

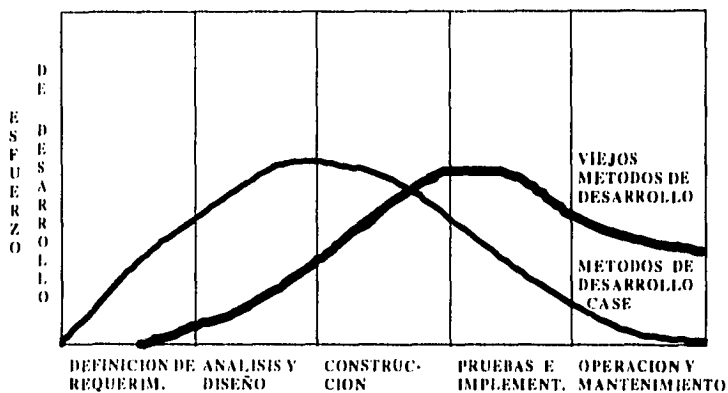


FIGURA 2.3

La figura 2.3 hace una comparación entre el desarrollo tradicional y la automatización del software, mostrando los cambios existentes entre ambos métodos.

Los incrementos en la productividad con CASE vienen a ser:

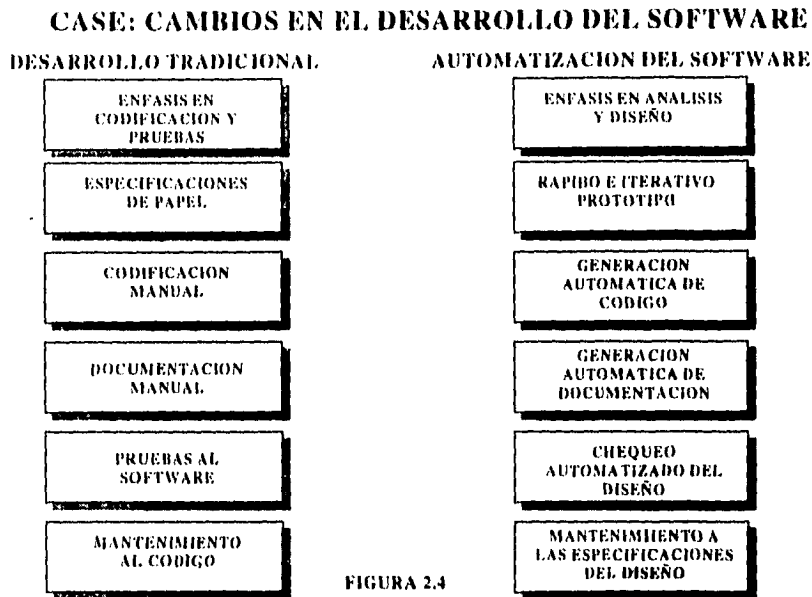
- Alcanzar lo más posible los requerimientos de los usuarios, a partir de la transformación automatizada de las especificaciones hasta llegar al código
- Una mejor documentación
- Una modularización del sistema
- Un mantenimiento más sencillo

Uno de los objetivos de CASE es construir sistemas de tal forma que el costo de mantenimiento pueda tenerse bajo control, no permitiendo que exceda los límites presupuestados en las fases iniciales del desarrollo. Si el trabajo de mantenimiento puede ser reducido considerablemente, uno de los mayores problemas que causan la crisis del software estará siendo atacado efectivamente.

Por lo que se mencionó anteriormente es notorio que el ciclo tradicional de desarrollo de sistemas probablemente no encaje de manera exacta con la metodología CASE, esto debido a que existen dos diferencias primordiales que deberán ser tomadas en cuenta en el momento del desarrollo de sistemas:

- Las transformaciones son automáticas entre las fases y fácilmente pueden ser repetidas automáticamente con modificaciones.
- Existe un repositorio de la aplicación que mantiene el modelo de información y todos los datos que son necesarios para realizar un retroceso, hacer modificaciones o para documentar el proceso.

La figura 2.4 hace una comparación entre el desarrollo tradicional y la automatización del software, mostrando los cambios existentes entre ambos métodos.



Estas diferencias se deben, principalmente a que el ciclo de vida tradicional fué diseñado para métodos de desarrollo de sistema manuales, sin que existiera un flujo de información y de procesos automatizados entre una etapa y otra. Por lo mismo, esta orientado a las tácticas y especificaciones de cada fase en específico, y no ve el desarrollo en un plan general; en cambio, el objetivo de CASE no es obtener productos aislados que mejoren el desarrollo de sistemas para una etapa dada y hasta un punto determinado sin ninguna interfase o conexión con las demás etapas. "La finalidad es obtener un conjunto de productos CASE que mejoren:

- La productividad de requerimientos, definición y diseño; llamado "Upper CASE"
- La productividad de la codificación y pruebas; llamado "Lower CASE"
- La productividad en la modificación de las aplicaciones que están siendo desarrolladas en cualquier etapa de su desarrollo
- La utilización eficiente de técnicas de calidad asegurada a través del desarrollo del ciclo de vida

- La productividad en la integración de aplicaciones, a través del uso de datos compartidos, módulos de programación y etapas de desarrollo
- La productividad en el mantenimiento de las aplicaciones reteniendo en la base de datos toda la información usada en el desarrollo del sistema
- La calidad y productividad de muchas aplicaciones operacionales viejas, utilizando en ellas ingeniería reversible y reconstruyendo los sistemas usando ahora herramientas CASE”<sup>5</sup>

Tales ventajas en la productividad solamente serán obtenidas a través de la experiencia y del uso de diferentes herramientas CASE, las cuales deberán ser administradas y manejadas apropiadamente.

## 2.3 HERRAMIENTAS CASE

### 2.3.1 DEFINICION Y EVOLUCION

Las herramientas CASE se consideran la columna vertebral de esta filosofía, debido a que se enfocan en la productividad individual del analista.

Las herramientas están diseñadas para estaciones de trabajo, son de propósito general ya que pretenden utilizarse en el desarrollo de todo de tipo de sistemas, soportan un entorno personal, utilizan gráficos para especificar y documentar los sistemas, reúnen todas las fases del ciclo de vida del software, utilizan la inteligencia artificial para realizar automáticamente algunas tareas de desarrollo y mantenimiento del software.

Existen herramientas CASE para (ver figura 2.5):

**Diagramación:** Son herramientas para dibujar diagramas estructurados y especificaciones gráficas.

**Generadores de pantallas y de informes:** Estas herramientas se utilizan para hacer prototipos para el usuario y para crear especificaciones del sistema.



**Repositorio e informes:** Son herramientas para almacenar, obtener y consultar la información del sistema.

**Verificación y análisis:** Estas herramientas sirven para verificar la consistencia y para corregir las especificaciones del sistema.

**Generadores de código:** Se utilizan para generar código ejecutable partiendo de las especificaciones del sistema.

**Mantenimiento:** Son herramientas para redocumentar, reestructurar y analizar el sistema actual.

### LAS HERRAMIENTAS CASE CUBREN VARIOS TIPOS DE DESARROLLO DE SOFTWARE Y MANTENIMIENTO

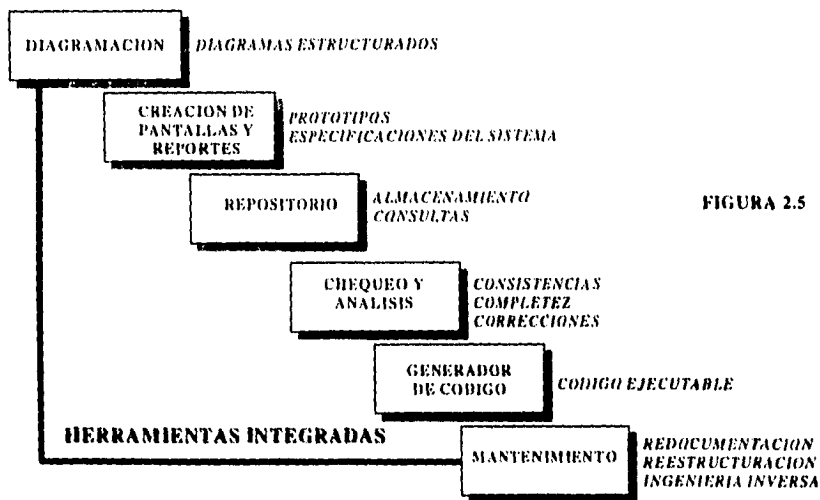


FIGURA 2.5

#### *Evolución de las Herramientas CASE.* (Ver figura 2.6)

Los inicios de CASE se remontan a la década de los 80's con la introducción de la documentación asistida por computadora y las herramientas para diagramación. Su objetivo era producir automáticamente la documentación requerida por varias metodologías estructuradas de desarrollo (Yourdon, Jackson y Martin). Unas cuantas compañías producían un número de

herramientas CASE que manejaban partes específicas del ciclo de vida del desarrollo del software.

A mediados de la misma década, las herramientas CASE fueron mejoradas para cumplir fundamentalmente con dos funciones:

- La verificación automática de diagramas estructurados
- El almacenamiento de diagramas estructurados en librerías diseñadas automáticamente, llámense diccionarios, repositorios y/o enciclopedias.

A finales de los 80's, se llevó a cabo la integración de algunas herramientas del desarrollo de software. Algunas herramientas se expandieron a diferentes fases del ciclo de vida del desarrollo. Otras han sido ligadas bajo sistemas operativos estándares de tal forma que la información puede pasarse de una etapa a otra. En esta fase se logró la generación automática del código a partir de las especificaciones del diseño y la unión entre el diseño automático y la programación automática.

A principios de los 90's las herramientas del desarrollo del software no solo son ligadas a través de todo el ciclo de vida, sino que también:

- Estructuran a los sistemas metodológicamente para la administración del proyecto
- Permiten la calendarización de las tareas a través del proyecto
- Brindan entrenamiento basado en computadora
- Son apoyados por una red de área local
- Se incorpora como característica la reutilización del software como metodología de desarrollo; en lugar de construir un nuevo software los analistas pueden reusar componentes de software almacenados en los repositorios CASE
- Desarrollan una interfase amigable con el usuario

### EVOLUCION DE LA TECNOLOGIA CASE

INICIO 80's	MEDIADOS 80's	FINALES 80's	INICIO 90's
DOCUMENTACION ASISTIDA POR COMPUTADORA	ANALISIS, DISEÑO Y CHEQUEO AUTOMATIZADO	GENERACION AUTOMATICA DE CODIGO A PARTIR DE LAS ESPECIFICACIONES DE DISEÑO	REUSO DE CODIGO COMO METODOLOGIA DE DESARROLLO
DIAGRAMACION	REPOSITDRIO	INTERFASES ENTRE LA AUTOMATIZACION DEL DISEÑO Y LA AUTOMATIZACION DE LA PROGRAMACION	METODOLOGIA INTELIGENTE DE "NAVEGACION"
HERRAMIENTAS DE ANALISIS Y DISEÑO		"REPOSITARIOS" DE DATOS	ENCAPSULAMIENTO DE PROCESOS Y DATOS (OBJETOS)

FIGURA 2.6

### 2.3.2 CLASIFICACION DE LOS PRODUCTOS CASE

CASE puede ser clasificado por función o por su uso en el ciclo de vida del desarrollo de sistemas.

#### 2.3.2.1 CLASIFICACION POR FUNCION

Una forma útil de clasificar los productos CASE es por el área funcional en la cual son empleados. Dentro de este enfoque CASE es utilizado en las siguientes tareas: asistencia en la administración (manejo) y cumplimiento (realización) del proceso CASE; análisis en las áreas que son difícilmente manejadas manualmente; y automatización de las tareas manuales.

La tarea de asistencia incluye a cualquier herramienta CASE que acumula y realiza un seguimiento de la información, o hace que la información esté disponible de una forma organizada y lista para manejarse por otros sistemas o manualmente. Estas herramientas, asisten a los administradores de CASE en acumular la información que será valiosa para CASE en la búsqueda del incremento en la eficiencia y efectividad, mientras reduce el esfuerzo del

personal y el rango de error. Son herramientas que controlan los procesos de desarrollo de sistemas o auxilian en el manejo de los datos que están siendo utilizados.

Algunas de las herramientas CASE de asistencia son:

- Herramientas de supervisión, tales como planeación, control de proyectos, administración de proyectos, y aseguramiento de calidad
- Herramientas de manejo de diccionario, tales como:
  - \* Almacenamiento y recuperación de información del repositorio
  - \* Editores para captura de información, incluyendo diagramas, formas, matrices y texto
  - \* Producción de tablas de información y referencias cruzadas
- Herramientas de diseño de pantallas y reportes, que están ligadas al repositorio
- Herramientas de dibujo para crear, modificar, y producir diagramas, y reportes visuales que puedan proveer una entrada en la base de datos

La tarea de análisis incluye a las herramientas CASE que auxilian en la realización de este trabajo que de otra forma resultaría sumamente complejo y tomaría demasiado tiempo el desarrollarlo con métodos manuales. Tales herramientas aumentan las capacidades del personal de desarrollo de aplicaciones basadas en CASE. En general, las herramientas de análisis aseguran un alto nivel en la calidad de los sistemas.

Algunas de las herramientas de análisis CASE resultan útiles para:

- Proporcionar verificaciones de diseño, tales como revisiones de consistencia y un balanceo nivel a nivel del conjunto de diagramas de flujo
- Proveer referencias cruzadas de la aplicación
- Producir reportes complejos confiables
- Aplicar reglas del negocio conocidas para el diseño
- Analizar posibles inconsistencias y redundancias en el diseño de datos.
- Forzar la creación de reglas de tipo administrativo
- Rastrear los componentes durante el desarrollo del sistema, incluyendo el código, las especificaciones y el diseño de módulos

La tarea de automatización incluye herramientas CASE que construyen un puente entre las fases del ciclo de vida. Las herramientas proveen matrices de información que han sido elaboradas en una fase para su uso en otra. El sistema CASE automáticamente construye un puente entre las fases de desarrollo, eliminando algunas actividades manuales y haciendo otras mucho más seguras.

Algunas de las técnicas de automatización CASE se encargan de:

- La generación de programas, hasta un sistema operacional completo, desde el diseño de información, produciendo código, pantallas, reportes, etc.
- La generación de información estándar desde un medio ambiente tipo Cobol
- La producción de código con una gran consistencia y un rango de error sumamente reducido
- La creación de puentes entre las diferentes fases del ciclo de vida y entre diferentes técnicas y herramientas CASE
- Normalización de la Base de Datos

El sistema CASE ideal moverá la información desde los requerimientos finales del usuario hasta un sistema ejecutable de forma automática. Toda la información durante el proceso estará concentrada en el repositorio, asegurando que la documentación se obtendrá automáticamente.

### ***2.3.2.2 CLASIFICACION POR SU USO EN EL CICLO DE VIDA DE DESARROLLO***

Otra forma común y útil de categorizar o clasificar los productos CASE es por las fases que abarcan en el ciclo de vida de sistemas en donde son empleados.

La siguiente figura (2.7) muestra las etapas en las cuales las diferentes herramientas CASE pueden ser ubicadas. Cualquier herramienta CASE puede ser empleada en una o más de una fase, o solamente en una tarea específica de ella, a elección del analista.

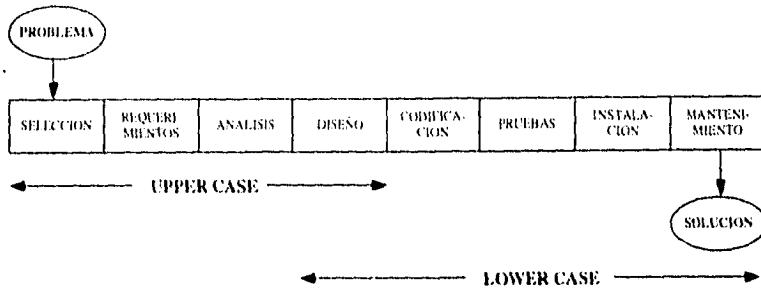


FIGURA 2.7

Las herramientas tipo Upper CASE son utilizadas en las actividades iniciales del desarrollo del software. Esto significa que involucran tanto al usuario final como a los analistas que juegan un papel crítico para que el desarrollo del sistema se inicie correctamente. El factor más importante de las herramientas Upper CASE es su capacidad para transportar, o transformar, datos entre diferentes herramientas CASE. Muchas de estas herramientas son genéricas y pueden ser usadas con una gran variedad de herramientas CASE. Es fundamental que cualquier herramienta Upper CASE que se haya seleccionado tenga comunicación libre y directa con el repositorio central de información, o la base de datos del diseño del proyecto.

Algunas de las herramientas Upper CASE ayudan en:

- La selección y planeación del sistema, incluyendo modelo empresarial, modelos relacionales, planeación del sistema, cálculos de probabilidades y presentación de gráficas
- La definición de requerimientos del sistema, incluyendo una especificación gráfica, utilerfas de pantallas, diagramas de flujos de datos, y procesador de palabras integrado
- El análisis del sistema, incluyendo análisis estructurado, modelo de entidad-relación, diagrama de modelo de datos, diccionario integrado y diseño de reportes

- El diseño de sistema, incluyendo diseño estructurado, diseño orientado a objetos, diagramas estructurados, diagramas de modelo de datos, descripción de procesos, descripción modular, descripción de elementos y diagramas de entidad-relación

Las herramientas tipo Lower CASE se utilizan en las últimas etapas del ciclo de desarrollo del software. Están enfocadas a los desarrolladores de sistemas y tienen el objeto de tomar las salidas resultantes de las etapas anteriores y dar como salida una solución final, la cual este totalmente probada y que requiera en un futuro un mínimo de mantenimiento. Este tipo de herramientas se auxilian de lenguajes de cuarta generación para el mejor desempeño de sus tareas.

Algunas de las herramientas Lower CASE son:

- Herramientas de diseño de sistemas usadas en sesiones de trabajo que producen código
- Herramientas de codificación del sistema, incluyendo generación de código fuente, librerías de copiado y el uso de código reutilizable
- Herramientas de prueba del sistema, incluyendo diagramas de flujo de datos, diagramas estructurados, diseño de pantallas, y diseño de reportes
- Herramientas de implantación de sistemas, incluye documentación automática, procesador de palabras integrado y presentaciones gráficas
- Herramientas de mantenimiento de sistemas, incluyendo reportes operacionales, análisis de error, etc.

### 2.3.3 CATEGORIAS DE LAS HERRAMIENTAS CASE

Las herramientas se enfocan en el soporte de diferentes fases del ciclo de vida del software en el desarrollo de sistemas.

Para distinguir las herramientas CASE, se han definido básicamente tres categorías:

**CASE TOOLKIT:** Representan un conjunto de herramientas CASE integradas que han sido diseñadas para trabajar juntas y automatizar parcial o totalmente una fase del ciclo de vida de software.

**CASE WORKBENCH:** Son un conjunto de herramientas CASE integradas que han sido diseñadas para trabajar juntas y automatizar o proveer ayuda sistematizada a todo el ciclo de vida del software, incluyendo: análisis, diseño, codificación, implantación, pruebas y documentación.

Un CASE WORKBENCH provee de soporte funcional completo para el desarrollo de software, de tal modo que:

- Crea requerimientos gráficos del sistema y diseña especificaciones
- Revisa, analiza y hace referencias cruzadas de la información del sistema
- Almacena, administra y hace reportes de la información del sistema
- Construye prototipos del sistema y sistemas de simulación
- Genera códigos del sistema acompañados de su respectiva documentación
- Prueba, valida y analiza programas

Para lo cual debe poseer las siguientes características:

- Capacidad gráfica
- Verificación de errores
- Repositorio de información
- Conjunto de herramientas integradas
- Cobertura de todo el ciclo de vida
- Soporte para prototipos
- Generación automática de código
- Soporte de metodología estructurada

**CASE METHODOLOGY COMPANION:** Son un conjunto de herramientas CASE que automatiza tareas en una metodología CASE particular y/o automatiza la producción de documentación y otras necesidades requeridas por la metodología; puede ser:

- CASE Methodology Companion Toolkit
- CASE Methodology Companion Workbench



## 2.4 REPOSITORIO, REINGENIERIA Y RELACIONAL

La filosofía CASE, engloba tres elementos principales:

- El repositorio, donde se encuentra toda la definición del sistema
- La reingeniería, que permite regresar y modificar la etapa de análisis de un sistema ya existente
- Relacional que se enfoca a la utilización de bases de datos con estructuras relacionales

### 2.4.1 REPOSITORIO CENTRAL DE INFORMACION

El repositorio central de información también llamado enciclopedia, se considera como el elemento crítico y esencial de cualquier conjunto de herramientas integradas CASE, ya sean *toolkits* o *workbenches*. "Es el mecanismo que sirve para definir, almacenar y administrar la información así como los objetos requeridos para desarrollar, mantener y ejecutar sistemas de una corporación, constituyendo un medio a partir del cual es posible la integración, estandarización y documentación de las herramientas CASE"<sup>6</sup>, ya que mantiene la información suficiente para el diseño del sistema y sus procedimientos, de tal forma que el código pueda ser generado automáticamente e inclusive sea posible su reutilización.

Se le considera como la llave para el incremento de la productividad, debido a que provee a los analistas de información consistente y controlada en el momento en que la necesite y en el formato que les resulte útil.

Otros de los beneficios que proporciona el compartir la información del sistema, es que se aumenta la comunicación con los usuarios, se incrementa la integridad del sistema debido a que los datos son capturados una sola vez y almacenados en un lugar lo que permite simplificar el mantenimiento, las migraciones y conversiones de los sistemas.

Como se puede ver, el repositorio, es el punto central de las aplicaciones desarrolladas en CASE, debido a que no solamente provee el control

administrativo que se necesita sobre los datos, sino que los documenta en diagramas, facilitando su uso y comprensión de las estructuras de datos en un desarrollo de aplicaciones rápido, recolectando y seleccionando toda la información que es necesaria en el proceso de desarrollo de la aplicación, brindando métodos automáticos para convertir la información en formas útiles y necesarias.

#### **2.4.1.1 CONTENIDO DEL REPOSITORIO**

La información que se considera relevante para incluirse dentro del repositorio de información CASE abarca:

##### ***Información de la Organización:***

- Estructura Organizacional
- Modelo del Negocio
- Reglas del Negocio

##### ***Información del Sistema en nivel lógico y físico***

###### **Datos:**

Elementos de datos  
Registros de datos  
Almacenamiento de datos  
Modelo de datos

###### **Procesos:**

Procesos  
Funciones  
Entidades Externas  
Modelos

**Gráficas:**

- Diagramas de flujo de datos
- Diagramas de organización
- Diagramas de Modelo de datos
- Diagramas de Entidad-Relación
- Pantallas y reportes

**Relaciones**

***Información de Administración***

- Propiedad
- Auditoría
- Seguridad
- *Staffing*
- Reglas del método
- Modelo del ciclo de vida
- Modelo de información del repositorio

**2.4.2 REINGENIERIA**

Es el proceso de modificar las definiciones de datos y/o procesos lógicos de un sistema (o programa) ya existente, con la ayuda de herramientas automáticas para:

1. Mejorar su futuro mantenimiento
2. Actualizar su tecnología
3. Extender sus expectativas de vida
4. Capturar sus componentes en un repositorio
5. Incrementar la productividad de los encargados del mantenimiento

***Objetivos***

- Mejor manejo de cartera de sistemas existentes
- Proveer asistencia automática para mantenimiento
- Reducir errores y costos de mantenimiento
- Incrementar la productividad de los encargados del mantenimiento de sistemas

- Incrementar el intercambio de STAFF de mantenimiento
- Elaborar sistemas fáciles de comprender, modificar y probar
- Hacer posible la conversión y migración de sistemas
- Apegarse a los estándares
- Mejorar la respuesta de solicitud de mantenimiento
- Mejorar la moral del equipo de mantenimiento
- Proteger y alargar la vida de los sistemas
- Usar herramientas CASE para soportar sistemas existentes
- Reusar componentes de sistemas existentes

#### *Mantenimiento*

Es el proceso de conservación de SOFTWARE “corriendo” o mejorando sus características.

#### *Razones para realizar Reingeniería:*

- Fallas frecuentes en la producción
- Problemas de desempeño
- Tecnología desfasada
- Problemas de integración en sistemas
- Mala calidad de código
- Dificultad o riesgo de peligro al cambiar
- Dificultad al probar
- Sistemas caros de mantener
- Incremento de problemas

### **2.4.3 RELACIONAL**

La filosofía CASE, tiene también su fundamento en el Modelo de Datos Relacional. Debe ser relacional porque es precisamente este modelo el que nos permite tener un diseño adecuado de la distribución de los datos en nuestras estructuras, para con ello evitar redundancia, inconsistencia, rigidez, etc. y favorecer al buen desempeño de los sistemas.

Para que exista una buena definición de datos, deben seguirse ciertas directrices, las cuales servirán de soporte al sistema. Es necesario tomar en

cuenta las propuestas de Chris Date, para fundamentar la definición de los datos.

## **2.5 CONSIDERACIONES PARA LA IMPLANTACION DE CASE**

Tomar una decisión sobre la utilización de la filosofía CASE en el desarrollo de sistemas no es fácil y menos lo es aún si tomamos en cuenta que la experiencia ha demostrado que su uso no siempre garantiza el éxito del proyecto, debido a que existen consideraciones obvias que algunas empresas, por lo mismo, suelen pasar por alto. En primer lugar tenemos simplemente que la filosofía CASE no puede mejorar la productividad y calidad del software si no se usa, no se debe ver a CASE como una panacea, es necesario además de adquirirla, que los analista sean entrenados adecuadamente y se encaminen para utilizar el enfoque de ingeniería mientras construyen sistemas, y así mismo que se involucren con las metodologías de desarrollo. Su actitud es tan importante que puede afectar considerablemente ya sea en el éxito o fracaso del uso de la filosofía CASE.

En este momento es oportuno aclarar que la filosofía CASE no substituye a una buena administración del proyecto. Muchos de los proyectos de software se salen de control porque la administración no se preocupa por establecer estándares de desarrollo ni de procesos, y peor aún, ni siquiera define una estrategia donde se indique como se va a implantar CASE en la empresa.

Por estas consideraciones y otras más, CASE se ve como una filosofía de alto riesgo y cuando una organización la conoce puede optar por tomar dos posiciones:

- Simplemente no hacer nada
- Adoptar la nueva tecnología

La decisión no es sencilla, sin embargo debe estar fundada no únicamente como respuesta a la presión del usuario para construir sistemas de manera más rápida, o en el deseo del analista por ponerse en contacto con los avances tecnológicos más avanzados.

La gráfica que se muestra en la figura 2.8 menciona la evolución del cambio que debe darse en la organización con el objetivo de lograr la institucionalización de CASE en la misma. Es necesario mencionar que no siempre se logra éxito en la implantación de CASE, debido a la resistencia al cambio por parte de la gente involucrada.

### EVOLUCION DEL CAMBIO

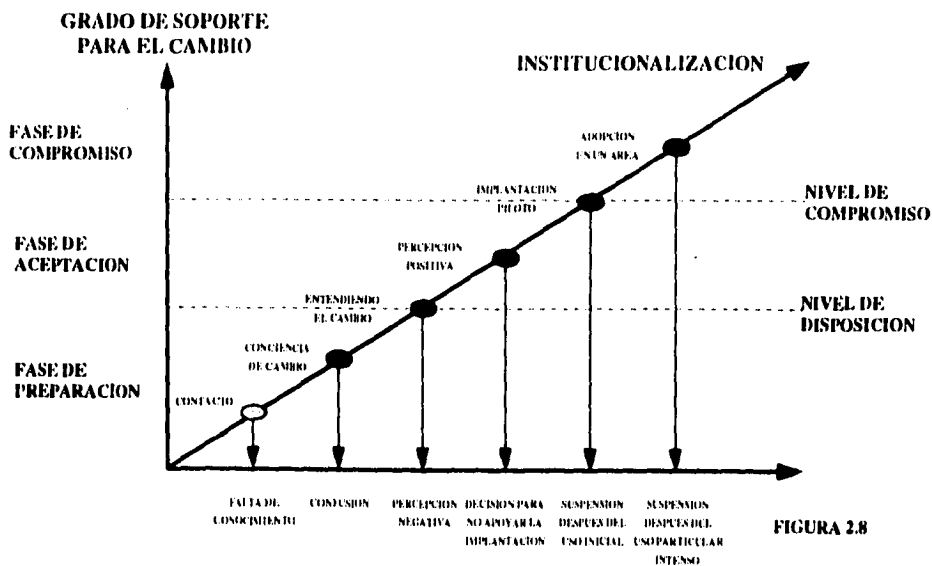


FIGURA 2.8

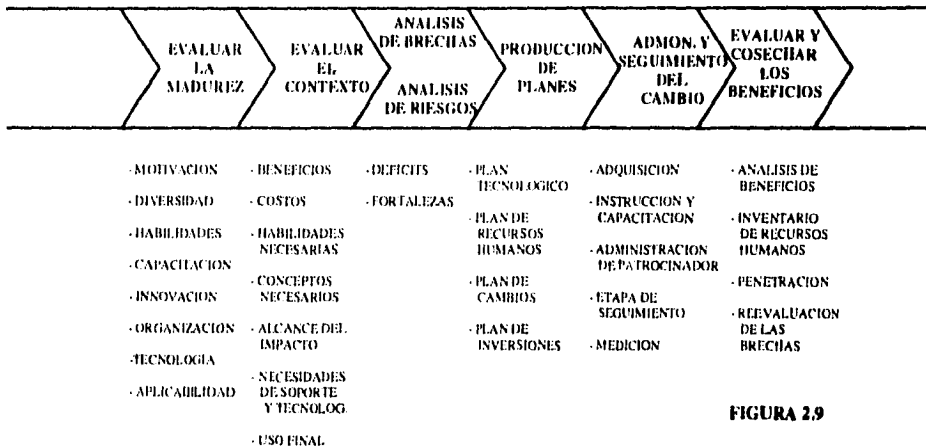
La indecisión es causada en gran parte por el riesgo de fracaso en el que se puede incurrir por una toma de decisiones errónea.

En la figura 2.9 se muestra una estrategia que se puede utilizar para implantar CASE.

Una empresa puede tomar una decisión razonable sobre el uso de la tecnología CASE una vez que haya definido una estrategia donde se :

- Definan las necesidades de la organización
- Demuestre que la nueva tecnología puede satisfacer esas necesidades
- Desarrolle un plan para implantar la nueva tecnología

**ESTRATEGIA PARA LA IMPLANTACION DE CASE**



**FIGURA 2.9**

**UNA IMPLANTACION DE CASE CON EXITO ES UNA FUNCION DE EXPECTATIVAS, CONTEXTO, VISION, TECNOLOGIA, MEDICION Y ADMINISTRACION DEL CAMBIO.**

**2.5.1 PUNTOS BASICOS QUE SE SUGIEREN CUBRIR PARA IMPLANTAR CASE**

Cuando una empresa decide mejorar la calidad de su software, y decide implantar alguna herramienta CASE, debe tomar en cuenta ciertas consideraciones, entre las que citaremos algunas de la más importantes:

- “Definir el proceso del ciclo de vida de sistemas de software especificando sus pasos, los estándares y procedimientos que los controlan y la calidad asegurada para el nuevo producto.

- Definir el orden de aplicación de la tecnología CASE al proceso basado en las necesidades y prioridades de la organización.
- Seleccionar las herramientas CASE para automatizar y soportar el proceso y apoyar los estándares y procedimientos.
- Utilizar herramientas y metodologías CASE en proyectos piloto.
- Introducir las herramientas y metodologías CASE probadas en gran escala a través de la organización en programas de entrenamiento, reuniones administrativas y de reforzamiento
- Evaluar el impacto de CASE aplicando medidas predeterminadas de éxito y manteniendo un comité para mejoras continuas del proceso de ciclo de vida de software y calidad del producto de software. “<sup>7</sup>”

### **2.5.2 CONSIDERACIONES TECNICAS PARA SELECCIONAR HERRAMIENTAS CASE**

En este apartado no se pretende dar una guía de compra, únicamente se mencionan los puntos en lo que se debe hacer énfasis en el momento de seleccionar alguna herramienta, criterios sugeridos por Carma Mc. Clure

#### **“Requerimientos de Hardware**

- Nivel Workstation
  - Memoria y capacidad de procesamiento
  - Calidad en gráficos
  - Capacidad para conectarse en red
  - Capacidad de multitasking
  - Conectividad
- Para dos o tres plataformas atadas
  - Conectividad entre ambas
  - Soporte multiusuario
  - Interfases para incorporar diccionarios y bases de datos
  - Bridges para 4GL
  - Ligar múltiples repositorios de sistemas de información
  - Rapidez y eficiencia
  - Seguridad y control
  - Back ups
  - Acceso concurrente al disco



- Las herramientas deben incluir: Diccionarios, DBMS y 4GLs con herramientas CASE debidamente integradas
- Tipos de sistemas soportados por las herramientas
- Herramientas de calidad de documentación
- Herramientas de soporte y mantenimiento
- Entrevistas con los actuales usuarios de las herramientas
- Reputación del vendedor y de la línea de producto
- Calidad de las capacidades técnicas de la herramienta como son:
  1. Interfase gráfica
  2. Nivel de integración de la herramienta
  3. Interfases con herramientas externas
  4. Repositorio CASE
  5. Soporte de cobertura del ciclo de vida
  6. Soporte de metodología estructurada “.”<sup>8</sup>

---

<sup>1</sup> James Martin, Structured Techniques: The Basis for CASE. Edit. Prentice Hall, 1988.

<sup>2</sup> Qed, Information Sciences Inc. CASE the Potencial and the Pitfalls. Pág 1, 1989

<sup>3</sup> Qed, Information Sciences Inc. CASE the Potencial and the Pitfalls. Pág 13 , 1989

<sup>4</sup> Qed, Information Sciences Inc. CASE the Potencial and the Pitfalls. Pág 119, 1989

<sup>5</sup> Qed, Information Sciences Inc. CASE the Potencial and the Pitfalls. Pág 57, 58, 1989

<sup>6</sup> Carma McClure, CASE is the Software Automation, Pág. 53 , 1989

<sup>7</sup> Carma McClure, CASE is the Software Automation, Pág. 178, 1989

<sup>8</sup> Carma McClure, CASE is the Software Automation, Pág. 180, 1989

## **CAPITULO III: AMBIENTE CASE PARA EL DESARROLLO DEL SISTEMA**

En este capítulo se hace una revisión sobre el marco de trabajo y filosofía CASE que se utiliza para desarrollar el Sistema de Control de Atraques y Servicios de La Marina.

Elegimos la estrategia formal de Unisys Corporation para la obtención de soluciones ASD (Advanced Solution Development), definida como una estructura abierta que integra herramientas de desarrollo avanzado de soluciones, métodos y servicios.

El objetivo principal de ASD es relacionar las tecnologías con las metas del negocio, lo que permite asegurar que la estrategia de sistemas soporta la estrategia del negocio. ASD se constituye por tres partes fundamentales:

- “Principios: Sus elementos son el sentido común, y la especificación del cliente
- Plataforma de Integración: Ambiente de desarrollo integrado para la construcción de soluciones basadas en los principios de ASD
- Herramientas y Servicios: Son un conjunto específico de tecnologías de software y servicios de consultoría para soluciones rápidas y efectivas que compaginen con las necesidades del negocio”<sup>1</sup>

LSA (LINC Systems Approach) y LINC (Language Interface Network Compiler) son productos de Unisys, LSA es la metodología escrita específicamente para soportar el uso de herramientas Upper y Lower CASE LINC, y LINC es un ambiente de cuarta generación para el desarrollo de sistemas.

La filosofía que maneja LINC, otorga especial importancia a los conceptos, necesidades y objetivos del negocio que dirigen tanto la metodología como la tecnología para obtener los beneficios de una toma de decisiones efectiva, flexibilidad y eficiencia operativa. Según esta filosofía el desarrollo de sistemas de información debe basarse en dos conceptos:

- “Diseño dirigido al negocio: Los sistemas de información deben ser un modelo del negocio
- Desarrollo Evolutivo: Los cambios a los sistemas de información deben responder a los cambios que sufre la empresa, manteniéndose siempre la integridad del sistema”<sup>2</sup>

LINC abarca una filosofía, metodología y tecnología para el desarrollo de sistemas de información utilizando productos CASE/4GL. LINC se usa principalmente para desarrollar sistemas de información para negocios aplicando la metodología LSA.

LSA y LINC soportan los principios, plataforma de integración, herramientas y servicios de ASD.

### **3.1 ENFOQUE DE SISTEMAS LINC (LSA, LINC Systems Approach)**

LSA es una metodología escrita específicamente para soportar la filosofía LINC y asistir a los analistas con el objetivo de aprovechar al máximo las ventajas que proporcionan las herramientas Upper y Lower CASE de LINC.

Aunque se pueden ocupar otras metodologías en el desarrollo de sistemas de información LINC, LSA es la preferida, ya que brinda procesos controlados y flexibles que guían desde la investigación del sistema hasta la solución del problema, considerando las metas del negocio, la figura 3.1 muestra una comparación entre el desarrollo tradicional de sistemas y el de LSA.

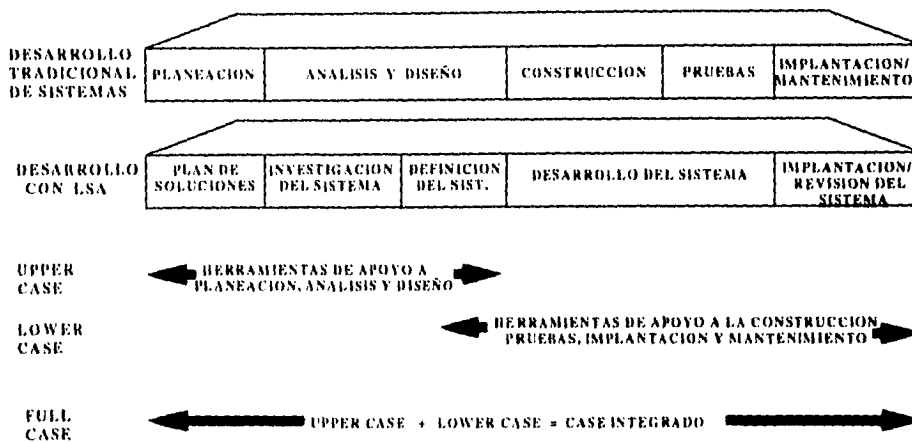


FIGURA 3.1

LSA implementa desarrollo de sistemas por equipos de proyecto pequeños, de tres a cinco analistas junto con dos o tres usuarios. Esto provee la flexibilidad que se necesita para manejar el cambio y la realización de parte del trabajo por los desarrolladores. Tales beneficios son:

- Eliminación de barreras de comunicación
- Incremento en la productividad y calidad
- Uso completo del poder de LINC

El ciclo de vida que sigue esta metodología consta de cinco pasos iterativos:

- Investigación del Sistema
- Definición del Sistema
- Desarrollo del Sistema
- Implantación del Sistema
- Revisión del Sistema

### **3.1.1 INVESTIGACION DEL SISTEMA**

Corresponde a la primera etapa dentro del ciclo de vida tradicional de sistemas (análisis) el cual se mencionó en el capítulo dos. Pone especial énfasis en las actividades y áreas funcionales alrededor de un segmento específico del negocio, con el objeto de entender la forma actual de hacer las cosas dentro del negocio, de este modo se investigan las actividades tanto manuales como automatizadas.

Antes de entrar a esta etapa se recomienda elaborar un Plan de Soluciones de Información. Este plan debe enfocarse al negocio como un todo y definir una estrategia para el uso de tecnología de información y de desarrollo de sistemas que soporten los objetivos estratégicos de la empresa.

Los resultados del plan se usan para identificar el alcance, objetivos y restricciones del nuevo sistema de información, así como los recursos requeridos para el diseño y desarrollo del sistema propuesto.

Los objetivos de esta fase son :

- Establecer metas del proyecto, requerimientos, fronteras y restricciones
- Identificar el personal y recursos requeridos para el diseño y desarrollo del sistema
- Definir las actividades y áreas funcionales involucradas en el segmento del negocio
- Hacer una presentación del proyecto

### **3.1.2 DEFINICION DEL SISTEMA**

La segunda fase define al sistema de información como un modelo de objetos LINC, cuyo alcance y objetivos fueron determinados en la etapa anterior.

La definición del sistema comienza creando un modelo conceptual del nuevo sistema basado en las actividades y demás características identificadas en la primera fase, después se definen los atributos técnicos del sistema, tales como: seguridad, respaldos y recuperaciones, coexistencia, interfases gráficas con el usuario, hardware y software.

En esta etapa, se determina el tamaño del sistema y los costos de desarrollo estimados utilizando para ello la técnica de Análisis de Puntos de Función contra el modelo de objetos LINC.

Se diseñan los mecanismos de conversión de sistemas existentes al nuevo sistema como interfases a sistemas externos.

El Asistente de Diseño LINC (LDA, LINC Design Assistant) o cualquier otra herramienta Upper CASE, a través de la Interfase CASE LINC (LCI; LINC CASE Interface) puede utilizarse en esta etapa para modelar el sistema, producir prototipos y documentación para el sistema propuesto.

Los objetivos de esta fase son :

- Definir el modelo de objetos LINC (Características y Contenido Técnico)
- Calcular el tamaño y los costos estimados
- Desarrollar prototipos
- Diseñar procesos de conversión
- Diseñar pruebas de aceptación
- Determinar el impacto del ambiente con el usuario
- Obtener la aprobación del usuario

### **3.1.2.1 MODELO BASADO EN OBJETOS**

LSA incorpora una aplicación práctica de la técnica de modelado para un diseño basado en objetos.

Este modelo permite a los analistas del sistema concentrarse en todos los objetos (en LSA son las actividades del negocio) en lugar de tener que dividir el análisis y diseño de las actividades del negocio en tres conjuntos separados, por ejemplo: datos, procesos, y control de flujos.

LSA describe como construir un modelo del sistema del negocio bajo investigación. A través de la fase de desarrollo de LSA únicamente se trabaja con un modelo, refinándose continuamente conforme se diseña y desarrolla la especificación del sistema LINC.

El modelo del negocio LINC es una representación gráfica de todos los objetos y sus interrelaciones en un sistema de información LINC; cada objeto debe tener datos asociados, reglas del negocio, pantallas, formatos de reportes, relaciones y control de flujo de información.

En términos de LINC un objeto es una especificación integrada que describe un elemento del sistema que se está diseñando.

Esta metodología describe los objetos discretos de la organización en términos de bloques de construcción LINC: Eventos, Componentes, Profiles, Reportes, Lógicas Globales e Inquiries. Conjuntamente, los eventos y los componentes, forman el modelo que representa al sistema del negocio; mientras que los Profiles, Inquiries y Reportes representan el mecanismo para tener acceso a los datos del modelo y generar la información necesaria.

Los Eventos se utilizan para representar las actividades del negocio en el sistema, por lo tanto son las transacciones diarias que lleva a cabo la organización; se lleva un registro en el cual se reflejan los cambios o movimientos que sufren las actividades.

Los Componentes forman la infraestructura que soporta a dichas actividades (Eventos); la información de los Componentes sirve para validar y referenciar a las actividades del negocio, por medio de la llave primaria (Ordinate).

Los Profiles, son mecanismos que proporcionan flexibilidad de acceso a la base de datos del sistema de información. A través de las vistas funcionales se accede a cierta parte de la base de datos que está basada en una función particular que es ejecutada por el usuario. Las vistas también pueden acceder a ciertos datos siempre y cuando se cumpla determinada condición.

Los reportes en LINC, existen por dos razones:

- Para presentar la información que se deriva de los datos registrados en las actividades del negocio
- Para la consolidación de los datos

La Lógica Global es una secuencia, es un tipo especial de Ispec, que es invocado por otros Ispecs o Reportes; se utiliza cuando se requiere estandarizar un procedimiento que es usado por varios Ispecs, para



encabezados o títulos comunes y para evitar validaciones repetidas en una rutina o formato. La Lógica Global proporciona consistencia, congruencia y ahorra tiempo durante el desarrollo del sistema.

### **3.1.3 DESARROLLO DEL SISTEMA**

En esta fase se complementa la especificación del sistema LINC.

Las principales tareas realizadas en esta fase son las siguientes :

- Desarrollo completo del sistema de información
- Desarrollo y prueba del software soporte
- Desarrollo de la seguridad del sistema
- Preparación de toda la documentación
- Pruebas completas y de benchmarking

En realidad, esta etapa se considera como la base para que en la siguiente fase se pueda instalar el sistema en producción.

### **3.1.4 IMPLANTACION DEL SISTEMA**

El objetivo de esta etapa es instalar el sistema probado y aceptado por el usuario en un ambiente LINC Run-Time. Esto involucra entrenar al usuario para la operación del sistema, la instalación del hardware, software y la conversión de datos. Es en este momento cuando se pasa del ambiente de desarrollo al ambiente de mantenimiento.

Los objetivos de esta fase son:

- Instalar Hardware y Software
- Dirigir el entrenamiento
- Instalar el software en producción
- Convertir datos
- Comenzar con la utilización del nuevo sistema

### **3.1.5 REVISION DEL SISTEMA**

Para llevar a cabo esta etapa es necesario que el sistema se haya liberado a producción y esté funcionando por un periodo previamente acordado, es entonces cuando se puede hacer una revisión de sus capacidades, funcionalidad, desempeño, y operación.

El objetivo es medir el éxito del proyecto y prepararse para el siguiente ciclo de desarrollo.

Los propósitos a alcanzar en esta etapa son:

- Identificar oportunidades para mejorar la operación del sistema
- Evaluar y otorgar prioridades al mejoramiento del sistema
- Determinar métricas actuales del sistema
- Identificar la utilización de las ediciones de hardware y software
- Determinar la productividad del desarrollo
- Revisión del proceso de desarrollo

### **3.2 ASISTENTE DE DISEÑO LINC (LDA: LINC Design Assistant)**

LDA es una herramienta Upper CASE para uso específico en el desarrollo de sistemas de información LINC. LDA opera en una computadora personal o en una workstation, y se usa para definir el modelo del negocio LINC, basándose en la metodología LINC Systems Approach (LSA). Cuando dicho modelo se define, se transfiere al sistema de desarrollo LINC II en el HOST para formar el soporte de la especificación del sistema LINC.

LDA en conjunción con el medio ambiente de desarrollo LINC II soporta las siguientes fases del ciclo de desarrollo de sistemas:

- Investigación (Análisis)
- Definición (Diseño)
- Desarrollo
- Implantación
- Revisión

La figura 3.2 muestra que fases del ciclo de desarrollo de sistemas soporta LDA, LSA y LINC Development.

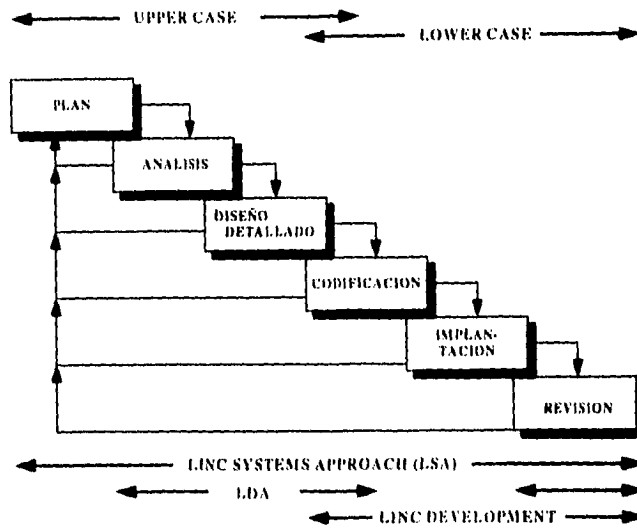


FIGURA 3.2

Se le considera como una herramienta de fácil uso, cuyas características operativas son:

- Dirigida por íconos
- Gráficos en color
- Sistema Operativo DOS
- Ambiente Windows
- Capacidad de multiusuario, basado en LAN

LDA III es la versión que elegimos para desarrollar el ejemplo definido en la sección del planteamiento del problema, su ambiente incluye:

- "LDA Development, permite el diseño y el desarrollo completo de un sistema.
- Repositorio, contiene la definición de los segmentos del negocio.

- LDA Run-Time, interpreta la definición del segmento del negocio que contiene el repositorio, incluye un debugger, y almacena las transacciones de la base de datos en un archivo log.
- Run-time database, almacena la información del sistema LINC al ejecutarse LDA Run-Time, la estructura de la Base de Datos es tomada directamente del repositorio.
- Run-time debugger, permite tener un seguimiento de la ejecución de la lógica.
- Interface to host LINC Development System, permite extraer completa o parcialmente el segmento de negocio de LDA III hacia LINC Development, así como realizar una reingeniería.”<sup>3</sup>

### ***Estructura de un Modelo de Negocio LINC***

Un modelo de negocio LINC está formado por los siguientes componentes :

- Segmentos del Negocio
- Areas Funcionales
- Actividades
- Objetos

El modelo cubre a la organización como un todo, es decir diversos segmentos del negocio; se divide en áreas funcionales, las cuales en su momento son desglosadas en actividades relacionadas y registradas.

Una actividad del negocio, esta compuesta por un grupo de objetos, los cuales conjuntamente ejecutan una función del negocio. Un área funcional puede estar compuesta por varias actividades; y a su vez una actividad puede existir en mas de un área funcional.

Después de que se ha identificado el modelo que representa a la organización, los pasos involucrados son :

- Agregar Actividades

Las actividades y áreas funcionales se registran. Las actividades se agrupan por áreas funcionales creando una matriz funcional área/actividad. Las matrices ayudan a visualizar y comparar áreas funcionales, actividades y datos.

- Definición de las Actividades

Cada actividad se analiza en su turno, cada una se descompone en uno o más eventos. Cada evento se define con sus pantallas y reglas del negocio. Los componentes que soportan cada evento son definidos junto con sus pantallas, reglas del negocio, y lógica global cuando se requiera. Por último se definen las relaciones que existen entre eventos y componentes.

- Agregar perfiles y reportes

Habiendo definido eventos y componentes del modelo es necesario definir los perfiles (vistas) que serán usados para acceder a la base de datos del sistema de información LINC, junto con los reportes e inquires (consultas) que se usarán.

- Definición de los grupos de datos

Los grupos de datos usados en el modelo se definen en el diccionario de datos, ya sea en el global o en el local.

- Definición del modelo

LDA proporciona varias facilidades que permiten revisar el modelo para asegurar que alcanzará un desempeño adecuado (Auditoría de diseño y Análisis de puntos de función).

### **3.2.1 AUDITORIA DE DISEÑO**

La facilidad de auditoría de diseño analiza el modelo o área funcional y reporta cualquier elemento que pueda indicar un diseño pobre que impida un desempeño adecuado del sistema de información resultante. El análisis cubre el uso racional de :

- Eventos
- Componentes
- Grupo de datos (Duplicidad y redundancia)
- Perfiles
- Uso del disco
- Palabras reservadas

La facilidad de auditoría de diseño es un efectivo y eficiente método de predicción de cual será el desempeño del sistema, en una etapa temprana del ciclo de desarrollo.

A través de la ingeniería reversible, es posible que a partir de una especificación del sistema desarrollado en LINC II se pueda obtener el modelo LDA, por lo que una auditoría de diseño se puede realizar a partir de cualquier especificación existente.

### **3.2.2 ANALISIS DE PUNTOS DE FUNCION (FPA: Function Point Analysis)**

Los puntos de función son una técnica para medir la productividad del software; se basa en la inspección de la aplicación y mide el tamaño del sistema en términos de números de funciones. También sirve para determinar el tamaño del proyecto a desarrollar, con esto se puede realizar una estimación del costo y número de personas requeridas para su realización.

“ Los puntos de función se determinan (en opinión del usuario y/o el analista) en dos pasos:

1. Clasificar y contar los siguientes elementos de un programa a tres niveles de complejidad:

- entradas externas del usuario
- salidas externas del usuario
- consultas
- archivos maestros desde el punto de vista lógico del usuario
- interfase con otros programas de aplicaciones

A continuación se aplica un factor de ponderación al total de cada elemento y en cada clasificación de complejidad se suman los resultados para tener el total de puntos de función no ajustados.

2. Los puntos de función no ajustados deben ajustarse según el nivel de influencia (0 a 5) de cada uno de los siguientes factores sobre el diseño y la implantación del programa:

- Actualización en línea de archivos maestros
- Comunicaciones de datos
- Procesamiento distribuido
- Objetivos de desempeño
- Estrechez de la configuración
- Volumen de las transacciones
- Captura de datos en línea
- Transacciones interactivas en línea
- Complejidad de procesamiento
- Diseño que permite volver a utilizarlo
- Facilidad de conversión e instalación
- Facilidad de operación
- Posibilidad de instalación múltiple
- Facilidad de modificación y uso

Este rango puede modificar los puntos de función no ajustados dentro del rango  $\pm 35\%$ .<sup>4</sup>

Para proyectos LINC, utilizando una metodología estructurada se usa un factor de productividad de 4.5 a 5 hrs. por punto de función. Utilizando la metodología LSA el factor va de 2.5 a 3.5 hrs. por punto de función.

LDA provee de una facilidad automatizada que usa reglas y estándares de FPA.

Esta facilidad analiza un modelo del negocio LINC y asigna un valor al punto de función, ya sea de la organización o a nivel del área funcional. A través de FPA, se provee de la habilidad para calcular:

- Puntos de función no ajustados (Brutos)
- Puntos de función ajustados

El factor de ajuste utilizado para calcular los puntos de función ajustados, se deriva del peso de una serie de factores complejos.

Para poder calcular dichos puntos, es necesario seguir las siguientes reglas en cada una de las actividades consideradas para tal efecto:

- Cuantificar entradas
- Cuantificar salidas
- Cuantificar consultas
- Cuantificar archivos lógicos
- Cuantificar interfases
- Ajustes de complejidad

**Cuantificar entradas:**

Para considerar las entradas, es necesario cuantificar cada una de ellas dentro de los límites del área funcional LDA, considerando como tales las altas, bajas y cambios en un archivo lógico (Componente o Evento). Una entrada se considera única si tiene un formato diferente o si su diseño requiere una lógica de procedimiento diferente de otras entradas con el mismo formato. Incluye ambas entradas: de usuarios y de sistemas externos.

Los componentes son típicamente contabilizados como tres entradas: altas, bajas y cambios. Los eventos típicamente se cuentan como una entrada (altas). Cuenta una salida por cada entrada para errores o mensajes de confirmación.

Tabla de complejidad de entradas

Tipos de Archivo	Elementos de Datos Referenciados		
	1 a 12	5 a 15	16 ó más
0 ó 1	simple	simple	promedio
2	simple	promedio	complejo
3 ó mas	promedio	complejo	complejo

Para bajar o incrementar la complejidad, pero no por más de un nivel, es necesario considerar los siguientes puntos:

- Movimiento del cursor
- Otros factores humanos
- Conversión de datos
- Desempeño del sistema



**Cuantificar salidas:**

Para cuantificar las salidas, es necesario contar cada una de ellas dentro del área funcional de LDA. Una salida se considera única si tiene formato diferente o si el diseño requiere una lógica de proceso diferente de otras salidas con el mismo formato. Incluye salidas que se dejan directamente como reportes y mensajes al sistema.

Los despliegues de información en pantalla (consultas y mensajes en video) se cuentan típicamente como salida. Cada formato de reporte (LINC report frame o Screen inquiry) se cuenta como una salida.

Los campos de suma en un reporte de detalle no se cuentan como salidas únicas.

La complejidad se puede medir de acuerdo a:

Tabla de complejidad de Salidas

Tipos de Archivo	Elementos de Datos Referenciados		
	1 a 5	6 a 19	20 ó más
0 ó 1	simple	simple	promedio
2 ó 3	simple	promedio	complejo
4 ó más	promedio	complejo	complejo

Para los reportes se recomienda usar la siguiente información adicional de reglas de complejidad:

**Simple:**

- Una o dos columnas
- Derivaciones de datos simples

**Promedio:**

- Múltiples columnas
- Múltiples derivaciones de datos

Compleja:

- Complejas derivaciones de datos
- Múltiples y complejas referencias a archivos
- Consideraciones significantes de desempeño

Para disminuir o aumentar la complejidad por más de un nivel, se debe tomar en cuenta:

- Factores humanos
- Número de tipos de subtotales
- Derivaciones de datos
- Desempeño del sistema

***Cuantificar consultas:***

Contar cada una de las combinaciones de E/S donde una entrada provoque una salida inmediata. Una consulta es única si tiene un formato diferente de otros tipos de consultas del mismo formato. Incluye consultas que entran directamente del usuario y aquellas que entran de sistemas externos.

Para distinguir consultas de entradas, consideramos que la entrada de datos de una consulta influye únicamente en la búsqueda, no actualiza a ningún archivo lógico. Las pantallas de ayuda se cuentan como una consulta.

Las pantallas de los menús que únicamente sirven para seleccionar no se cuentan como consultas, sólo las que además proporcionen recuperación de datos. No se consideran consultas los mensajes de error o confirmación asociados a una consulta.

Para evaluar la complejidad :

1. Medir la parte de Entrada como si fuera una entrada.
2. Medir la parte de Salida como si fuera una salida.
3. La complejidad de la consulta es la mayor de las dos.

Además hay que considerar para aumentar o disminuir la complejidad, no más de un nivel:

- Desempeño del Sistema
- Criterios de Búsqueda

- Recuperaciones
- Respaldos
- Distribución múltiple
- Grado de complejidad de conversión de datos

**Cuantificar archivos lógicos:**

En este punto, se cuentan los archivos lógicos no archivos físicos. Considera a los componentes y eventos de uso Salida ó Entrada como un archivo. Además cuenta el primer profile de un componente o evento como un archivo lógico. No se consideran a los profiles subsiguientes de uno ó más eventos como archivos lógicos.

Se utiliza la siguiente tabla para medir la complejidad:

Tabla de complejidad de Consultas

Tipos de Archivo	Elementos de Datos Referenciados		
	1 a 19	20 a 50	51 ó más
1	simple	simple	promedio
2 a 5	simple	promedio	complejo
6 ó más	promedio	complejo	complejo

Además hay que considerar lo siguiente para aumentar o disminuir la complejidad por no más de un nivel.

- Desempeño del sistema
- Criterios de búsqueda
- Respaldos
- Recuperaciones

**Interfases a Archivos Externos:**

Aumenta cada uno de los principales grupos de datos o información utilizados pero que se mantienen fuera del área funcional LDA.

La matriz completa de puntos de función calculados se puede invocar desde una hoja de cálculo para su posterior análisis; esta facilidad también permite calcular el tamaño de un sistema existente, determinándose por ingeniería reversible a LDA.

Esto proporciona los detalles para determinar cómo los esfuerzos actuales de los desarrolladores se comparan contra un estándar dado o para construir un registro histórico de la productividad del personal.

### 3.2.3 CONSTRUCCION DE PROTOTIPOS

El pintador de pantallas tiene la facilidad de permitir simular una sesión con el usuario, desplegando un número de pantallas de salida.

También, provee la habilidad para ver el aspecto de una parte específica del sistema de información. Visto en ejecución con el usuario, tiene un impacto significativo en la reducción del tiempo de desarrollo e incremento en la satisfacción del usuario. En la figura 3.3 se muestra el desarrollo de un prototipo en LDA.

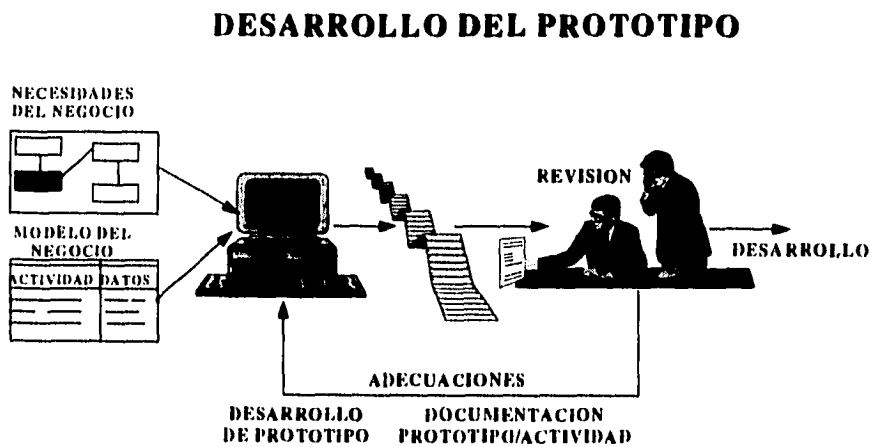


FIGURA 3.3

La mayor parte de la definición del modelo se realiza gráficamente usando íconos y mouse. Esto provee una rápida y fácil construcción del modelo, mientras al mismo tiempo hace fácil la verificación visual en su exactitud.

Las capacidades gráficas de LDA tienen las mismas características que las de cualquier otro producto dirigido a computadoras personales con manejo de mouse.

Aunque la definición de la especificación es la función principal del sistema de desarrollo LINC II, gran parte se puede definir con LDA. Esto agrega un mecanismo efectivo para el análisis y diseño del sistema de información LINC.

---

<sup>1</sup> UNISYS, LINC System Approach, Pág. 1-4, 1992.

<sup>2</sup> UNISYS, LINC Systems Approach, Pág. 1-5, 1992.

<sup>3</sup> UNISYS, LINC II, Technical Overview, The Business. CASE, Pág. 3-4, 1992.

<sup>4</sup> Rudolph , E. Lenguajes de cuarta generación y productividad en el desarrollo de Sistemas, Pág. 27-28, 1983.

## **CAPITULO IV: INVESTIGACION DEL SISTEMA**

El presente capítulo se refiere a la primera fase de la metodología LSA; contiene información detallada sobre "La Marina Acapulco": su ubicación, giro principal, objetivo estratégico y servicios que proporciona. Se muestran de manera específica los problemas detectados junto con la administración del negocio, lo cual nos permite tener una visión sobre la situación actual de la empresa, y da paso para delimitar el sistema de información definiendo objetivos, alcances, recursos humanos y materiales; obteniendo como resultado el plan del proyecto a desarrollar.

### **4.1 PLAN DE SOLUCIONES**

Antes de dar inicio a la fase de investigación del sistema, presentamos el Plan de Soluciones de Información, que define el alcance y objetivos del segmento del negocio bajo estudio; este plan enfoca al negocio como un todo, planea una estrategia para el uso de la tecnología de información y para el desarrollo de sistemas que soporten los objetivos estratégicos de la empresa; y esto representa la entrada para iniciar el ciclo de vida de LSA (LINC System Approach).

### **OBJETIVOS DE LA EMPRESA**

"La Marina Acapulco es un megaproyecto cuyo objetivo estratégico principal es satisfacer la demanda de servicios náuticos para vacacionistas de fin de semana en el puerto de Acapulco."<sup>1</sup>

Esta empresa se divide en tres ramas: el Club Náutico, los Condominios y la Marina Turística, siendo ésta última su giro principal; a continuación mencionamos los servicios que se proporcionan en cada una de las ellas.

## **MARINA TURISTICA**

“Se entiende por Marinas Turísticas al conjunto de instalaciones marítimas y terrestres contruídas para proporcionar abrigo y servicios a embarcaciones de recreo y deportivas.”<sup>2</sup>

La Marina Acapulco ofrece: los servicios de una Marina Turística autorizada por el Gobierno Federal (Full Service Marina), slips (lugar donde son colocadas las embarcaciones en el muelle) individuales desde 36 hasta 150 pies.

La Marina ha sido planeada para ofrecer:

- 272 Slips individuales para atraque a largo plazo de yates y veleros
- 60 posiciones de atraque para embarcaciones menores
- Servicio de succión (*Pump Out*) de los desechos sanitarios
- Suministro de agua potable
- Grúa y rampa para el botado de embarcaciones menores
- Servicio de energía eléctrica
- Servicio telefónico a la embarcación
- Suministro de combustible
- Teléfonos públicos
- Radio comunicación 24 horas
- Instalaciones ocultas

0

La concepción de La Marina Acapulco incluye los servicios de:

- Capitanía de puerto
- Recinto fiscal
- Migración y aduana
- Información meteorológica por satélite

Además cuenta con:

- Casilleros para equipos de embarcaciones
- Servicios Generales para tripulaciones
- Ferretería marina y servicios de aprovisionamiento



La seguridad para los usuarios y las embarcaciones se refuerza con:

- Equipo de rescate
- Lancha salvavidas
- Sistema de Control de incendios a base de agua y químicos por tierra y mar
- Frecuencias de radio para comunicaciones
- Vigilancia terrestre y marina las 24 horas
- TV. vía satélite hasta el muelle
- Estacionamiento de automóviles
- Medidas de protección especializadas
- Alumbrado general
- Reciprocidad con otras marinas a nivel Internacional a través del Club Náutico La Marina

“El plan maestro del Desarrollo fue aprobado por la SEDUE (Secretaría de Desarrollo Urbano y Ecología), SCT (Secretaría de Comunicaciones y Transportes), Gobierno del Estado de Guerrero y autoridades competentes, para garantizar que el mismo mejore y conserve la ecología de la zona.”<sup>3</sup>

### ***CLUB NAUTICO***

“El Club Náutico La Marina responde a la creciente demanda de instalaciones náutico turísticas de primera categoría que ofrezcan hospitalidad, excelente calidad en lo servicios y amplia gama de actividades acuáticas recreativas para disfrutar al máximo.

El Club Náutico La Marina ofrece:

- Playa exclusiva, tranquila y segura
- Club de playa con equipamiento de lujo
- Amplios jardines
- Alberca, wet bar y grill bar
- Servicio de vigilancia y salvavidas
- Alberca de agua dulce flotante en el mar
- Cancha de Paddle Tennis
- Seguridad y privacidad, control de accesos y vigilancia

- Club House dotado de:
  - Restaurante bar con terraza cubierta.
  - Baños con vapor
  - Vestidores
  - Gimnasio
- Area para eventos sociales de primera categoría.”<sup>4</sup>

El Club cuenta con facilidades para:

- Botado de embarcaciones menores
- Estacionamiento seco para embarcaciones menores
- Muelle para amarre de embarcaciones hasta de 24 pies de eslora

El Club Náutico La Marina ofrece posibilidades para realizar:

- Regatas y cursos de velleo
- Torneos y práctica de la pesca deportiva
- Torneos y cursos de perfeccionamiento de esquí acuático
- Travesías y paseos en la Bahía de Acapulco y en altamar
- Instructores de velleo, pesca, esquí, windsurf, buceo y otros
- Integración de equipos de competencia
- Eventos sociales y culturales

## **CONDOMINIOS**

“Los Condominios del Desarrollo Náutico Turístico La Marina cuentan instalaciones que permiten gozar del ambiente del Club Náutico.

Los Condominios La Marina ofrecen:

- Penthouses y condominios de una, dos y tres recámaras
- Acceso directo a La Marina Acapulco
- Albercas
- Areas jardinadas
- Administración central
- Aprovechamiento de las instalaciones del Club Náutico La Marina
- Conserjería las 24 horas
- Seguridad, vigilancia y control de accesos las 24 horas
- Estacionamiento techado para un automóvil por departamento

- Mantenimiento de edificios, jardines y albercas
- Televisión vía satélite
- Aire acondicionado central con controles individuales.<sup>5</sup>

### **PROBLEMATICA**

Trabajando con la administración de Promotora Marina de Acapulco, los problemas que se detectaron fueron los siguientes:

- Falta de oportunidad en la obtención de la información que se envía a la Capitanía de Puerto.
- Consumo de largos periodos de tiempo en la elaboración de reportes que arrojan información sobre el consumo de servicios por embarcación, por slips y de la Marina en general.
- Controles inexistentes que determinen la ocupación de la Marina.
- Retardos e información no disponible para la elaboración de facturas.
- Falta de información oportuna para la toma de decisiones.
- Procesos lentos y complejos para generar información a sus clientes sobre propagandas o eventos que lleve a cabo La Marina.
- Información no disponible para hacer proyecciones financieras y solicitudes de crédito.
- Procedimiento complicado e inexacto para la obtención del costo de la renta por slip por embarcación.

Con base en lo anterior y a la información presentada en el apartado de Planteamiento del Problema, proponemos la elaboración del Sistema de Control de Embarcaciones y Servicios de La Marina (SICES), el cual permitirá agilizar y controlar de manera más eficiente las actividades que lleva a cabo La Marina en su giro principal. Es importante mencionar que en las otras actividades del negocio (Club Náutico y Condominios) existen problemas, sin embargo los que requieren de una solución inmediata a juicio de su propia administración son los de La Marina.

## **4.2. INVESTIGACION DEL SISTEMA**

### **4.2.1. OBJETIVOS, METAS Y RESTRICCIONES**

#### ***Objetivos:***

- Brindar mejor servicio al cliente
- Agilizar los procesos involucrados en la operación de La Marina
- Obtener información confiable, eficiente y completa para apoyar el proceso de toma de decisiones
- Agilizar los flujos de información
- Mejorar el control de los recursos disponibles
- Obtener un mejor seguimiento de la ubicación de las embarcaciones y de la ocupación de la Marina
- Mejor canalización de los esfuerzos del personal, enfatizando la atención al cliente
- Cumplir puntualmente con los requisitos legales de la operación de la Marina
- Incrementar la formación educativa del personal de la empresa

#### ***Metas:***

- Reducir en un 30% las labores administrativas del personal.
- Incrementar la ocupación de la Marina para que ésta fluctúe entre el 85% y 95% del total de su capacidad, lo cual se logrará atrayendo más clientes y ofreciendo mayor calidad en los servicios.
- Reducir los Costos de Operación en un 10% (en la Administración de la Marina).
- Agilizar los procesos de facturación, elaboración de reportes de gastos por embarcación, cuentas parciales de consumo de servicios (pre-factura), etc. obteniéndolos en pocos minutos.
- Agilizar el suministro de información necesaria para la contabilidad, estando disponible el primer día hábil después de cada mes.
- Calendarizar un plan educativo con el fin de capacitar al personal en la operación del nuevo sistema durante los quince días siguientes a la liberación del sistema.

**Restricciones:**

El sistema no cubrirá ningún manejo contable, ni de análisis ni de proyecciones financieras, pero proporciona todas las salidas necesarias para integrar en un futuro el módulo correspondiente a la contabilidad.

Para la etapa de definición del sistema se cuenta con el equipo y software necesarios, sin embargo una de las restricciones es que actualmente La Marina no cuenta con el hardware y software necesarios para la fase de implantación.

En lo que se refiere al aspecto económico se cuenta con los recursos para las tres primeras fases que son investigación, definición y desarrollo del sistema, y en base a los resultados obtenidos en éstas etapas, la Administración y el equipo de analistas, tomarán la decisión de continuar el proyecto y por lo tanto asignar los recursos para su implantación.

#### **4.2.2 PERSONAL Y APTITUDES**

Por las dimensiones y complejidad del sistema, se requieren cuatro personas con Licenciatura en Informática, Ingeniería en Sistemas ó carrera afín; que cuenten con conocimientos de la metodología de desarrollo LSA y conozcan el manejo de alguna herramienta CASE, de preferencia LDA, tengan amplios conocimientos de las plataformas: PC, mainframe ó Unix, y gran capacidad analítica en el diseño de sistemas, cualidades de liderazgo y trabajo en equipo.

#### **4.2.3. ESTANDARES DEL PROYECTO**

Considerando la paquetería con la que cuenta La Marina, definimos los siguientes estándares para presentar a la Administración los avances del proyecto.

- Herramienta CASE que se va ocupar: LDAIII
- Procesador para entrega de reportes: WINWORD
- Elaboración de Presentaciones: POWERPOINT

#### 4.2.4. ORGANIGRAMA Y DESCRIPCION DE LAS AREAS FUNCIONALES

A continuación se presenta el organigrama de la Operadora Náutica La Marina de cual se desprenden los siguientes segmentos del Negocio:

- Auditoría
- Adquisiciones
- Cruceros Santa Lucía
- La Marina Turística
- Club Náutico
- Condominios

#### ORGANIGRAMA DE LA MARINA



FIGURA 4.1

El sistema en desarrollo involucra al segmento de La Marina Turística, el cual comprende las siguientes áreas funcionales:

##### **Contabilidad.**

Es el área funcional encargada del registro, revisión y control de las operaciones financieras de la empresa.

##### **Control de embarcaciones**

Es el área funcional que se encarga de administrar y controlar los espacios disponibles para el atraque de embarcaciones de La Marina.

***Vigilancia***

Es el área funcional que se dedica a mantener la seguridad dentro de La Marina, así como de dar aviso de las anomalías existentes en la zona de atraque.

***Mantenimiento de instalaciones***

Esta área funcional se encarga de verificar y hacer posible el óptimo funcionamiento de las instalaciones de La Marina.

***Prestación de servicios***

Es el área funcional que se encarga de registrar, controlar y suministrar la diversidad de servicios que se proporcionan a las embarcaciones atracadas en La Marina: agua, luz, TV por satélite, etc..

De las áreas funcionales mencionadas anteriormente, las involucradas en el sistema aparecen remarcadas en el organigrama; y son las que aparecen en la matriz resumen de actividad/área funcional.

ACTIVIDADES	CONTROL DE EMBARCACIONES	SERVICIOS
1. Registro de embarcaciones que arriban.	X	
2. Asignación de Slip.	X	
3. Cambio de Slip	X	
4. Entrega de Reporte de Capitanía de Puerto.	X	
5. Entrega de Reporte a Policía Federal de Caminos y Puertos.	X	
6. Facturación.	X	X
7. Proporcionar Estado de Cuenta al Cliente.	X	
8. Registro de Servicios.		X
9. Actualización de Tarifas.		X
10. Elaboración de Reportes de Ingresos para Contabilidad.	X	
11. Información de Servicios que proporciona La Marina.		X
12. Elaboración de Orden de Salida.	X	

#### 4.2.5 REQUERIMIENTOS DEL USUARIO

Junto con el usuario se plantearon tres tipos de requerimientos: captura y consultas de información en línea y reportes, todo esto con el fin de tener información completa y disponible para otras aplicaciones.

##### 4.2.5.1 CAPTURA

###### *Captura de tarifas por servicio*

Se utilizará para registrar las tarifas de cada servicio que proporciona La Marina.

###### *Captura de los datos por embarcación*



Esta pantalla se utilizará para dar de alta los datos generales de cada una de la embarcaciones que atraquen en La Marina. (Solicitud de servicios de atraque)

***Registro de servicios***

Esta pantalla se utilizará para registrar los servicios que se proporcionan a cada una de las embarcaciones atracadas en La Marina; por lo que debe reflejar las transacciones que se realizan.

***Captura de slips***

Esta pantalla deberá permitir registrar los slips con los que cuenta La Marina.

***Captura de movimientos de slips***

Esta pantalla se utilizará para poder re-asignar slips a las embarcaciones que ya se encuentran atracadas en La Marina.

#### **4.2.5.2 CONSULTAS**

***Consulta de ocupación de La Marina***

Servirá para conocer que slips se encuentran ocupados y porque embarcación.

***Consulta de datos generales por embarcación***

Esta pantalla proporcionará todos los datos de determinada embarcación.

***Consulta de tarifas por servicio***

Servirá para obtener información sobre los servicios que brinda La Marina, así como las tarifas de cada uno de éstos.

#### **4.2.5.3 REPORTE**

***Reporte de Ocupación de La Marina***

Este reporte proporciona el total de embarcaciones, y de pies, así como los slips que se encuentran ocupados en la fecha en que se solicitó el reporte, pudiéndolo generar de manera diaria y mensual.

***Reporte de ingresos***

Este reporte refleja el total de ingresos obtenidos por La Marina en un mes, indicando el monto de cada uno de los servicios que se brindaron al cliente y que ya han sido facturados. La periodicidad de este reporte es mensual.

***Reporte de Capitanía de Puerto***

Este reporte proporciona los datos generales de cada una de las embarcaciones que han estado en La Marina: nombre de la embarcación y del capitán, país de origen, puerto de entrada y destino, y fecha de arribo; su periodicidad es mensual y se entrega a la Capitanía del Puerto.

***Reporte para la Policía Federal de Caminos y Puertos***

Este reporte sirve para cumplir con uno de los requisitos legales de La Marina, proporciona los datos generales de las embarcaciones atracadas y la generación del reporte es quincenal.

***Reporte de Estado de Cuenta por Embarcación***

Se utiliza para informar al cliente su saldo, sin ser una factura; la periodicidad de este reporte es variable, ya que se emite cada vez que lo solicita un cliente.

***Factura***

Este proceso consiste en la generación automática de la factura al momento en que el cliente cierra su cuenta.

***Reporte para generar la orden de salida***

Este reporte se genera una vez que el cliente ha liquidado su cuenta, es un requisito para poder abandonar las instalaciones de La Marina, se utiliza como control.

#### **4.2.6 IDENTIFICAR OTROS RECURSOS PARA LA INVESTIGACION Y DEFINICION DEL SISTEMA**

Los recursos necesarios para la Investigación, Definición y Desarrollo del Sistema son:

Software:

- DOS (Ultima versión)
- WINDOWS (Ultima versión)
- WINWORD (Ultima versión)
- POWER POINT (Ultima versión)
- WORD GRAPH
- Software para el manejo del mouse
- LDAIII

Hardware: (Requisitos Mínimos)

- 2 Computadoras Personales 80386
- 8 Mb en RAM
- Disco Duro de 130 Mb
- Drive 3 1/2 HD
- Monitor VGA
- 2 Mouse
- Impresora Laser Jet III
- Impresora de Matriz de puntos

#### **4.2.7 RESUMEN DEL SISTEMA DE INFORMACION Y PLAN DEL PROYECTO**

El sistema SICES (Sistema Integral de Control de Embarcaciones y Servicios) estará diseñado para auxiliar en la administración de La Marina de una manera más eficiente y que proporcione mayores beneficios al negocio. Ofrece un control total de la administración de la Marina por medio de la integración de cada aspecto existente en la misma. El sistema permitirá el acceso de distintos tipos de información: monitoreo, control de acceso, ocupación, servicios, facturación, etc.

La eficiencia de este sistema se obtendrá de varias formas:

- Preparación, generación e impresión automática de las facturas y de otras cuentas de servicios.
- Generación automática de los reportes requeridos por la administración.
- Seguimiento de gastos generados por embarcación durante su estancia en La Marina.
- Información de slips disponibles en los que se pueden acomodar nuevas embarcaciones o transferir embarcaciones ya ubicadas en otros slips.
- Ajuste automático para los rangos de costos de atraques para la renta de slips por tipo de embarcación.

Estas son algunas de las características que pueden mejorar la eficiencia y aumentar los ingresos de La Marina, lo que ayuda al cumplimiento de su objetivo principal.

Implantando todas las capacidades del sistema se pueden reducir en horas las labores que desarrollan actualmente los empleados de la Marina, lo que permitirá tener actividades más productivas y de esta forma recuperar su costo de una manera más rápida.

Este sistema será creado para usarse de forma sencilla, ya que está integrado y estructurado por áreas funcionales, lo que ofrecerá un amplio rango de facilidades para cumplir totalmente con las demandas de La Marina.

Todas las pantallas que conforman al sistema serán diseñadas bajo los mismos estándares, por lo que si se sabe manejar una, prácticamente se sabrán manejar todas.

#### **4.2.8 PRESENTACION**

##### ***Problemas Detectados***

- Falta de oportunidad en la obtención de la información que se envía a la Capitanía de Puerto.
- Consumo de largos periodos de tiempo en la elaboración de reportes.
- Controles inexistentes que determinen la ocupación de la Marina.
- Retardos e información no disponible para la elaboración de facturas.

### ***Problemas Detectados***

- Falta de información oportuna para la toma de decisiones.
- Procesos lentos y complejos para generar información a sus clientes.
- Información no disponible para hacer proyecciones financieras y solicitudes de crédito.
- Procedimiento complicado e inexacto para la obtención del costo de la renta por slip por embarcación.

### ***Propuesta***

- Elaboración del Sistema de Control de Embarcaciones y Servicios de La Marina (SICES), el cual permitirá agilizar y controlar de manera más eficiente las actividades que lleva a cabo La Marina en su giro principal.

### ***Objetivos del Sistema***

- Brindar mejor servicio al cliente.
- Agilizar los procesos involucrados en la operación de La Marina.
- Obtener información confiable, eficiente y completa para apoyar el proceso de toma de decisiones.
- Agilizar los flujos de información.
- Mejorar el control de los recursos disponibles.
- Obtener un mejor seguimiento de la ubicación de las embarcaciones y de la ocupación de La Marina.

### ***Objetivos del Sistema***

- Mejor canalización de los esfuerzos del personal, enfatizando la atención al cliente.
- Cumplir puntualmente con los requisitos legales de la operación de La Marina.
- Incrementar la formación educativa del personal de la empresa.

### ***Metas del Sistema***

- Reducir en un 30% las labores administrativas del personal.
- Incrementar la ocupación de La Marina para que ésta fluctúe entre el 85% y 95% del total de su capacidad, lo cual se logrará atrayendo más clientes y ofreciendo mayor calidad en los servicios.
- Reducir los Costos de Operación en un 10% (en la Administración de La Marina).

### ***Metas del Sistema***

- Agilizar los procesos de facturación, elaboración de reportes de gastos por embarcación, cuentas parciales de consumo de servicios, etc. obteniéndolos en menor tiempo.
- Agilizar el suministro de información necesaria para la contabilidad, estando disponible el primer día hábil después de cada mes.
- Calendarizar un plan educativo con el fin de capacitar al personal en la operación del nuevo sistema durante los quince días siguientes a la liberación del mismo.



### ***Restricciones del Sistema***

- No se tiene contemplado realizar manejo contable de la información, así como tampoco se harán proyecciones financieras.
- Se cuenta con el equipo necesario para las Etapas de Definición y Desarrollo, sin embargo por ahora se carece del mismo para la Etapa de Implantación.

### ***Personal y Aptitudes***

- Cuatro personas con estudios de Licenciatura en Informática, Ingeniería de Sistemas o carrera afín.
- Conocimientos:
  - LSA, LDA
  - PC, Mainframe, Unix
  - Análisis de Sistemas
  - Liderazgo
  - Trabajo en Equipo

### ***Estándares del Proyecto***

- Herramienta CASE a utilizar:            **LDAlll**
- Procesador para entrega de reportes:   **WINWORD**
- Elaboración de Presentaciones:       **POWERPOINT**

### ***Requerimientos del Usuario***

---

- Reporte de Ocupación de La Marina.
- Reporte de ingresos.
- Reporte de Capitanía de Puerto.
- Reporte para la Policía Federal de Caminos y Puertos.
- Reporte de estado de cuenta por embarcación.
- Factura.
- Orden de Salida.

### ***Requerimientos del Usuario***

- Consulta y captura de datos generales por embarcación (Solicitud de Servicios de Atraque).
- Consulta y captura de las tarifas por servicio.
- Registro de Servicios.
- Consulta y captura de slips.
- Captura de movimientos de slips.

### ***Otros Recursos (Software)***

- DOS (Ultima versión)
- WINDOWS (Ultima versión)
- WINWORD (Ultima versión)
- POWER POINT (Ultima versión)
- Software para el manejo del mouse
- LDAIII

## ***Otros Recursos (Hardware)***

### **Requisitos Mínimos**

2 Computadoras Personales 80386

- 8 Mb en RAM
- Disco Duro de 130 Mb
- Drive 3 1/2 HD
- Monitor VGA
- 2 Mouse
- Impresora Laser Jet III
- Impresora de Matriz de puntos

## ***Otros Recursos (Hardware)***

### **En Series A**

#### **Software**

- InfoConnet que incluya FileExpress
- Sistema Operativo MARC 4.0 en adelante
- DMSII 3.92 en adelante
- LINCII Development (Ultima versión)
- Hardware
- Un Equipo Serie A con dos packs (uno para software ambiental, y otro para softwre LINC)

---

<sup>1</sup> Manual de Organización de La Marina de Acapulco, 1989. Pag. 3.

<sup>2</sup> Instructivo para el Concesionamiento de Marinas Turísticas y de Puertos de Abrigo, 1989, Puertos Mexicanos. Pag. 1.

<sup>3</sup> Manual de Organización de La Marina de Acapulco, 1989. Pag. 4.

<sup>4</sup> Manual de Organización de La Marina de Acapulco, 1989. Pag. 12.

<sup>5</sup> Manual de Organización de La Marina de Acapulco, 1989. Pag. 17-20.

## CAPITULO V: DEFINICION DEL SISTEMA

Este capítulo corresponde a la segunda fase de la metodología LSA: *Definición del Sistema*; durante esta fase se define específicamente el sistema de información, respetando alcances, objetivos y actividades descritos en la fase de investigación del sistema. Se retoma la matriz de actividad/área funcional (descrita en el capítulo anterior en el punto de identificación de Areas Funcionales) para empezar a elaborar el modelo de objetos LINC, el cual se va refinando constantemente durante esta fase; para que finalmente se consiga uno o más prototipos del sistema.

En este punto, también se definen los atributos técnicos de la aplicación mostrando los diferentes ambientes de hardware que se pueden elegir.

Siguiendo el cuadro de la Metodología LSA incluido en el capítulo tres, la primera actividad dentro de la fase de *Definición del Sistema* es la determinación de los estándares de diseño.

### 5.1 ESTANDARES DE DISEÑO

Los estándares que seguiremos para la denominación de los objetos involucrados en el modelo se basan en nuestra experiencia laboral y en las características propias de la herramienta upper CASE que utilizamos.

El número máximo de caracteres permitidos depende del objeto LINC:

#### *Ispecs (componentes y eventos)*

Se manejan cinco caracteres que se utilizarán de la siguiente manera:

Las dos primeras letras referirán el tipo de ispec, es decir, catálogo, menú y proceso, el siguiente caracter identifica el tipo de operación que realiza el ispec a la base de datos: mantenimiento, consulta, actualización, etc.. Las dos últimas letras proporcionan información de la aplicación.

Ejemplo:

**Catálogo de Servicios: CAMSE**  
donde: **CA** = Catálogo  
**M** = Mantenimiento  
**SE** = Servicios

**Proceso para Transacciones: PRMTR**  
donde: **PR** = Proceso  
**M** = Mantenimiento  
**TR** = Transacciones

### **Reportes**

Para los reportes se cuenta con un máximo de diez caracteres, los cuales se distribuirán de la siguiente manera:

Los tres primeros caracteres para identificar que se trata de un reporte, la siguiente letra se utilizará para distinguir la periodicidad del proceso (diario, quincenal, mensual y eventual) y el resto de los caracteres serán para información propia del reporte.

Ejemplo:

**Reporte Diario por embarcación: REPDXEMBRC**  
donde: **REP** = Reporte  
**D** = Diario  
**XEMBRC** = Por Embarcación

### **Perfiles**

Para definir los perfiles (vistas) se cuenta con un máximo de nueve caracteres; los cuales se utilizarán de la siguiente manera:

Las primeras dos letras se utilizarán para denotar que se trata de un perfil, los siguientes cinco caracteres corresponderán al nombre de la estructura a la que se hace referencia y las últimas dos posiciones son para definir el número consecutivo sobre la estructura a la cual corresponde el perfil.

Ejemplo:

**Profile de Embarcaciones-Servicios: PFCAMSE01**  
 donde: **PF** = Profile  
**CAMSE** = Catálogo de Servicios  
**01** = Primer profile de CAMSE

### **Otros estándares**

En cuanto a los demás estándares definidos para las variables del diccionario de datos, se optó porque fueran nombres mnemotécnicos y representativos de los datos a los que hacen referencia.

Para las pantallas y reportes se elaboraron lógicas globales en donde se diseñaron encabezados, títulos y pies de página que se ocupan en gran parte del diseño; el nombre definido para las lógicas globales, fue de acuerdo a la función que realizan.

## **5.2 MODELO DE OBJETOS LINC**

A partir de la matriz de actividades/área funcional, se definen los objetos y actividades involucrados en el modelo del negocio.

A continuación se muestra una tabla con cada uno de ellos:

<b>ACTIVIDAD</b>	<b>OBJETO</b>
Asignar slips	CAMSL
	PRMSL
	MENSL
	PFPRMSL01
Elaborar Estado de Cuenta	PRCEC
	REPECTA
Facturar	PRMFA
	PRIFA
	REPEFAC
	PFPRMFA01
	PFPRMFA02
Elaborar Orden de Salida	REPEORSAL
Registrar embarcaciones	CAMEM



ACTIVIDAD	OBJETO
	PRMCO
	REPDOCMAR
	REPMOCMAR
Realizar reporte de Capitanía de Puerto	REPMCAPTO
Realizar reporte de Ingresos	REPMING
Realizar reporte para la Policía Federal	REPQPOLFED
Actualizar tarifas	CAMSE
Informar servicios	PRCSE
Registrar servicios	MENSE
	PRCEC
	PRMTR
	PFPRMTR01
	PFPRMTR02
	PFPRMTR03
	MENGR
	MENRE

En el siguiente paso de la metodología LSA, se define para cada objeto la información correspondiente: descripción, tipo de objeto, uso, volumen de transacciones esperado, datos que lo conforman, características de edición para los datos y reglas del negocio asociadas con cada objeto.

Para presentar lo anterior, clasificamos la información en varias tablas, en la primera incluimos el nombre del objeto, su descripción, el tipo a que corresponde, su uso y el volumen de transacciones esperado; en las demás se muestran los objetos, sus datos y las características de éstos. Las tablas se clasificaron por los diferentes tipos de objetos: specs, profiles, reportes y lógicas globales.

**Claves:**

COM	Componente	EVE	Evento
PRO	Profile	REP	Reporte
GLG	Lógica Global	E	Entrada
S	Salida	INS	Insertable
ASC	Ascendente		

OBJETO	DESCRIPCION	TIPO	USO	VOLUMEN
CAMSL	Catálogo de slips	COM	E-S	100
CAMEM	Catálogo de embarcaciones	COM	E-S	100
CAMSE	Catálogo de servicios	COM	E-S	100
MENSL	Menú de slips	COM	E	
MENSE	Menú de servicios	COM	E	
MENGR	Menú general del sistema	COM	E	
MENRE	Menú de reportes	COM	E	
PRMSL	Proceso para asignar slips	COM	E-S	100
PRIFA	Proceso que inicia reportes	COM	E	
PRMFA	Proceso para factura	COM	E-S	100
PRMCO	Proceso para consecutivo	COM	S	100
PRCSE	Consulta de servicios	COM	E	100
PRCEC	Proceso para Edo. de cuenta	COM	E	
PRMTR	Proceso para transacciones	EVE	E-S	1000
PFPRMSL01	Profile sobre PRMSL	PRO		
PFPRMFA01	Profile sobre PRMFA	PRO		
PFPRMFA02	Profile sobre PRMFA	PRO		
PFPRMTR01	Profile sobre PRMTR	PRO		
PFPRMTR02	Profile sobre PRMTR	PRO		
PFPRMTR03	Profile sobre PRMTR	PRO		
REPECTA	Reporte de Edo. de Cuenta	REP		
REPEFAC	Reporte para generar Factura	REP		
REPEORSAL	Reporte de Orden de Salida	REP		
REPDOCMAR	Reporte de Ocupación Diario	REP		
REPMOCMAR	Reporte de Ocupación Mens.	REP		
REPMCAPTO	Reporte de Capitanía de Pto.	REP		
REPMING	Reporte de Ingresos	REP		
REPQPOLFED	Reporte para Policía Federal	REP		
GL-NAVEGA	Lógica para navegación	GLG	INS	
GL-NAVEGACP	Lógica de navegación copyf.	GLG	INS	
GL-OPMENU	Lógica para desplegar opción	GLG	INS	
GL-TITULO	Lógica de títulos	GLG	INS	

## 5.2.1 ISPECS

OBJETO	DATO	USO	EDICION	LON	DEC
CAMSL	LOA	E-S	Numérico	3	-
	OCUPADO	E-S	Numérico	1	-
	SLIP	E-S	Alfabético	10	-
CAMEM	BANDERA	E-S	Alfabético	3	-
	CIUDAD	E-S	Alfabético	15	-
	COD-POST	E-S	Alfabético	5	-
	CVE-EMBAR	E-S	Numérico	8	-
	DESCUENTO	E-S	Numérico	2	-
	DIRECCION	E-S	Alfabético	30	-
	FAX	E-S	Alfabético	10	-
	FEC-ARRIBO	E-S	Fecha	6	-
	FEC-RSAL	E-S	Fecha	6	-
	FEC-SALIDA	E-S	Fecha	6	-
	LOA	E-S	Numérico	3	-
	NOM-CAP	E-S	Alfabético	20	-
	NOM-EMBARC	E-S	Alfabético	25	-
	NUM-SEGURO	E-S	Alfabético	10	-
	NUM-TC	E-S	Alfabético	16	-
	PAIS	E-S	Alfabético	3	-
PROPIET	E-S	Alfabético	20	-	
PTO-DESTIN	E-S	Alfabético	10	-	
PTO-ENTRA	E-S	Alfabético	10	-	
SLIP	E-S	Alfabético	10	-	
TELEFONO	E-S	Alfabético	10	-	
CAMSE	CVE-SERV	E-S	Numérico	3	-
	DESCRIP	E-S	Alfabético	20	-
	TARIFA	E-S	Numérico	5	2
MENSE	OP-MENU	E	Numérico	1	-
MENGR	OP-MENU	E	Numérico	1	-
MENRE	OP-MENU	E	Numérico	1	-
MENSL	OP-MENU	E	Numérico	1	-
PRMSL	CVE-EMBAR	E	Numérico	8	-
	FEC-ASIGSL	E	Fecha	6	-
	SLIP	E	Alfabético	10	-
PRIFA					
PRMFA	CVE-EMBAR	E-S	Numérico	8	-
	FEC-SALIDA	S	Fecha	6	-
	FOLIO	E-S	Numérico	5	-
	NOM-FACT	E-S	Alfabético	20	-

OBJETO	DATO	USO	EDICION	LON	DEC
	RFC	E-S	Alfabético	13	-
	TOTAL	S	Numérico	10	2
PRMCO	CONSECUTIV	S	Numérico	8	-
PRCSE	CVE-SERV	E	Numérico	3	-
	DESC	E	Alfabético	20	-
	SERV	E	Numérico	3	-
	TARIFA	E	Numérico	5	2
PRCEC	CVE-EMBAR	E	Numérico	8	-
PRMTR	CVE-EMBAR	E-S	Numérico	8	-
	CVE-SERV	E-S	Numérico	3	-
	FEC-DIA	E-S	Fecha	6	-
	MONTO	E-S	Numérico	8	2
	PAGADO	E-S	Numérico	1	-
	SLIP	E-S	Alfabético	10	-

### 5.2.3 REPORTE

OBJETO	DATO	EDICION	LON	DE
REPECTA	F05-EMBAR	Alfabético	25	-
	F05-NOMBRE	Alfabético	20	-
	F10-CANT	Numérico	3	-
	F10-DESC	Alfabético	20	-
	F10-IMP	Numérico	8	2
	F15-TOTAL	Numérico	10	2
REPEFAC	CAMEM.NOMEMBARC	Alfabético	25	-
	PRMFA.CVE-EMBAR	Numérico	8	-
	PRMFA.FEC-SALIDA	Fecha	6	-
	PRMFA.NOM-FACT	Alfabético	20	-
	PRMFA.RFC	Alfabético	13	-
	F10-CANT	Numérico	3	-
	F10-IMP	Numérico	8	2
	F10-SERV	Alfabético	20	-
	F15-IVA	Numérico	7	2
	F15-SUBT	Numérico	10	2
	F15-TOTAL	Numérico	10	2
REPEORSAL	CAMEM.CVE-EMBAR	Numérico	8	-
	CAMEM.NOMEMBARC	Alfabético	6	-
	F05-FECSAL	Fecha	6	-
REPDOCMAR	F05-FEC	Fecha	6	-
	F05-PAG	Numérico	2	-

OBJETO	DATO	EDICION	LON	DE
	F10-CVEMBA	Numérico	8	-
	F10-LOA	Numérico	3	-
	F10-NOMEMB	Alfabético	25	-
	F10-SLIP	Alfabético	10	-
	F15-TOTEMB	Numérico	3	-
	F15-TOTFT	Numérico	5	-
	F20-CVEMBA	Numérico	8	-
	F20-LOA	Numérico	3	-
	F20-NOMEMB	Alfabético	25	-
	F20-SLIP	Alfabético	10	-
REPMOCMAR	F05-FEC	Fecha	6	-
	F05-PAG	Numérico	2	-
	F10-CVEMBA	Numérico	8	-
	F10-LOA	Numérico	3	-
	F10-NOMEMB	Alfabético	25	-
	F10-SLIP	Alfabético	10	-
	F15-TOTEMB	Numérico	3	-
	F15-TOTFT	Numérico	5	-
	F20-CVEMBA	Numérico	8	-
	F20-LOA	Numérico	3	-
	F20-NOMEMB	Alfabético	25	-
	F20-SLIP	Alfabético	10	-
REPMCAPTO	F05-FECHA	Fecha	6	-
	F05-PAG	Numérico	2	-
	F10-FECAR	Fecha	6	-
	F10-NAC	Alfabético	3	-
	F10-NOMCAP	Alfabético	20	-
	F10-NOMEM	Alfabético	25	-
	F10-PTODES	Alfabético	10	-
	F10-PTOEN	Alfabético	10	-
	F20-FECAR	Fecha	6	-
	F20-NAC	Alfabético	3	-
	F20-NOMCAP	Alfabético	20	-
	F20-NOMEM	Alfabético	25	-
	F20-PTODES	Alfabético	10	-
	F20-PTOEN	Alfabético	10	-
REPQPOLFED	F05-FECHA	Fecha	6	-
	F05-PAG	Numérico	2	-
	F10-FECAR	Fecha	6	-
	F10-NAC	Alfabético	3	-
	F10-NOMCAP	Alfabético	20	-

OBJETO	DATO	EDICION	LON	DE
	F10-NOMEM	Alfabético	25	-
	F10-PTODES	Alfabético	10	-
	F10-PTOEN	Alfabético	10	-
	F20-FECAR	Fecha	6	-
	F20-NAC	Alfabético	3	-
	F20-NOMCAP	Alfabético	20	-
	F20-NOMEM	Alfabético	25	-
	F20-PTODES	Alfabético	10	-
	F20-PTOEN	Alfabético	10	-
REPMING	F10-CANT	Numérico	3	-
	F10-IMP	Numérico	8	2
	F10-SERV	Alfabético	20	-
	F15-TOTAL	Numérico	10	2

### 5.2.3 PROFILES Y LOGICAS GLOBALES

OBJETO	DATO	ORG	EDICION	LON	DEC
PFPRMSL01	CVE-EMABAR	ASC	Numérico	8	-
	FEC-ASIGSL	ASC	Fecha	6	-
PFPRMFA01	FOLIO	ASC	Numérico	5	-
	CVE-EMBAR	ASC	Numérico	8	-
PFPRMFA02	CVE-EMBAR	ASC	Numérico	8	-
	FEC-SALIDA	ASC	Fecha	6	-
PFPRMTR01	FEC-DIA	ASC	Fecha	6	-
	SLIP	ASC	Alfabético	10	-
	CVE-SERV	ASC	Numérico	3	-
PFPRMTR02	FEC-DIA	ASC	Fecha	6	-
	CVE-SERV	ASC	Numérico	3	-
PFPRMTR03	CVE-EMBAR	ASC	Numérico	8	-
	FEC-DIA	ASC	Fecha	6	-

OBJETO	DATO	EDICION	LON	DE
GL-NAVEGA	SIG-PANT	Alfabético	5	-
GL-NAVEGACP	SIG-PANT	Alfabético	5	-
GL-OPMENU	OP-MENU	Numérico	1	-
GL-TITULO				

A continuación se presenta una impresión del Modelo de Objetos LINC en LDA III.

## MARINA Directorio del Modelo

MARINA

CONTROLA

Controladora Náutica La Marina

EMBARCACIONES

Control de Embarcaciones

ASIGNAR-SLIP

Asignar slip a embarcación

CAMSL I/O

Catálogo de slips

LOA

DEFAULT Number 3 L.O.A

OCUPADO

DEFAULT Number 1 Ocupado

SIG-PANT

INPUT Alpha 5 SIG-PANTALLA

SLIP

DEFAULT Alpha 10 Slip

MENSL Input

Menú de slips

OP-MENU

DEFAULT Number 1 Opción del Menú

SIG-PANT

INPUT Alpha 5 SIG-PANTALLA

PRMSL I/O

Proceso para asignar slips

CVE-EMBAR

DEFAULT Number 8 Cve embarcación

FEC-ASIGSL

DEFAULT Date 6 Fecha slip

SIG-PANT

INPUT Alpha 5 SIG-PANTALLA

SLIP

DEFAULT Alpha 10 Slip

PFPRMSL01

PF PRMSL por clave-emb y fecha

CVE-EMBAR

1 Ascending

Clave de la embarcación

FEC-ASIGSL

2 Ascending

Fecha de asignación slip

EDO-CUENTA

Proporcionar edo. cuenta

REPEECTA

Estado de Cuenta del cliente

REPEECTA/00

Encabezado del reporte

REPEECTA/05

Alpha 25

SAS: NOM-EMBARC

F05-EMBAR

Alpha 20

SAS: PROPIET

F05-NOMBRE

Detalle

REPEECTA/10

Number 3

F10-CANT

Alpha 20

SAS: DESCRIP

F10-DESC

Z 8

2

F10-IMP

Totales

REPEECTA/15

Z 10

2

F15-TOTAL

FACTURAR

Elaborar factura

PRIFA Input

Proceso que inicia reportes

PRMFA I/O

Proceso para factura

CVE-EMBAR

DEFAULT Number 8 Cve embarcación

FEC-SALIDA

OUTPUT Date 6 Fecha de salida

FOLIO

DEFAULT Number 5 Folio

NOM-FACT

DEFAULT Alpha 20 Nombre

RFC

DEFAULT Alpha 13 R.F.C.

SIG-PANT

INPUT Alpha 5 SIG-PANTALLA

TOTAL

OUTPUT Number 10 2 Total

REPEFAC

Reporte para generar factura

REPEFAC/00

Encabezado

REPEFAC/05

Alpha 25

Nom embarcación

CAMEM.NOM-EMBARC

Number 8

Cve embarcación

PRMFA CVE-EMBAR

Date 6

Fecha de salida

PRMFA FEC-SALIDA

Alpha 20

Nombre

PRMFA NOM-FACT

Alpha 13

R.F.C.

PRMFA RFC

Detalle

REPEFAC/10

Number 3

F10-CANT

Z 8

2

F10-IMP

Alpha 20

SAS: DESCRIP

F10-SERV

Totales

REPEFAC/15

\$ 7

2

F15-IVA

\$ 10

2

F15-SUBT

\$ 10

2

F15-TOTAL

PFPRMFA01

PF PRMFA por folio y cve-emb

FOLIO

1 Ascending

Folio de la Factura

CVE-EMBAR

2 Ascending

Clave de la embarcación

PFPRMFA02

PF PRMFA por cve-emb.fec-sal

## MARINA Directorio del Modelo

<input type="checkbox"/> CVE-EMBAR		1 Ascending	
<input type="checkbox"/> FEC-SALIDA		2 Ascending	
<b>A</b> <input type="checkbox"/> ORDEN-SALIDA	Elaborar orden de salida		Clave de la embarcación
<b>R</b> <input type="checkbox"/> REPEORSAL	Reporte de orden de salida		Fecha de salida
<input type="checkbox"/> REPEORSAL/00			
<input type="checkbox"/> REPEORSAL/05			
<input type="checkbox"/> CAMEM CVE-EMBAR	Number	8	Cve embarcación
<input type="checkbox"/> CAMEM NOM-EMBARC	Alpha	25	Nom. embarcación
<input type="checkbox"/> F05-FECSAL	Date	6	SAS; FEC-RSAL
<b>A</b> <input type="checkbox"/> REG-EMBARCACIONES	Registrar emb. que arriban		
<b>C</b> <input type="checkbox"/> CAMEM I-O	Catálogo de embarcaciones		
<input type="checkbox"/> BANDERA	DEFAULT Alpha	3	Bandera
<input type="checkbox"/> CIUDAD	DEFAULT Alpha	15	Ciudad
<input type="checkbox"/> COD-POST	DEFAULT Alpha	5	C.P.
<input type="checkbox"/> CVE-EMBAR	DEFAULT Number	8	Cve embarcación
<input type="checkbox"/> DESCUENTO	DEFAULT Number	2	Descuento
<input type="checkbox"/> DIRECCION	DEFAULT Alpha	30	Dirección
<input type="checkbox"/> FAX	DEFAULT Alpha	10	Fax
<input type="checkbox"/> FEC-ARRIBO	DEFAULT Date	6	Fecha de arribo
<input type="checkbox"/> FEC-RSAL	DEFAULT Date	6	Fec real salida
<input type="checkbox"/> FEC-SALIDA	DEFAULT Date	6	Fecha de salida
<input type="checkbox"/> LOA	DEFAULT Number	3	L.O.A.
<input type="checkbox"/> NOM-CAP	DEFAULT Alpha	20	Nombre Capitan
<input type="checkbox"/> NOM-EMBARC	DEFAULT Alpha	25	Nom. embarcación
<input type="checkbox"/> NUM-SEGURO	DEFAULT Alpha	10	Número de seguro
<input type="checkbox"/> NUM-TC	DEFAULT Alpha	16	Tarjeta Crédito
<input type="checkbox"/> PAIS	DEFAULT Alpha	3	País
<input type="checkbox"/> PROPIET	DEFAULT Alpha	20	Propietario
<input type="checkbox"/> PTO-DESTIN	DEFAULT Alpha	10	Puerto Destino
<input type="checkbox"/> PTO-ENTRA	DEFAULT Alpha	10	Puerto Entrada
<input type="checkbox"/> SIG-PANT	INPUT Alpha	5	SIG-PANTALIA
<input type="checkbox"/> SLIP	DEFAULT Alpha	10	Slip
<input type="checkbox"/> TELEFONO	DEFAULT Alpha	10	Teléfono
<b>C</b> <input type="checkbox"/> PRMCO Output	Consecutivo de embarcaciones		
<input type="checkbox"/> CONSECUTIV	OUTPUT Number	8	Consecutivo
<b>R</b> <input type="checkbox"/> REPDOCMAR	Reporte Diario Ocupacion Mar		
<input type="checkbox"/> REPDOCMAR/00			
<input type="checkbox"/> REPDOCMAR/05			
<input type="checkbox"/> F05-FEC	Date	6	SAS; FEC-DIA
<input type="checkbox"/> F05-PAG	Number	2	SAS; TIP-EMBARC
<input type="checkbox"/> REPDOCMAR/10			
<input type="checkbox"/> F10-CVEMBA	Number	8	SAS; CVE-EMBAR
<input type="checkbox"/> F10-LOA	Number	3	SAS; LOA
<input type="checkbox"/> F10-NOMEMB	Alpha	25	SAS; NOM-EMBARC
<input type="checkbox"/> F10-SLIP	Alpha	10	SAS; SLIP
<input type="checkbox"/> REPDOCMAR/15			
<input type="checkbox"/> F15-TOTEMB	Number	3	Totales
<input type="checkbox"/> F15-TOTFT	Number	5	Totales
<input type="checkbox"/> REPDOCMAR/20			
<input type="checkbox"/> F20-CVEMBA	Number	8	SAS; CVE-EMBAR
<input type="checkbox"/> F20-LOA	Number	3	SAS; LOA
<input type="checkbox"/> F20-NOMEMB	Alpha	25	SAS; NOM-EMBARC
<input type="checkbox"/> F20-SLIP	Alpha	10	SAS; SLIP
<b>R</b> <input type="checkbox"/> REP'MOCMAR	Reporte Mensual de Ocupacion		
<input type="checkbox"/> REP'MOCMAR/00			
<input type="checkbox"/> REP'MOCMAR/05			
<input type="checkbox"/> F05-FEC	Date	6	Eneabezado
<input type="checkbox"/> F05-PAG	Number	2	Eneabezado
<input type="checkbox"/> REP'MOCMAR/10			
<input type="checkbox"/> F10-CVEMB	Number	8	SAS; CVE-EMBAR
<input type="checkbox"/> F10-LOA	Number	3	SAS; LOA
<input type="checkbox"/> F10-NOMEMB	Alpha	25	SAS; NOM-EMBARC



## MARINA Directorio del Modelo

<ul style="list-style-type: none"> <li><input type="checkbox"/> F10-SLIP</li> <li><input checked="" type="checkbox"/> REP MOCMAR/15               <ul style="list-style-type: none"> <li><input type="checkbox"/> F15-TOTEMB</li> <li><input type="checkbox"/> F15-TOTFT</li> </ul> </li> <li><input checked="" type="checkbox"/> REP MOCMAR/20               <ul style="list-style-type: none"> <li><input type="checkbox"/> F20-CVEMB</li> <li><input type="checkbox"/> F20-LOA</li> <li><input type="checkbox"/> F20-NOMEMB</li> <li><input type="checkbox"/> F20-SLIP</li> </ul> </li> <li><input checked="" type="checkbox"/> REP-CAPTANIA               <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP M CAPTO                   <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP M CAPTO/00</li> <li><input checked="" type="checkbox"/> REP M CAPTO/05                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F05-FECHA</li> <li><input type="checkbox"/> F05-PAG</li> </ul> </li> <li><input checked="" type="checkbox"/> REP M CAPTO/10                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F10-FECAR</li> <li><input type="checkbox"/> F10-NAC</li> <li><input type="checkbox"/> F10-NOMCAP</li> <li><input type="checkbox"/> F10-NOMEM</li> <li><input type="checkbox"/> F10-PTODIES</li> <li><input type="checkbox"/> F10-PTOEN</li> </ul> </li> <li><input checked="" type="checkbox"/> REP M CAPTO/20                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F20-FECAR</li> <li><input type="checkbox"/> F20-NAC</li> <li><input type="checkbox"/> F20-NOMCAP</li> <li><input type="checkbox"/> F20-NOMEM</li> <li><input type="checkbox"/> F20-PTODIES</li> <li><input type="checkbox"/> F20-PTOEN</li> </ul> </li> </ul> </li> </ul> </li> <li><input checked="" type="checkbox"/> REP-INGRESOS               <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP M ING                   <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP M ING/00</li> <li><input checked="" type="checkbox"/> REP M ING/05</li> <li><input checked="" type="checkbox"/> REP M ING/10                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F10-CANT</li> <li><input type="checkbox"/> F10-IMP</li> <li><input type="checkbox"/> F10-SERV</li> </ul> </li> <li><input checked="" type="checkbox"/> REP M ING/15                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F15-TOTAL</li> </ul> </li> </ul> </li> </ul> </li> <li><input checked="" type="checkbox"/> REP-POLICIA FED               <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP Q POL FED                   <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> REP Q POL FED/00</li> <li><input checked="" type="checkbox"/> REP Q POL FED/05                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F05-FECHA</li> <li><input type="checkbox"/> F05-PAG</li> </ul> </li> <li><input checked="" type="checkbox"/> REP Q POL FED/10                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F10-FECAR</li> <li><input type="checkbox"/> F10-NAC</li> <li><input type="checkbox"/> F10-NOMCAP</li> <li><input type="checkbox"/> F10-NOMEM</li> <li><input type="checkbox"/> F10-PTODIES</li> <li><input type="checkbox"/> F10-PTOEN</li> </ul> </li> <li><input checked="" type="checkbox"/> REP Q POL FED/20                       <ul style="list-style-type: none"> <li><input type="checkbox"/> F20-FECAR</li> <li><input type="checkbox"/> F20-NAC</li> <li><input type="checkbox"/> F20-NOMCAP</li> <li><input type="checkbox"/> F20-NOMEM</li> <li><input type="checkbox"/> F20-PTODIES</li> <li><input type="checkbox"/> F20-PTOEN</li> </ul> </li> </ul> </li> </ul> </li> <li><input checked="" type="checkbox"/> SERVICIOS               <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> ACT-TARIFAS</li> <li><input type="checkbox"/> CAMSE</li> </ul> </li> </ul>	<table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">Alpha</td> <td style="width: 10%;">10</td> <td style="width: 60%;">SAS: SLIP</td> </tr> <tr> <td>Totales</td> <td></td> <td></td> </tr> <tr> <td>Number</td> <td>3</td> <td></td> </tr> <tr> <td>Number</td> <td>5</td> <td></td> </tr> <tr> <td>Extracción</td> <td></td> <td></td> </tr> <tr> <td>Number</td> <td>8</td> <td>SAS: CVE-EMBAR</td> </tr> <tr> <td>Number</td> <td>3</td> <td>SAS: LOA</td> </tr> <tr> <td>Alpha</td> <td>25</td> <td>SAS: NOM-EMBARC</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: SLIP</td> </tr> </table> <p>Entregar reporte de Cap. Pto. Reporte Mensual de Capitanía</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%;"></td> <td style="width: 10%;"></td> <td style="width: 60%;">Encabezado</td> </tr> <tr> <td>Date</td> <td>6</td> <td></td> </tr> <tr> <td>Number</td> <td>2</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Detalle</td> </tr> <tr> <td>Date</td> <td>6</td> <td>SAS: FEC-ARRIBO</td> </tr> <tr> <td>Alpha</td> <td>3</td> <td>SAS: PAIS</td> </tr> <tr> <td>Alpha</td> <td>20</td> <td>SAS: NOM-CAP</td> </tr> <tr> <td>Alpha</td> <td>25</td> <td>SAS: NOM-EMBARC</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-DESTIN</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-ENTRA</td> </tr> <tr> <td></td> <td></td> <td>Extract</td> </tr> <tr> <td>Date</td> <td>6</td> <td>SAS: FEC-ARRIBO</td> </tr> <tr> <td>Alpha</td> <td>3</td> <td>SAS: PAIS</td> </tr> <tr> <td>Alpha</td> <td>20</td> <td>SAS: NOM-CAP</td> </tr> <tr> <td>Alpha</td> <td>25</td> <td>SAS: NOM-EMBARC</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-DESTIN</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-ENTRA</td> </tr> </table> <p>Elaborar rep de ing. por atr Reporte Mensual de Ingresos</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%;"></td> <td style="width: 10%;"></td> <td style="width: 60%;">Frame de encabezado</td> </tr> <tr> <td></td> <td></td> <td>Frame de detalle</td> </tr> <tr> <td>Number</td> <td>3</td> <td></td> </tr> <tr> <td>Z</td> <td>8</td> <td>2</td> </tr> <tr> <td>Alpha</td> <td>20</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Frame de totales</td> </tr> <tr> <td>Z</td> <td>10</td> <td>2</td> </tr> </table> <p>Entregar reporte Policia Fed Reporte Quin. para Policia Fed</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%;"></td> <td style="width: 10%;"></td> <td style="width: 60%;">ENCABEZADO</td> </tr> <tr> <td>Date</td> <td>6</td> <td></td> </tr> <tr> <td>Number</td> <td>2</td> <td></td> </tr> <tr> <td></td> <td></td> <td>DETALLE</td> </tr> <tr> <td>Date</td> <td>6</td> <td></td> </tr> <tr> <td>Alpha</td> <td>3</td> <td>SAS: PAIS</td> </tr> <tr> <td>Alpha</td> <td>20</td> <td>SAS: NOM-CAP</td> </tr> <tr> <td>Alpha</td> <td>25</td> <td>SAS: NOM-EMBARC</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-DESTIN</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-ENTRA</td> </tr> <tr> <td></td> <td></td> <td>EXTRACT</td> </tr> <tr> <td>Date</td> <td>6</td> <td>SAS: FEC-ARRIBO</td> </tr> <tr> <td>Alpha</td> <td>3</td> <td>SAS: PAIS</td> </tr> <tr> <td>Alpha</td> <td>20</td> <td>SAS: NOM-CAP</td> </tr> <tr> <td>Alpha</td> <td>25</td> <td>SAS: NOM-EMBARC</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-DESTIN</td> </tr> <tr> <td>Alpha</td> <td>10</td> <td>SAS: PTO-ENTRA</td> </tr> </table> <p>Prestación de Servicios Actualizar tarifas de serv Catálogo de servicios</p>	Alpha	10	SAS: SLIP	Totales			Number	3		Number	5		Extracción			Number	8	SAS: CVE-EMBAR	Number	3	SAS: LOA	Alpha	25	SAS: NOM-EMBARC	Alpha	10	SAS: SLIP			Encabezado	Date	6		Number	2				Detalle	Date	6	SAS: FEC-ARRIBO	Alpha	3	SAS: PAIS	Alpha	20	SAS: NOM-CAP	Alpha	25	SAS: NOM-EMBARC	Alpha	10	SAS: PTO-DESTIN	Alpha	10	SAS: PTO-ENTRA			Extract	Date	6	SAS: FEC-ARRIBO	Alpha	3	SAS: PAIS	Alpha	20	SAS: NOM-CAP	Alpha	25	SAS: NOM-EMBARC	Alpha	10	SAS: PTO-DESTIN	Alpha	10	SAS: PTO-ENTRA			Frame de encabezado			Frame de detalle	Number	3		Z	8	2	Alpha	20				Frame de totales	Z	10	2			ENCABEZADO	Date	6		Number	2				DETALLE	Date	6		Alpha	3	SAS: PAIS	Alpha	20	SAS: NOM-CAP	Alpha	25	SAS: NOM-EMBARC	Alpha	10	SAS: PTO-DESTIN	Alpha	10	SAS: PTO-ENTRA			EXTRACT	Date	6	SAS: FEC-ARRIBO	Alpha	3	SAS: PAIS	Alpha	20	SAS: NOM-CAP	Alpha	25	SAS: NOM-EMBARC	Alpha	10	SAS: PTO-DESTIN	Alpha	10	SAS: PTO-ENTRA
Alpha	10	SAS: SLIP																																																																																																																																																					
Totales																																																																																																																																																							
Number	3																																																																																																																																																						
Number	5																																																																																																																																																						
Extracción																																																																																																																																																							
Number	8	SAS: CVE-EMBAR																																																																																																																																																					
Number	3	SAS: LOA																																																																																																																																																					
Alpha	25	SAS: NOM-EMBARC																																																																																																																																																					
Alpha	10	SAS: SLIP																																																																																																																																																					
		Encabezado																																																																																																																																																					
Date	6																																																																																																																																																						
Number	2																																																																																																																																																						
		Detalle																																																																																																																																																					
Date	6	SAS: FEC-ARRIBO																																																																																																																																																					
Alpha	3	SAS: PAIS																																																																																																																																																					
Alpha	20	SAS: NOM-CAP																																																																																																																																																					
Alpha	25	SAS: NOM-EMBARC																																																																																																																																																					
Alpha	10	SAS: PTO-DESTIN																																																																																																																																																					
Alpha	10	SAS: PTO-ENTRA																																																																																																																																																					
		Extract																																																																																																																																																					
Date	6	SAS: FEC-ARRIBO																																																																																																																																																					
Alpha	3	SAS: PAIS																																																																																																																																																					
Alpha	20	SAS: NOM-CAP																																																																																																																																																					
Alpha	25	SAS: NOM-EMBARC																																																																																																																																																					
Alpha	10	SAS: PTO-DESTIN																																																																																																																																																					
Alpha	10	SAS: PTO-ENTRA																																																																																																																																																					
		Frame de encabezado																																																																																																																																																					
		Frame de detalle																																																																																																																																																					
Number	3																																																																																																																																																						
Z	8	2																																																																																																																																																					
Alpha	20																																																																																																																																																						
		Frame de totales																																																																																																																																																					
Z	10	2																																																																																																																																																					
		ENCABEZADO																																																																																																																																																					
Date	6																																																																																																																																																						
Number	2																																																																																																																																																						
		DETALLE																																																																																																																																																					
Date	6																																																																																																																																																						
Alpha	3	SAS: PAIS																																																																																																																																																					
Alpha	20	SAS: NOM-CAP																																																																																																																																																					
Alpha	25	SAS: NOM-EMBARC																																																																																																																																																					
Alpha	10	SAS: PTO-DESTIN																																																																																																																																																					
Alpha	10	SAS: PTO-ENTRA																																																																																																																																																					
		EXTRACT																																																																																																																																																					
Date	6	SAS: FEC-ARRIBO																																																																																																																																																					
Alpha	3	SAS: PAIS																																																																																																																																																					
Alpha	20	SAS: NOM-CAP																																																																																																																																																					
Alpha	25	SAS: NOM-EMBARC																																																																																																																																																					
Alpha	10	SAS: PTO-DESTIN																																																																																																																																																					
Alpha	10	SAS: PTO-ENTRA																																																																																																																																																					

### MARINA Directorio del Modelo

<input type="checkbox"/> CVE-SERV	DEFAULT	Number	3	Clave Servicio
<input type="checkbox"/> DESCRIP	DEFAULT	Alpha	20	Descripción
<input type="checkbox"/> SIG-PANT	INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> TARIFA	DEFAULT	Number	5	2 Tarifa
<b>A</b> <input type="checkbox"/> FACTURAR				
<input type="checkbox"/> PRIFA	Input			Proceso que genera reportes
<input type="checkbox"/> PRMFA	I-O			Proceso para factura
<input type="checkbox"/> CVE-EMBAR	DEFAULT	Number	8	Cve embarcación
<input type="checkbox"/> FEC-SALIDA	OUTPUT	Date	6	Fecha de salida
<input type="checkbox"/> FOLIO	DEFAULT	Number	5	Folio
<input type="checkbox"/> NOM-FACT	DEFAULT	Alpha	20	Nombre
<input type="checkbox"/> RFC	DEFAULT	Alpha	13	R.F.C.
<input type="checkbox"/> SIG-PANT	INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> TOTAL	OUTPUT	Number	10	2 Total
<b>R</b> <input type="checkbox"/> REPEFAC				Reporte para generar factura
<input type="checkbox"/> REPEFAC/00				
<input type="checkbox"/> REPEFAC/05				Encabezado
<input type="checkbox"/> CAMEM.NOM-EMBARC		Alpha	25	Nom. embarcación
<input type="checkbox"/> PRMFA CVE-EMBAR		Number	8	Cve embarcación
<input type="checkbox"/> PRMFA.FEC-SALIDA		Date	6	Fecha de salida
<input type="checkbox"/> PRMFA.NOM-FACT		Alpha	20	Nombre
<input type="checkbox"/> PRMFA RFC		Alpha	13	R.F.C.
<input type="checkbox"/> REPEFAC/10				Detalle
<input type="checkbox"/> F10-CANT		Number	3	
<input type="checkbox"/> F10-IMP		Z.	8	2
<input type="checkbox"/> F10-SERV		Alpha	20	SAS, DESCRIP
<input type="checkbox"/> REPEFAC/15				Totales
<input type="checkbox"/> F15-IVA		\$	7	2
<input type="checkbox"/> F15-SUBT		\$	10	2
<input type="checkbox"/> F15-TOTAL		\$	10	2
<b>P</b> <input type="checkbox"/> PPRMFA01				PF PRMFA por folio y eve-emb
<input type="checkbox"/> FOLIO				1 Ascending
<input type="checkbox"/> CVE-EMBAR				2 Ascending
<b>P</b> <input type="checkbox"/> PPRMFA02				PF PRMFA por eve-emb,fee-sal
<input type="checkbox"/> CVE-EMBAR				1 Ascending
<input type="checkbox"/> FEC-SALIDA				2 Ascending
<b>A</b> <input type="checkbox"/> INF-SERVICIOS				Informar serv. de La Marina
<input type="checkbox"/> PRCSSE	Input			Consulta de servicios
<input type="checkbox"/> CVE-SERV	DEFAULT	Number	3	Clave Servicio
<input type="checkbox"/> DESC	DEFAULT	Alpha	20	SAS, DESCRIP
<input type="checkbox"/> SERV	DEFAULT	Number	3	SAS, CVE-SERV
<input type="checkbox"/> SIG-PANT	DEFAULT	Alpha	5	SIG PANTALLA
<input type="checkbox"/> TARIFA	DEFAULT	Number	5	2 Tarifa
<b>A</b> <input type="checkbox"/> REG-SERVICIOS				Registrar servicios
<input type="checkbox"/> MENSE	Input			Menú de servicios
<input type="checkbox"/> OP-MENU	DEFAULT	Number	1	Operación del Menú
<input type="checkbox"/> SIG-PANT	INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> PROCEC	Input			Elaborar edo cuenta a cliente
<input type="checkbox"/> CVE-EMBAR	DEFAULT	Number	8	Cve embarcación
<input type="checkbox"/> SIG-PANT	INPUT	Alpha	5	SIG-PANTALLA
<b>E</b> <input type="checkbox"/> PRMTR	I-O			Proceso registro transacciones
<input type="checkbox"/> CVE-EMBAR	DEFAULT	Number	8	Cve embarcación
<input type="checkbox"/> CVE-SERV	DEFAULT	Number	3	Clave Servicio
<input type="checkbox"/> FEC-DIA	DEFAULT	Date	6	Fecha Servicio
<input type="checkbox"/> MONTO	DEFAULT	Number	8	2 Monto
<input type="checkbox"/> PAGADO	DEFAULT	Number	1	Pagado
<input type="checkbox"/> SIG-PANT	INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> SLIP	DEFAULT	Alpha	10	Slip
<b>P</b> <input type="checkbox"/> PPRMTR01				PF PRMTR fecha,slip y eve-serv
<input type="checkbox"/> FEC-DIA				1 Ascending
<input type="checkbox"/> SLIP				2 Ascending
<input type="checkbox"/> CVE-SERV				3 Ascending
				Fecha para el reg. del serv
				Slip
				Clave del servicio

**MARINA Directorio del Modelo**

<input type="checkbox"/> PFPRMTR02		PF PRMTR por fecha,eve-serv			
<input type="checkbox"/> FEC-DIA		1 Ascending			Fecha para el reg. del serv
<input type="checkbox"/> CVE-SERV		2 Ascending			Clave del servicio
<input type="checkbox"/> PFPRMTR03		PF PRMTR por eve-emb y fecha			
<input type="checkbox"/> CVE-EMBAR		1 Ascending			Clave de la embarcación
<input type="checkbox"/> FEC-DIA		2 Ascending			Fecha para el reg. del serv
<input type="checkbox"/> MENGR	Input	Menú general del sistema			
<input type="checkbox"/> OP-MENU		DEFAULT	Number	1	Opción del Menú
<input type="checkbox"/> SIG-PANT		INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> MENRE	Input	Menú de Reportes			
<input type="checkbox"/> OP-MENU		DEFAULT	Number	1	Opción del Menú
<input type="checkbox"/> SIG-PANT		INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> EVENT	Output	Default Event			
<input type="checkbox"/> CVE-EMBAR		OUTPUT	Number	8	Cve embarcación
<input type="checkbox"/> CVE-SERV		OUTPUT	Number	3	Clave Servicio
<input type="checkbox"/> FEC-DIA		OUTPUT	Date	6	Fecha Servicio
<input type="checkbox"/> MDNTO		OUTPUT	Number	8	2 Monto
<input type="checkbox"/> PAGADO		OUTPUT	Number	1	Pagado
<input type="checkbox"/> SLIP		OUTPUT	Alpha	10	Slip
<input type="checkbox"/> GL-NAVEGA		Insertable			Lógica global de navegación
<input type="checkbox"/> SIG-PANT		INPUT	Alpha	5	SIG-PANTALLA
<input type="checkbox"/> GL-NAVEGACP		Insertable			Lógica gl. de navega copy from
<input type="checkbox"/> SIG-PANT		DEFAULT	Alpha	5	SIG PANTALLA
<input type="checkbox"/> GL-OPMENU		Insertable			Display de opción
<input type="checkbox"/> OP-MENU		DEFAULT	Number	1	Opción del Menú
<input type="checkbox"/> GL-TITULO		Insertable			Lógica global de titulos

#### **5.2.4 REGLAS DEL NEGOCIO**

En este punto, se definen todas las reglas del negocio asociadas a cada uno de los objetos que conforman el modelo LINC, a continuación se presentan como ejemplo las reglas del negocio para los segmentos de: controla y embarcaciones. También se incluyen ejemplos de dos pantallas que consideramos las más representativas del sistema: Catálogo de Embarcaciones (CAMEM) y Registro de Transacciones (PRMTR).

## **Reglas de Negocio de CONTROLA**

1. Se divide en dos áreas funcionales:

### **EMBARCACIONES**

En embarcaciones se involucran las actividades relacionadas con el registro, asignación de slips y generación de reportes correspondientes a la Policía Federal de Caminos y Capitanía de Puerto y proceso de facturación.

### **SERVICIOS**

Es el área encargada de proporcionar información sobre los servicios que proporciona La Marina, también se encarga de la actualización de las tarifas correspondientes y del registro de los servicios consumidos por cada embarcación durante su estancia.

2. Las Principales actividades del Negocio, se puede enlistar de la siguiente forma:

- Registrar Embarcaciones.
- Asignar Slips.
- Facturar.
- Generar Estados de Cuenta del Cliente.
- Registrar Transacciones.
- Actualización de Tarifas.
- Generar reportes para la Capitanía de Puerto y Policía Federal de Caminos.
- Generar reportes de Ingresos.
- Desplegar información de servicios que se ofrecen.

### **(CONTROLA) Reglas de Negocio de EMBARCACIONES**

1. Al momento en que llega una embarcación a La Marina, lo primero que se tiene que hacer es registrarla, por lo cual es necesario capturar los datos generales requeridos.
2. Con la información obtenida en el punto anterior, se debe asignar el slip correspondiente a la embarcación, de acuerdo a sus medidas de LOA y características generales.
3. Cuando el cliente lo requiera puede solicitar un estado de cuenta que le informe sobre sus gastos y consumos a la fecha.
4. Otra de las actividades asignadas a esta área funcional, es la correspondiente a la generación de la factura, la cual se realiza cuando el cliente decide cerrar su cuenta. Este proceso ejecuta a su vez la orden de salida para que la embarcación pueda abandonar las instalaciones de La Marina.
5. Con la información capturada en el punto uno, se generan los reportes que La Marina tiene que entregar a la Policía Federal de Caminos y a la Capitán de Puertos para cumplir con las obligaciones que le marca la ley.
6. Se entrega a Contabilidad mensualmente un reporte de ingresos donde se reflejan los rubros de los servicios que le generaron recursos monetarios a La Marina.

**TESIS SIN PAGINACION**

**COMPLETA LA INFORMACION**

(CONTROLA) Pantalla de CAMEM

CAMEM0000095JUL02 #ACTM><MAI>

• IR A <SIG-P>

• OPERADORA NAUTICA LA MARINA  
• SISTEMA DE CONTROL DE EMBARCACIONES Y SERVICIOS

• CATALOGO DE EMBARCACIONES

•

---

CLAVE EMBARCACION <CVE-EMBA>    NOMBRE EMBARCACION <NOM-EMBARC    >  
L.O.A. <LOA>                    BANDERA <BAN>                    CAPITAN <NOM-CAP                    >  
FECHA ARRIBO #FEC-AR>    POSIBLE SALIDA <FEC-SA>    SALIDA REAL #FEC-RS>  
SEGURO <NUM-SEGURO>                    DESCUENTO <DE>                    PUERTO ENTRADA <PTO-ENTRA >  
PUERTO DESTINO <PTO-DESTIN>                    SLIP <SLIP                    >  
PROPIETARIO <PROPIET                    >    DIRECCION <DIRECCION                    >  
CIUDAD <CIUDAD                    >    PAIS <PAI>                    TELEFONO <TELEFONO >  
C.P. <COD-P>    FAX <FAX                    >    TARJETA CREDITO <NUM-TC                    >

•

---



(CONTROLA) Pantalla de PRMTR

PRMTRT00000095JUL02 #ACTM>

•IR A <SIG-P>

•OPERADORA NAUTICA LA MARINA  
•SISTEMA DE CONTROL DE EMBARCACIONES Y SERVICIOS  
•REGISTRO DE TRANSACCIONES

FECHA #FEC-DI>  
CLAVE EMBARCACION <CVE-EMBA>  
CLAVE SERVICIO <CVE>  
MONTO POR SERVICIOS <MONTO .>  
PAGADO #P>  
SLIP <SLIP >

### **5.2.5 DICCIONARIO DE DATOS**

Aprovechando las facilidades de LDA, definimos los datos en el diccionario agregando cada uno vía la definición de pantallas de cada objeto. El diccionario se encuentra incluido dentro del repositorio definido para el sistema en LDAIII.

A continuación presentamos el diccionario de datos, como salida de LDA.

**Diccionario de Datos Local CONTROLA**

Name	Edti	Len	Dec	Description
BANDERA	Alpha	3		Bandera de la embarcación
CIUDAD	Alpha	15		Ciudad de la dir. del prop.
COD-POST	Alpha	5		Código Postal
CONSECUTIV	Number	8		Número consecutivo
CVE-EMBAR	Number	8		Clave de la embarcación
CVE-SERV	Number	3		Clave del servicio
DESCRIP	Alpha	20		Descripción del servicio
DESCUENTO	Number	2		Descuento de atraque
DIRECCION	Alpha	30		Dirección del propietario
FAX	Alpha	10		Número de fax del prop.
FEC-ARRIBO	Date	6		Fecha de arribo
FEC-ASIGSL	Date	6		Fecha de asignación slip
FEC-DIA	Date	6		Fecha para el reg. del serv.
FEC-RSAL	Date	6		Fecha real de salida
FEC-SALIDA	Date	6		Fecha de salida
FOLIO	Number	5		Folio de la Factura
LOA	Number	3		Longitud de area
MONTO	Number	8	2	Monto del servicio
NOM-CAP	Alpha	20		Nombre Capitán
NOM-EMBARC	Alpha	25		Nombre de la embarcación
NOM-FACT	Alpha	20		Nombre a quien factura
NUM-SEGURO	Alpha	10		Número de seguro
NUM-TC	Alpha	16		Número de tarjeta de crédito
OCUPADO	Number	1		Lógico de ocupación
OP-MENU	Number	1		Opción del Menú
PAGADO	Number	1		Lógico servicio pagado
PAIS	Alpha	3		País del propietario
PROPIET	Alpha	20		Propietario
PTO-DESTIN	Alpha	10		Puerto destino
PTO-ENTRA	Alpha	10		Puerto de entrada
RFC	Alpha	13		RFC de quien se factura
SIG-PANT	Alpha	5		Siguiente Pamalla
SLIP	Alpha	10		Slip
TARIFA	Number	5	2	Tarifa del servicio
TELEFONO	Alpha	10		Teléfono del propietario
TIP-EMBARC	Number	2		Tipo de Embarcación
TOTAL	Number	10	2	Total facturado

### **5.2.6 RELACIONES ENTRE OBJETOS**

Las siguientes gráficas muestran las relaciones entre los objetos que conforman el modelo. Gracias al uso de una herramienta CASE como LDA se definen y se validan automáticamente estas relaciones; permitiendo así una clarificación del análisis.

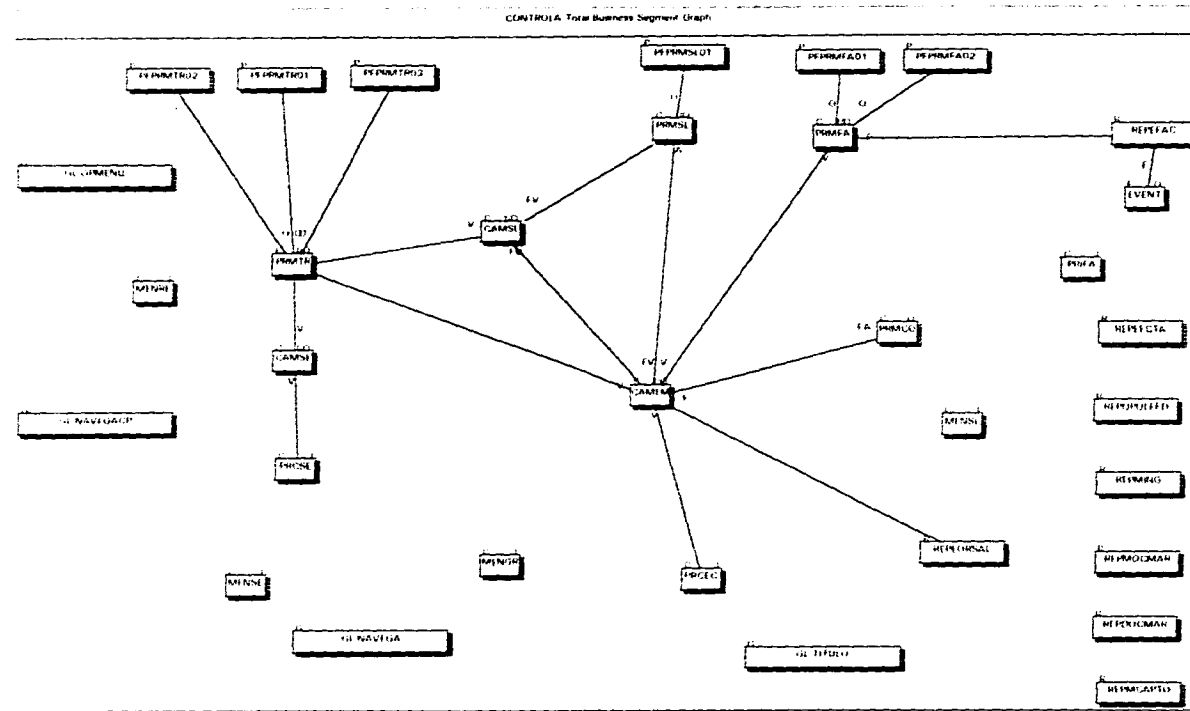
Las gráficas presentadas son:

- Purges, Recalls
- Autos, Flags, Overs, Validates
- Inserts
- References



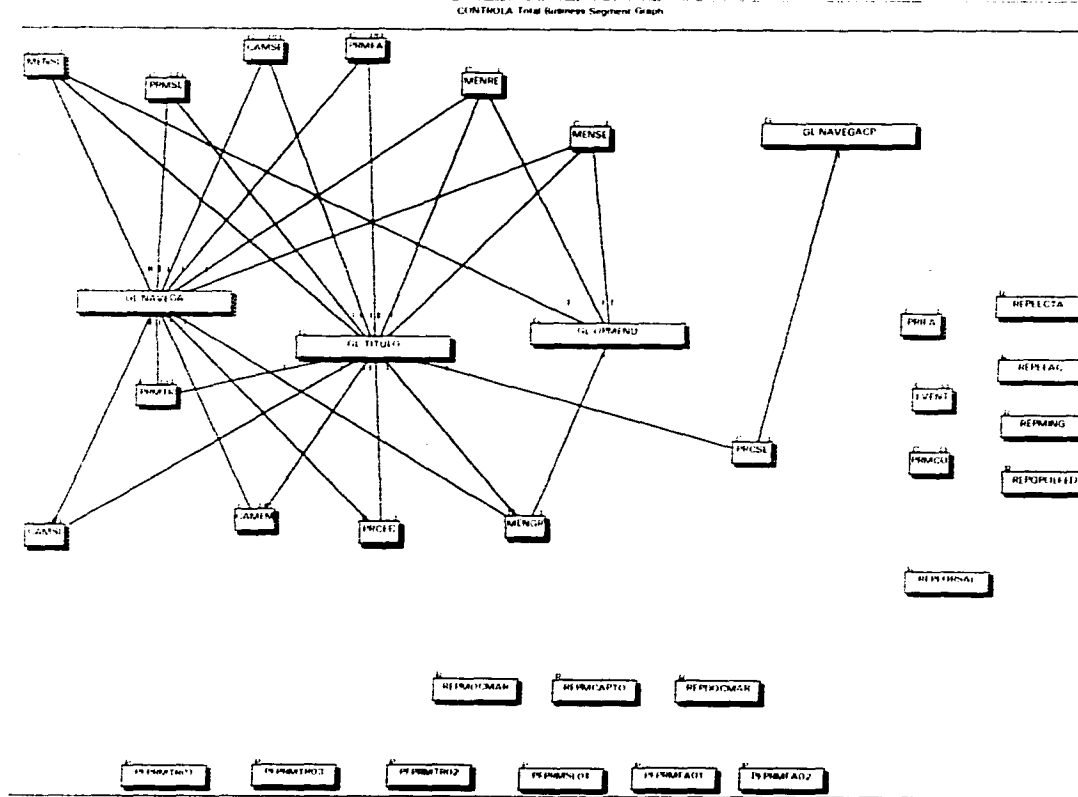
# Gráfica Total del Segmento de Negocio CONTROLA

Autos Flags Overs Validates



### Gráfica Total del Segmento de Negocio CONTROLA

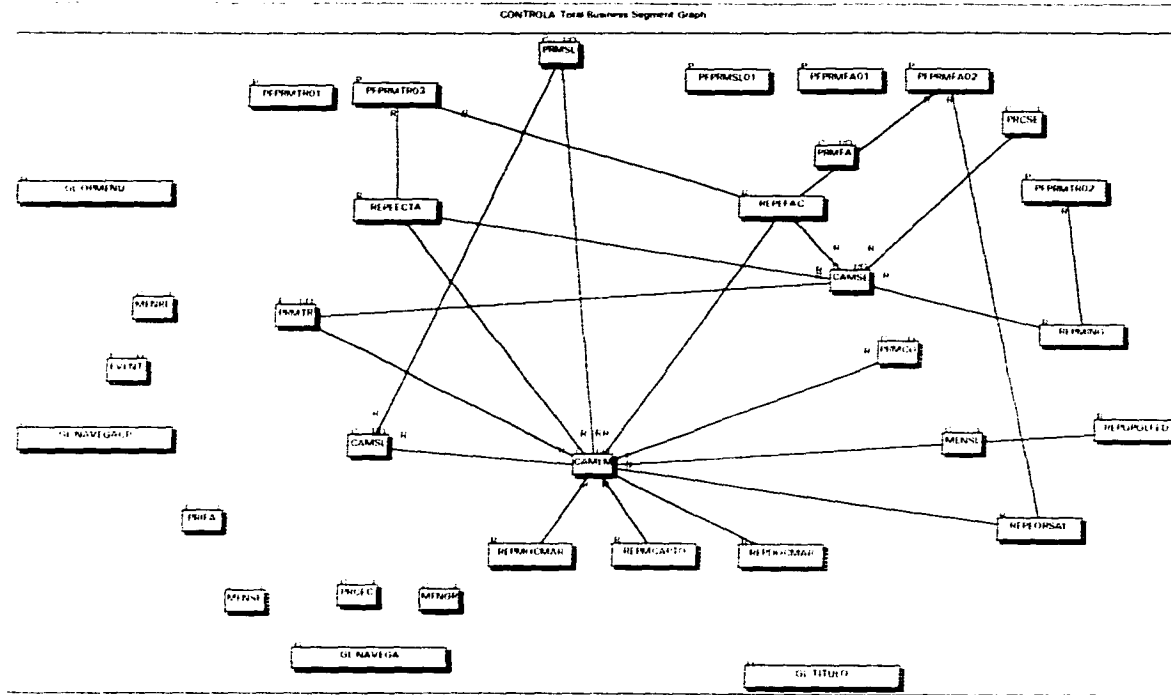
Inserts



LDA III 1.1

# Gráfica Total del Segmento de Negocio CONTROLA

References



LDA III I.1



### **5.2.7 REVISION DEL DISEÑO**

En esta tarea se hace una revisión del modelo como un todo para asegurar que exista consistencia y que se hayan alcanzado los objetivos iniciales del sistema. La herramienta CASE utilizada tiene la utilidad de Auditoría de Diseño (Design Audit) la cual se ejecuta contra el modelo, proporcionando así todos los elementos que indiquen un diseño pobre.

A continuación se muestra el resultado de la Auditoría de Diseño que se aplicó al sistema.

## Auditoría de Diseño para el Segmento de Negocio CONTROLA

### 1 Statistics

Functional Areas	2
Activities	11
Wildcards	0
Components	13
Events	1
Profiles	6
Reports	8
Global Logics	4
Local Dictionary Data Items	37

### 2 Reserved Words

#### 2.1 A Series Reserved Words

None

#### 2.2 PC Reserved Words

None

### 3 Data Item Analysis

#### 3.1 Duplicate Data Items

None

#### 3.2 Data Items Not Used

None

#### 3.3 Draft Data Item Usage

None

### 4 Event Analysis

#### 4.1 Event Set Record Layout

Name	Edit	Len	Dec	Description
ACTMTH	Number	4		
CVE-EMBAR	Number	8		Clave de la embarcación
CVE-SERV	Number	3		Clave del servicio
FEC-DIA	Date	6		Fecha para el reg. del serv.
GLB REPORT	Alpha	1		
INPUT-DATE	Alpha	7		
ISPEC	Alpha	5		
MONTO	Number	8	2	Monto del servicio
PAGADO	Number	1		Lógico servicio pagado
SLIP	Alpha	10		Slip
TRANNO	Number	6		

Total Event Set length is 59 using 11 data items.

**Auditoría de Diseño para el Segmento de Negocio CONTROLA**

**4.2 Event Set Disk Usage**

Name	Usage	Monthly Vol	Data Items	Data Item %	Utilization %
PRMTR	I-O	1000	11	100 %	100 %

Based on Monthly Volume the Event Set is 100 % utilized

**4.3 Event Set Data Item Usage**

Event Set Data Item Usage

Name	Event Count	Total Record %
ACTMTH	1	100 %
CVE-EMBAR	1	100 %
CVE-SERV	1	100 %
FEC-DIA	1	100 %
GLB.REPORT	1	100 %
INPUT-DATE	1	100 %
ISPEC	1	100 %
MONTO	1	100 %
PAGADO	1	100 %
SLIP	1	100 %
TRANNO	1	100 %

**5 Object Usage Analysis**

**5.1 Orphan Objects**

Name	Type	Description
GL-NAVEGA	Global Logic	Lógica global de navegación
GL-NAVEGACP	Global Logic	Lógica gl. de navega copy from
GL-OPMENU	Global Logic	Display de opción
GL-TITULO	Global Logic	Logica global de titulos
MENGR	Component	Menú general del sistema
MENRE	Component	Menú de Reportes

**5.2 Specs Not Referenced**

None

**5.3 Specs Which Do Nothing**

None

**5.4 Specs With No Data Items**

Name	Type	Usage	Description
PRIFA	Component	Input	Proceso que inicia reportes

**Auditoría de Diseño para el Segmento de Negocio CONTROLA**

5.5 Specs That Could Become Usage Input

None

**6 Profile Analysis**

6.1 Profiles Never Referenced

None

6.2 Profiles With No Ordinates

None

6.3 Profiles Only in Reports

Name	Ords	Description
PFPRMFA02	2	PF PRMFA por eye-emb.fcc-sal
PFPRMTR02	2	PF PRMTR por fecha.eye-serv
PFPRMTR03	2	PF PRMTR por eye-emb y fecha

6.4 Profile Rationalization

None

6.5 Profiles With Duplicates Allowed

Name	Ords	Description
PFPRMTR01	3	PF PRMTR fecha.slip y eye-serv
PFPRMTR02	2	PF PRMTR por fecha.eye-serv
PFPRMTR03	2	PF PRMTR por eye-emb y fecha

**7 LINC Name Validation**

None

### 5.3 ATRIBUTOS TECNICOS

En este punto se define el ambiente de hardware y software en los cuales se va a implantar el sistema, para ello es necesario considerar los requerimientos funcionales, técnicos y operativos.

Las posibles plataformas en las que se puede implantar el sistema son las siguientes:

DESARROLLO LINCII	AMBIENTE DE RUN TIME
Series A	Series A
	Series V
	TPE 1100/2200
	NPE 1100/2200
	Series Sistema 80
	Series U
Series 1100/2200	TPE 1100/2200
	NPE 1100/2200
	Series U
Series U	Series U
Series V	Series A
	Series V

Dependiendo del hardware que se elija se determinará el software ambiental como puede ser: sistema operativo, manejador de base de datos y el software propio de LINC para poder ejecutar el sistema (LINC Run Time).

### 5.4 TAMAÑO Y COMPLEJIDAD DEL SISTEMA

Para esta tarea se realiza una estimación con el fin de determinar el tamaño y la complejidad del sistema, para ello se utiliza la técnica de Puntos de Función que se explicó en el capítulo tres.

LDA, también permite la generación automática de los Puntos de Función, a partir de los valores que se le den a los pesos específicos en el punto de procesamiento complejo; esto implica dar determinado valor a los siguientes factores:

- Comunicación de Datos
- Procesamiento Distribuido
- Objetivos de Desempeño (Performance)
- Configuración
- Alto rango de transacciones
- Consulta y captura en línea
- Eficiencia del usuario final
- Actualización en línea
- Procesamiento complejo
- Código re-utilizable
- Facilidades de conversión e instalación
- Facilidades operativas
- Instalación en múltiples sitios
- Facilidad de cambio

A continuación se presenta la salida de Puntos de Función del sistema, utilizando LDA, en la cual se obtuvo como resultado 261 Puntos Ajustados con los cuales se determinó la complejidad del sistema.

**Análisis de Puntos por Función para el Segmento de Negocio CONTROLA**

**1 Specs**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
CAMEM	4	5	22	0	6	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0	0
CAMSE	0	0	4	0	6	0	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
CAMSL	0	0	4	0	6	0	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
MENGR	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
MENRE	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
MENSE	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
MENSL	0	0	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PRCEC	0	1	0	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PRCSE	0	2	0	0	10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
PRIFA	0	0	11	0	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PRMCO	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PRMFA	0	1	5	2	6	0	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
PRMSL	2	4	4	0	6	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0	0
PRMTR	0	5	10	1	2	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0

**2 Reports**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
REPDOCMAR	0	1	0	0	0	12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REPECTA	0	3	0	0	0	6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
REPEFAC	2	4	0	0	0	11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
REPEORSAL	1	2	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REPMCAPTO	0	1	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REPMING	0	2	0	0	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REPMOCMAR	0	1	0	0	0	12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REPQPOLFED	0	1	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

**3 Profiles**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
PFRMFA01	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

**Análisis de Puntos por Función para el Segmento de Negocio CONTROLA**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
PFPRMFA02	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PFPRMSL01	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PFPRMTR01	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PFPRMTR02	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PFPRMTR03	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

**4 Wildcards**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	



## Analisis de Puntos por Función para el Segmento de Negocio CONTROLA

### 5 Totals

#### 5.1 External Inputs

Simple	10	*	3	=	30
Average	0	*	4	=	0
Complex	7	*	6	=	42
Total	17				72

#### 5.2 External Outputs

Simple	6	*	4	=	24
Average	1	*	5	=	5
Complex	1	*	7	=	7
Total	8				36

#### 5.3 External Inquires

Simple	9	*	3	=	27
Average	0	*	4	=	0
Complex	3	*	6	=	18
Total	12				45

#### 5.4 Logical Master Files

Simple	13	*	7	=	91
Average	0	*	10	=	0
Complex	0	*	15	=	0
Total	13				91

**Análisis de Puntos por Función para el Segmento de Negocio CONTROLA**

**5.5 Interfaces to Other Systems**

Simple	0	*	5	=	0
Average	0	*	7	=	0
Complex	0	*	10	=	0
Total	0				0

**Análisis de Puntos por Función para el Segmento de Negocio CONTROLA**

**6 Processing Complexity**

1	Data Communications	2
2	Distributed Data or Processing	1
3	Performance Objectives	3
4	Tight Configuration	3
5	High Transaction Rate	4
6	On-line Inquiry/Data Entry	4
7	End User Efficiency	4
8	On-line Update	5
9	Complex Processing	2
10	Code Reusability	2
11	Conversion/Installation Ease	3
12	Operational Ease	4
13	Multiple Site Installation	1
14	Facilitate Change	4

**7 Totals**

**Total Unadjusted Function points are 244**

**Total Degree of Influence is 42**

**Adjustment Factor is  $(0.65 + (0.01 * 42)) = 1.07$**

**Final Function Point Count is 261**

## CAPITULO VI DESARROLLO DEL SISTEMA

En este capítulo se presenta la tercera fase de la metodología LSA: *Desarrollo del Sistema*, su objetivo principal es terminar el desarrollo del Sistema de Información, para ello se retoma el prototipo evolutivo de la fase de *Definición del Sistema*. Esto incluye llevar a cabo las siguientes tareas: completar el desarrollo de las rutinas de Lógicas Globales, desarrollar la lógica para eventos, componentes, profiles y reportes, creación de la pantalla de inicio a la aplicación junto con la lógica de seguridad en caso de ser necesario, así como el proceso de salida del sistema y completar la documentación en base a las nuevas características del sistema especificadas en esta fase.

Para este punto se continua utilizando la herramienta LDA; ya que a pesar de ser Upper CASE tiene aplicabilidad en esta etapa que corresponde al Lower CASE.

La salida final que se obtiene es un prototipo depurado que en última instancia permite continuar con la etapa de implantación, migrando el sistema al ambiente de desarrollo LINC en un mainframe.

### 6.1 CONVERSION DEL MODELO DE OBJETOS A ESPECIFICACIONES DE LINC

En este punto se muestran como salida de LDA algunas lógicas desarrolladas en esta etapa; también se muestra la impresión de un ejemplo de una factura, obtenida a través del sistema, el Estado de Cuenta del Cliente y el menú principal en el cual están las rutinas de acceso de la aplicación.

(CONTROLA) Lógica Principal REPEFAC/00

```
00001 DT: LAST PPRMFA02 (G-NINE2 G-NINE1)
00002
00003 DT: FROM PPRMTR03 (PRMFA.CVE-EMBAR GLB.ZEROS)
00004 DW: PRMFA.CVE-EMBAR NOT = EVENT.CVE-EMBAR
00005 ME: ATTENTION (NO HAY DATOS)
00006 EE:
00007 MV: EVENT.CVE-SERV SD-CVESERV
00008 BK:
00009 END:
00010
00011 DW: GLB.STATUS NOT = GLB.SPACES
00012 ME: ATTENTION (NO HAY REGISTROS)
00013 EE:
00014
00015 BP: 05
00016 ADVANCE: (5)
00017
00018 DT: FROM PPRMTR03 (PRMFA.CVE-EMBAR GLB.ZEROS)
00019 DW: PRMFA.CVE-EMBAR NOT = EVENT.CVE-EMBAR
00020 BK:
00021 END:
00022 DW: EVENT.CVE-SERV = SD-CVESERV
00023 : AND
00024 : DW: EVENT.PAGADO = GLB.ZEROS
00025 ADD: EVENT.MONTO F10-IMP
00026 ADD: (1) F10-CANT
00027 ADD: EVENT.MONTO F15-TOTAL
00028 FL: (1) EVENT.PAGADO
00029 MV: (1) SD-BAND
00030 END:
00031 DW: EVENT.CVE-SERV NOT = SD-CVESERV
00032 : AND
00033 : DW: EVENT.PAGADO = GLB.ZEROS
00034 PF: 10
00035 MV: GLB.ZEROS F10-IMP
00036 MV: GLB.ZEROS F10-CANT
00037 MV: EVENT.CVE-SERV SD-CVESERV
00038 ADD: EVENT.MONTO F10-IMP
00039 ADD: (1) F10-CANT
00040 ADD: EVENT.MONTO F15-TOTAL
00041 FL: (1) EVENT.PAGADO
00042
00043 END:
00044
00045 END:
00046
00047 DW: SD-BAND = (1)
00048 PF: 10
00049 END:
00050
00051 PF: 15
00052
00053 FL: F15-TOTAL PRMFA.TOTAL
```

(CONTROLA) Lógica Pre-LINC PRMTR

00001  
00002 : \*\*\*\*\* PRE LINC PRMTR  
00003 INS; GL-NAVEGA  
00004  
00005 DW; SIG-PANT = GLB.SPACES  
00006  
00007 DW; CVE-EMBAR = GLB.ZEROS OR  
00008 DW; CVE-SERV = GLB.ZEROS  
00009 ME; ATTENTION (CLAVE DE EMBARCACION Y SERVICIO REQUERIDAS)  
00010 CU; CVE-EMBAR  
00011 MV; (\*\*\*\*\* GLB.ERROR  
00012 EE;  
00013  
00014 LU; FROM CVE-EMBAR (CAMEM)  
00015 DW; CAMEM.CVE-EMBAR = CVE-EMBAR  
00016 DW; CAMEM.FEC-RSAL NOT = GLB.ZEROS  
00017 ME; ATTENTION (LA EMBARCACION YA PARTIO)  
00018 MV; (\*\*\*\*\* GLB.ERROR  
00019 EE;  
00020 MV; CAMEM.SLIP SLIP  
00021 END;  
00022 BK;  
00023 END;  
00024  
00025 DW; MONTO = GLB.ZEROS  
00026 LU; FROM CVE-SERV (CAMSE)  
00027 DW; CVE-SERV = CAMSE.CVE-SERV  
00028 MV; CAMSE.TARIFA MONTO  
00029 END;  
00030 BK;  
00031 END;  
00032 END;  
00033  
00034 END;  
00035  
00036  
00037  
00038  
00039  
00040  
00041  
00042  
00043  
00044  
00045  
00046  
00047  
00048  
00049  
00050  
00051

(CONTROLA) Lógica Pre-LINC CAMEM

```
00001
00002 : ***** PRE LINC CAMEM
00003
00004 INS: GL-NAVEGA
00005
00006 DW: MAINT = (ADD) AND
00007 DW: CVE-EMBAR NOT = GLB.ZEROS
00008 ME: ATTENTION (NO CAPTURE CVE. DE EMBARCACION PARA AGREGAR)
00009 CU: CVE-EMBAR
00010 EE:
00011
00012 :***VALIDACION DEL SLIP***:
00013 DW: MAINT = (ADD)
00014 LU: FROM SLIP (CAMSL)
00015 DW: CAMSL.SLIP NOT = SLIP
00016 ME: ATTENTION (EL SLIP NO ESTA DADO DE ALTA EN EL CATALOGO)
00017 EE:
00018 DW: CAMSL.OCUPADO NOT = GLB.ZEROS
00019 ME: ATTENTION (EL SLIP ESTA OCUPADO)
00020 CU: SLIP
00021 EE:
00022 SB: (20) CAMSL.LOA GIV: SD-I.LOA
00023 DW: LOA > CAMSL.LOA
00024 ME: ATTENTION (LONGITUD DE EMBARCACION MAYOR QUE SLIP)
00025 CU: SLIP
00026 EE:
00027 DW: LOA < SD-LOA
00028 ME: ATTENTION (LONGITUD DE EMBARCACION MENOR QUE SLIP)
00029 CU: SLIP
00030 EE:
00031 BK:
00032 END:
00033 END:
00034
00035
00036
00037 DW: MAINT = (ADD)
00038
00039 LOOK.UP: EVERY (PRMCO)
00040 BK:
00041 END:
00042
00043 DW: GLB.STATUS NOT = GLB.SPACES
00044 MV: GLB.SPACES GLB.STATUS
00045 MV: GLB.YYMMDD SD-FECHA
00046 MV: SD-INI-CONS SD-CONSEC
00047
00048 AUTO.ENTRY: PRMCO
00049 AUTO:GR-CONSEC CONSECUTIV
00050 AUTO:WRITE&CLEAR
00051
00052 MV: GR-CONSEC CVE-EMBAR
00053 EE:
00054
00055 MV: PRMCO.CONSECUTIV POS: (1) LE: (6) SD-FECHA
00056 MV: PRMCO.CONSECUTIV POS: (7) LE: (2) SD-CONSEC
```

LDA III 1.1

(CONTROLA) Lógica Pre-LINC CAMEM

```
00057
00058 DW: GLB.YYMMDD = SD-FECHA
00059   ADD: (01) SD-CONSEC
00060   END;
00061
00062 DW: GLB.YYMMDD > SD-FECHA
00063   MV: (01) SD-CONSEC
00064   MV: GLB.YYMMDD SD-FECHA
00065   END;
00066
00067   MV: GR-CONSEC SD-CONSI
00068   FL: SD-CONSI PRMCO.CONSECUTIV
00069   MV: SD-CONSI CVE-EMBAR
00070
00071   MV: GLB.YYMMDD FEC-ARRIBO
00072
00073   LU: FROM SLIP (CAMSL) SECURE
00074   DW: SLIP = CAMSL.SLIP
00075   FL: (1) CAMSL.OCUPADO
00076   BK;
00077   END;
00078   END;
00079
00080   LU: FROM SLIP (CAMSL)
00081   BK;
00082   END;
00083
00084   AUTO.ENTRY: PRMSL
00085   AUTO: CVE-EMBAR CVE-EMBAR
00086   AUTO: FEC-ARRIBO FEC-ASIGSL
00087   AUTO: SLIP SLIP
00088   AUTO: WRITE&CLEAR
00089
00090
00091   END: ** DEL DW: MAINT = ADD
00092
00093
00094   :*** VALIDACION DE CAMBIO DE SLIP ****:
00095   DW: MAINT = (CHG)
00096   LU: FROM CVE-EMBAR (CAMEM)
00097   DW: CVE-EMBAR = CAMEM.CVE-EMBAR
00098   DW: SLIP NOT = CAMEM.SLIP
00099   ME: ATTENTION (NO SE PERMITE CAMBIAR EL SLIP)
00100   EE;
00101   END;
00102   BK;
00103   END;
00104   END;
```





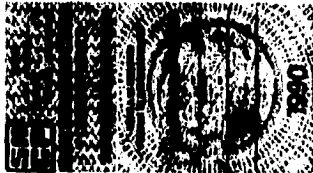
## Promotora Marina de Acapulco, S.A. de C.V.

Av. Costera Miguel Alemán #215  
Fraccionamiento Las Playas, Acapulco Gro.  
C.P. 39309

Factura N°:

FACTURADO A: ALEJANDRO GARCIA C.  
NOMBRE DE LA EMBARCACION: PACIFIC 95072903  
RFC: GACA710125810  
FECHA Y LUGAR DE EMISION: 95/07/29 ACAPULCO, MEXICO

CANT.	DESCRIPCION	TOTAL
001	ATRAQUE	9500.00
001	CHOCOLATE	100.00
SUBTOTAL		\$8347.82
IVA		\$1252.18
TOTAL		\$9600.00



SISTEMA DE CONTROL DE EMBARCACIONES Y SERVICIOS  
ESTADO DE CUENTA DEL CLIENTE

NOMBRE: MEX  
EMBARCACION: OCEAN

CANTIDAD DE SERVICIOS	NOMBRE DEL SERVICIO	IMPORTE
001	ATRAQUE	14500.00
001	CAFE	35.00
001	CHOCOLATE	100.00
	<b>TOTAL DE CONSUMO</b>	14635.00

(CONTROLA) Pantalla de MENGR

MENGR00000095JUL29 #ACTM>

• IR A <SIG-P>

•OPERADORA NAUTICA LA MARINA  
•SISTEMA DE CONTROL DE EMBARCACIONES Y SERVICIOS

•MENU PRINCIPAL

- 1. EMBARCACIONES
- 2. MENU DE REPORTES
- 3. MENU DE SERVICIOS
- 4. MENU DE SLIPS

OPCION <0>

## CONCLUSIONES

En el presente trabajo de tesis se hizo una revisión sobre los antecedentes de CASE, se explicó su filosofía, origen y definición, las metodologías en las que se puede apoyar, técnicas y herramientas que involucra, además de algunas consideraciones de índole administrativa. También se incluyó un apartado dedicado especialmente a justificar por qué se utiliza CASE en el desarrollo de sistemas de información, brindándose además una explicación de la metodología LSA y de la herramienta LDA III utilizadas en el proyecto, para finalmente detallar el análisis y diseño del sistema propuesto para el caso de La Marina de Acapulco, mostrando como resultado el prototipo obtenido.

Al trabajar con la filosofía CASE tuvimos la oportunidad de acercarnos de manera teórica y práctica a un nuevo enfoque en la construcción de sistemas. Este enfoque, que brinda todo un ambiente integrado donde se incluyen metodologías, herramientas, técnicas y otros conceptos, permitió que la búsqueda de soluciones a los requerimientos del usuario y la definición de sus necesidades se realizara con amplia participación de ellos, además de los analistas y de la alta y media gerencia.

De lo más importante en este trabajo fue comprobar e incorporar a nuestra experiencia las principales ventajas de utilizar CASE, entre las que podemos mencionar las siguientes:

En el pasado los sistemas eran analizados, diseñados, desarrollados y construidos "a mano", el ciclo de vida se volvía problemático, costoso y lento, en cambio al utilizar la filosofía CASE se acelera el proceso de desarrollo, trabajando con un ciclo de vida de sistemas automatizado que promueve una reducción de tiempo y costo, misma que se ve reflejada en el incremento de la productividad. CASE afronta el problema de productividad del software, apoyando todas las etapas del ciclo de vida de sistemas.

Esto se logra gracias al manejo de un repositorio, donde residen todos los datos involucrados con la organización, la construcción del proyecto y la

administración, lo que permite la generación automática de la información tanto para el usuario como para el equipo de desarrolladores.

Al tener un repositorio, el concepto de reutilización del software se hace práctico, ya que no sólo pueden reusarse módulos de código, sino también planes del proyecto, modelos de prototipos, de datos y especificaciones de diseño. La reutilización de los componentes del sistema, en lugar de la construcción de uno totalmente nuevo, es otra de las claves para incrementar considerablemente la productividad del software.

Uno de los puntos importantes para el desarrollo de sistemas, es poder contar con una métrica de productividad del software, en este caso utilizamos la técnica de Análisis de Puntos de Función, la cual nos permitió estimar tiempos y tamaño, lo cual resultó de gran ayuda para la administración del proyecto, tomando en cuenta que muchas veces de esto depende el éxito o fracaso del mismo.

Otra de las ventajas de CASE en su papel de filosofía, es la capacidad de coordinar todos los recursos necesarios para el desarrollo de sistemas sin importar su tamaño y complejidad.

Todo lo anterior converge en la obtención de un prototipo altamente depurado, ya que se tiene como respaldo muchas horas de planeación, lo que nos permite alcanzar al máximo los requerimientos del usuario, facilitando y reduciendo notablemente la etapa de mantenimiento que en la mayoría de las veces resulta sumamente costosa.

Esperamos que con la evolución de CASE este continúe siendo la filosofía idónea para la solución de sistemas administrativos, incrementando con ello sus potencialidades y alcances para ir al ritmo de las necesidades de un mundo cambiante.

CASE, desde su nacimiento a principios de los años 80's hasta ahora, ha ido evolucionando considerablemente; haciendo cada vez más fácil la

construcción de sistemas. Consideramos que CASE no va a quedarse estático en este punto, sino que va a continuar evolucionando y tratará en un futuro de ofrecer herramientas más poderosas, con mejor desempeño, más fáciles de usar y quizá menos costosas para poder llegar a un mayor número de empresas.

De igual manera, conforme avance la tecnología, las herramientas CASE se irán adecuando a nuevas plataformas; incluirán probablemente procesadores de lenguaje natural, sintetizadores de voz y de reconocimiento de frases, sistemas expertos, sistemas de diagnóstico y diseñadores inteligentes de software.

## **BIBLIOGRAFIA**

**ADAD, Rubén; MEDINA Miguel Angel. Fundamentos de las Estructuras de Datos Relacionales.** Grupo Noriega Editores, 1993 México. 226 p.p.

**CASE State-of-the-Practice and State-of-the-Art. Conference Proceedings,** Extended Intelligence Inc, Spring 1991.

**Manual de Organización de La Marina de Acapulco,** 1989.

**Instructivo para el Concesionamiento de Marinas Turísticas y de Puertos de Abrigo,** 1989, Puertos Mexicanos.

**MARTIN, James; McCLURE Carma. Structured Techniques: The Basis for CASE.** Edit. Prentice Hall, 1988 U.S.A..

**McCLURE, Carma. CASE is Software Automation.** Prentice Hall, 1989 U.S.A.. 290 p.p.

**MEJIA, Francisco David. VII Seminario Tecnológico: Automatización Bancaria.** Presentación Boca Ratón, Florida Grupo de Latino America y el Caribe. 1992 y 1993 México.

**QED, INFORMATION SCIENCES INC. CASE: The Potencial and the Pitfalls.** Wellesley Massachusetts, 1989 U.S.A.. 565 p.p.

- SPURR, Kathy; LAYZELL, Paul. **CASE: Current Practice/Future Prospects.** Wiley Profesional Computing, 1993 U.S.A., 246 p.p.
- TOWER, Larry E.. **CASE: Concepts and Implementation.** Edit. McGraw Hill, 1989 U.S.A., 232 p.p..
- UNISYS. **Student Guide: Function Point Analysis.**  
UNISYS Corporation, 1991 U.S.A.
- UNISYS. **LINC Sytem Approach: Managing System Development with a Flexible Methodology.** UNISYS Corporation, 1992 U.S.A.
- UNISYS. **Manual del Curso: LDA II en el Ciclo de Vida de Desarrollo de Sistemas.** Corporación UNISYS, 1992 México.
- UNISYS. **Student Guide: LINC II Information System-Orientation.**  
UNISYS Corporation, 1988 U.S.A.
- UNISYS. **LINC II: Executive Overview, The Business CASE.**  
UNISYS Corporation, 1992 U.S.A.
- UNISYS. **LDAIII: Development Refrence Manual.**  
UNISYS Corporation, 1993 U.S.A.
- UNISYS. **LDA III: Installation and Administration Guide.**  
UNISYS Corporation, 1993 U.S.A.
- UNISYS. **LINCII: Technical Overview, The Business CASE.**  
UNISYS Corporation, 1992 U.S.A.



**UNISYS. Student Guide: LINCII System Development.**  
(Volume #1). UNISYS Corporation, 1992 U.S.A.

**YOURDON Inc., Yourdon Information System Design Workshop.**  
Version 1.0, 1993 U.S.A.

**YOURDON Inc., Yourdon Information System Analysis Workshop.**  
Version 1.0, 1991 U.S.A.

**YOURDON Inc., Yourdon Systems Method.**  
1993 U.S.A., 682 p.p.