

8



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

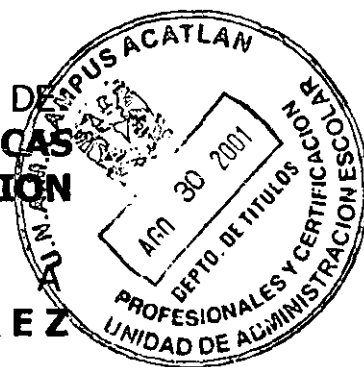
**ESCUELA NACIONAL DE ESTUDIOS  
PROFESIONALES "ACATLÁN"**

**ESTUDIO DE LA ETAPA DE ANALISIS DEL PROCEDIMIENTO  
"DESARROLLO DE APLICACIONES A LA MEDIDA", DE UNA  
EMPRESA CONSULTORA.**

**BAJO LA OPCION DE SEMINARIO TALLER EXTRACURRICULAR  
"ANALISIS DE LA PLANEACION"**

QUE PARA OBTENER EL TITULO DE  
**LICENCIADO EN MATEMATICAS**  
**APLICADAS Y COMPUTACION**

**P R E S E N T**  
**MIREYA CONTRERAS PEREZ**



ASESOR: ACT. LUZ MARIA LAVIN ALANIS

295867

SEPTIEMBRE 2001





Universidad Nacional  
Autónoma de México



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

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

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

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

# *Agradezco,*

*A Dios,*

*Por tantas Bendiciones.*

*A mis Padres;*

*Papá gracias por impulsarme a ser mejor;*

*Mamá gracias por tu gran amistad; Gracias  
por estar siempre conmigo.*

*Al Ferito,*

*Por ser mi inspiración y por demostrarme  
que el "querer" es "poder".*

*A Hueso,*

*Por enseñarme lo valioso que es la vida.*

*Al Ferote,*

*Por su amor y comprensión.*

*A Pedro, Mario y Gustavo,*

*Por su cariño.*

*A mis incondicionales amigas,*

*Maru, Rocio, Angeles, Mary,*

*Lilian, Gaby, Vicky y Lety.*

*Las llevo conmigo.*

*A los profesores de mi vida escolar,*

*Por el conocimiento transmitido.*

*A los maestros del seminario,*

*Hugo Reyes, Gerardo Roldán, Juan*

*Torres, Ignacio Lizárraga y Luz*

*María Lavín, Porque cada uno de*

*ustedes aportaron algo valioso a mi vida.*

*A esas aquellas personas que ahora no están  
conmigo, pero que en algún momento me  
motivaron a seguir adelante.*

## CONTENIDO

<b>INTRODUCCION</b> .....	I
<b>I. Análisis y Diseño de una Aplicación a la Medida</b> .....	1
<b>1.1 Definición de una Aplicación a la Medida</b> .....	2
<b>1.2 Ingeniería de Software</b> .....	2
1.2.1 Definición .....	2
1.2.2 Ciclo de Vida del Software .....	3
<b>1.3 Análisis y Diseño, una Herramienta de Construcción de Software</b> .....	4
<b>1.4 ¿Qué es ISO 9001?</b> .....	4
1.4.1 ISO 9000 en México .....	5
1.4.2 Proceso de Certificación ISO 9001.....	6
<b>1.5 Las Metodologías de Análisis de Sistemas de Información</b> .....	7
1.5.1 ¿Qué es una metodología? .....	7
1.5.2 El Modelo en sistemas de información .....	7
1.5.3 ¿Porqué una Metodología de Análisis en el Desarrollo de una Solución Informática? .....	8
1.5.4 El origen de SA/SD .....	11
1.5.5 El origen de UML .....	11
<b>1.6 ¿Porque implantar ISO-9000 en la empresa consultora?</b> .....	12
<b>1.7 Planeación Estratégica en el Estudio del procedimiento DAM</b> .....	13
<b>1.8 Situación Actual del Procedimiento Desarrollo de Aplicaciones a la Medida: Etapa de Análisis</b> .....	15
<b>1.9 Mejora Continua</b> .....	16
<b>1.10 Diagnóstico del problema</b> .....	16
1.10.1 La misión de calidad de la empresa consultora .....	17
1.10.2 Definición de componentes .....	18
1.10.3 Variables Controlables y No Controlables .....	18
<b>1.11 Diagrama de Contexto y de Flujo de Datos de la solución de la problemática actual</b> .....	19
<b>CONCLUSIONES</b> .....	23
<b>FUENTES DE CONSULTA</b> .....	24
<b>II. Consideraciones en el Análisis de las Metodologías Estructuradas "Yourdon" y Orientado a Objetos "UML"</b> .....	25
<b>2.1 Definición de Programación Estructurada</b> .....	26
<b>2.2 Definición de Programación Orientada a Objetos (OOP)</b> .....	27
<b>2.3 Características y definiciones básicas de la metodología</b> .....	29
2.3.1 Estructurada "Yourdon" .....	29
2.3.2 Orientada a Objetos "UML" .....	30
<b>2.4 Etapas y componentes del Análisis</b> .....	31
2.4.1 Metodología Estructurada "Yourdon" .....	31
2.4.2 Metodología Orientada a Objetos "UML" .....	38
<b>2.5 Seleccionando una metodología</b> .....	48
<b>CONCLUSIONES</b> .....	50
<b>FUENTES DE CONSULTA</b> .....	51

<b>III. Propuesta de Mejora a la Etapa de Análisis del Procedimiento de Desarrollo de Aplicaciones a la Medida</b> .....	53
<b>3.1 Aplicación de las metodologías en un caso práctico</b> .....	54
3.1.1 Caso práctico: Módulo de Verificación de Estructura y Carga de Información de los Archivos dinámicos que son reportados a la Comisión Nacional del SAR .....	54
3.1.2 Desarrollo de las etapas del análisis estructurado "Yourdon" .....	57
3.1.3 Desarrollo de las etapas del análisis orientado a objetos "UML" .....	94
<b>3.2 Evaluación de Resultados</b> .....	130
3.2.1 Pros y Contras de la metodología estructurada "Yourdon" .....	130
3.2.2 Pros y Contras de la metodología orientada a objetos "UML" .....	130
<b>3.3 Presentación de los cambios al procedimiento de Desarrollo de Aplicaciones a la Medida</b> .....	131
<b>3.4 Una Percepción del procedimiento DAM</b> .....	134
CONCLUSIONES .....	147
FUENTES DE CONSULTA .....	148
<b>CONCLUSIONES GENERALES</b> .....	149
<b>APENDICE</b> .....	151
<b>FUENTES DE CONSULTA GENERALES</b> .....	153

## INTRODUCCION

---

En la actualidad, existen muchas empresas que están intentando determinar como realizar su trabajo con calidad, esforzándose por llegar a una normatividad propia, tratando de obtener como primer meta la certificación de ISO 9001 y con ella seguir trabajando en la mejora continua de sus procesos. La empresa en que se apoya la presente investigación ya alcanzó esa primer meta, logrando dicha certificación en varias de sus áreas, incluyendo en la de nuestro interés: la Consultoría de Sistemas a la Medida; a la que se le asocia el procedimiento llamado "Desarrollo de Aplicaciones a la Medida (DAM)".

El DAM objetivamente establece las etapas a seguir en el desarrollo de proyectos a la medida, lo cual ha generado grandes beneficios a las personas involucradas en el desarrollo de sistemas, sin embargo, como participante de la realización efectiva de este procedimiento se han detectado algunas carencias en la etapa de análisis, específicamente al utilizar alguna de las metodologías de análisis que se encuentran especificadas en dicho procedimiento (Estructurada y UML), las fallas que se tienen son:

1. Ausencia de pasos.
2. No existen Guías de Apoyo para las actividades a realizar.
3. Es nula la existencia de ejemplos de las actividades de análisis.
4. Hay un desconocimiento, por parte del analista, de como realizar alguno o varios de los pasos por lo que requiere invertir tiempo en su estudio.
5. No hay indicadores que ayuden a la decisión de cual metodología utilizar.

Por consiguiente el no emplear una metodología correctamente durante el desarrollo del análisis y diseño repercute al momento de iniciar la construcción de la aplicación, ésta presenta inconsistencias, desfases de tiempo, malos dimensionamientos y omisiones dentro y para el desarrollo, de igual forma al momento de iniciar la etapa de pruebas, presenta carencias funcionales y de negocio, generando atrasos importantes en la entrega e implantación de la solución informática, lo que conlleva a tener un mayor gasto económico y físico.

Las carencias anteriormente mencionadas dieron como resultado la necesidad de buscar una mejora de carácter urgente al procedimiento de Desarrollo de Aplicaciones a la Medida actual, por lo que esta investigación propone una reflexión más profunda para la etapa de análisis en el procedimiento de Desarrollo de Aplicaciones a la Medida, agregando pasos y precisando los que ya estaban. ¿Cómo se logra esta mejora?, mediante el estudio de varios de los autores más relevantes de cada una de las metodologías. En el presente trabajo para el estudio de la metodología UML se eligió al autor que expresaba la metodología de manera mas amigable y concisa; y para la metodología estructurada se seleccionó los escritos de su creador, Edward Yourdon.

Al llevar acabo el análisis del problema, se hizo uso de la planeación obteniendo que la mejor manera de abordar este estudio es; en el primer capítulo se da entrada al estatus actual del DAM y el porqué del cambio en el procedimiento, también los conceptos relacionados con el tema en estudio; a continuación como segundo capítulo la exposición de las características y fases de las metodologías estructurada Yourdon y UML, que fueron utilizadas para la mejora del procedimiento DAM; y por

último en el capítulo tres se lleva a cabo la aplicación o ejecución de lo definido en el capítulo dos a un caso práctico, utilizando cada una de las metodologías en estudio, terminando con la presentación de los cambios determinados para la mejora al DAM.

Con la ayuda de la planeación se obtuvo el siguiente resultado, un procedimiento de fácil entendimiento, con formatos explícitos para cada una de las actividades a realizar, con guías de apoyo para toda persona involucrada en el desarrollo de sistemas de información, indicándole la manera de realizar alguno o varios de los pasos expuestos en el procedimiento, siendo también de gran utilidad para el principiante en el estudio de las metodologías de Análisis y Diseño de Sistemas de Información. Además se señaló las características que sobresalen de cada una de las metodologías revisadas, resaltando entre otras que: en la metodología Estructurada es de fácil entendimiento para aquellos que piensan en procesos o funciones, mientras que en UML se facilita para aquellos que analizan los componentes y objetos del problema a resolver; esto es con el fin de ayudar a la elección de cual metodología utilizar en la temprana e importante etapa del desarrollo de un Software que es la fase de ANÁLISIS.



# Capítulo I.

## ***Análisis y Diseño de una Aplicación a la Medida...***

**Objetivo:** *Definir los conceptos de: ISO 9000, modelo y metodología, presentar la situación actual de la etapa de análisis del DAM, y por último, señalar la importancia de utilizar una metodología en el análisis y diseño de sistemas de información.*

**"CUANDO LOS MEDIOS Y LOS FINES SON CLAROS,  
LAS INEFICIENCIAS SE VUELVEN MAS OBVIAS"**

**Stephen Robbins**

## 1.1. Definición de una Aplicación a la Medida...

El concepto de Aplicación a la Medida está definido como el programa o conjunto de programas concebidos para la realización de una o varias tareas que son requeridas para generar una ayuda o solución computacional de procesos o actividades específicas; es decir es un software especializado. Y el referirse al desarrollo de una aplicación a la medida, se habla del ciclo que se lleva a cabo desde el momento en que se requiere la creación de una aplicación a la medida hasta su puesta en marcha (área conocida como Ingeniería de Software).

Una aplicación a la medida es relacionada con los conceptos de Software, Sistema de Información y Solución informática.

## 1.2. Ingeniería de Software...

### 1.2.1. Definición

Es importante tener presente el concepto de Ingeniería de Software, para identificar en donde se encuentra la etapa de *análisis* de sistemas dentro de esta área.

El concepto de "Ingeniería de Software" apareció a fines de los 60's, cuando se observó que el desarrollo de grandes sistemas de software era un problema más parecido a los problemas de ingeniería que a los matemáticos o de ciencias naturales. Definiéndose que la Ingeniería de Software trata de la construcción de sistemas que son más grandes de lo que normalmente podría manejar un individuo, tiene aspectos técnicos y no técnicos. Teniendo como reto el crear software fácil de mantener, modificar y mejorar.

Además de tener un profundo conocimiento de las técnicas de computación, el ingeniero de software, debe poder comunicarse en forma oral y por escrito; debe apreciar los problemas que tienen los usuarios de sistemas para trabajar con un software cuyo funcionamiento puede no entender; también debe comprender los problemas de administración de proyectos relacionados con la producción de software. Entonces el ingeniero de software debe<sup>1</sup>:

- ✓ Ser un buen programador.
- ✓ Conocer bien las estructuras de datos y algoritmos.
- ✓ Saber programar en varios lenguajes.
- ✓ Conocer varios enfoques de análisis y diseño.
- ✓ Traducir los requerimientos vagos en especificaciones precisas.
- ✓ Conversar con el usuario en su área.
- ✓ Moverse en diversos niveles de abstracción:
  - Nivel de la aplicación
  - Abstracción del sistema
  - Código en detalle del sistema
- ✓ Saber construir un modelo de la aplicación.
- ✓ Ser miembro de un equipo de trabajo.
- ✓ Organizar y planear su trabajo y el de su equipo.

---

<sup>1</sup> IBARGÜENGOITIA, González Ma. Guadalupe. Ingeniería de Software, Notas de Clase Maestría en Ciencias de la Computación, Pg.4

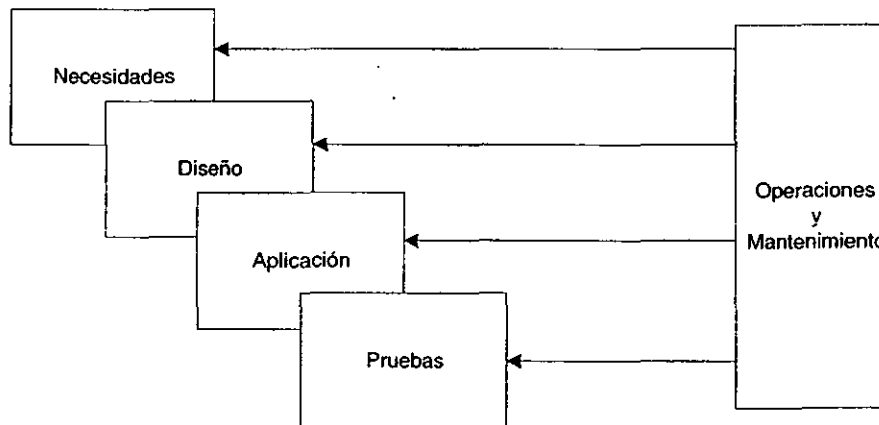
Además el término *software* no sólo abarca los programas de computación asociados con alguna aplicación o producto. Junto con los programas, el software incluye toda la documentación necesaria para instalar, usar, desarrollar y mantener esos programas. En cuanto a los sistemas grandes, la tarea de construir tales documentos es de magnitud comparable a la de desarrollar los programas<sup>2</sup>.

## 1.2.2. Ciclo de Vida del Software

Las etapas identificadas en la vida de un Software son<sup>3</sup>:

1. Análisis y definición de necesidades.
2. Diseño del sistema.
3. Aplicación y pruebas de unidades.
4. Pruebas del sistema.
5. Operación y mantenimiento.

Estas etapas se encuentran relacionadas como se muestra en la Fig. 1.1, Las primeras cuatro se proporcionan información unas a otras y la etapa de *operación y mantenimiento* retroalimenta a las anteriores. Aquí se observa el papel que juega el proceso de *análisis* en la Ingeniería de Software.



**Fig. 1.1<sup>4</sup>** El ciclo de vida del software

<sup>2</sup> SOMMERVILLE, Ian. "Ingeniería de Software". Addison-Wesley Iberoamericana. 2da. Edición. México D.F. 1988. Pag. 2

<sup>3</sup> SOMMERVILLE, Op Cit. Pag. 3

## **1.3. Análisis y Diseño, una Herramienta de Construcción de Software...**

El Análisis y Diseño de una Aplicación son consideradas como las herramientas necesarias en el Desarrollo de Sistemas de Información; y como se conceptualizó anteriormente parte significativa en la Ingeniería de Software. Estas herramientas a la fecha son poco aplicadas en la industria de construcción de Software; esto lo hace que se considere un interesante tema de estudio.

Estas herramientas se refieren al proceso de examinar una situación de cierta empresa con la intención de mejorarla mediante nuevos procedimientos y métodos<sup>5</sup>.

El Análisis. Es el proceso que sirve para recopilar (levantamiento de requerimientos) e interpretar los hechos, diagnosticar problemas y utilizar estos hechos a fin de mejorar el sistema, así como especificar qué es lo que el sistema debe hacer y como alcanzar su objetivo.

El Diseño. Es una actividad creativa y crítica, es un paso que transforma los requerimientos del software, obtenidos en el análisis, en un producto final; es como un plano para una construcción; especifica todas las características que se considerarán en el producto terminado.

## **1.4. ¿Qué es ISO 9001?**

Vale la pena empezar con una explicación sobre lo que es ISO 9000, pues bien, el concepto **ISO**: fue creado en 1947, por una organización no gubernamental llamada The International Organization for Standardization en Ginebra, Suiza, con el propósito de ayudar al desarrollo de estándares que faciliten la mejora del mercado global, en cuanto a su desarrollo y servicio, con esto se garantiza que el producto proporcionado, por las empresas certificadas, aseguren un grado mínimo de calidad. Actualmente abarca más de 130 países, cada uno con sus propios representantes. Ésta estandarización se aplica en varias actividades como lo son: económicas, tecnológicas y de ciencia. Las siglas "ISO" es un termino relacionado con la empresa que lo fundó; además que su traducción, en varios idiomas, es de fácil entendimiento.

Una definición acertada del termino calidad es la que nos proporciona Joseph M. Juran la cual es:

*"Todas las características del producto libre de deficiencias que satisfacen las expectativas de los clientes"*

Un estudio formal del tema ISO 9000<sup>6</sup> es muy amplio lo cual queda fuera del alcance del presente trabajo, por lo que se limitó a mencionar los puntos relacionado con la normativa ISO 9001, que es sobre la cual se apoyó el presente estudio.

Los modelos ISO, para el aseguramiento de la Calidad se definen en la tabla 1.1.

---

<sup>4</sup> SOMMERVILLE, Op Cit. Pag.4

<sup>5</sup> SENN, James A. "Análisis y Diseño de Sistemas de Información". Mc. Graw-Hill. 1a. edición. México 1990. Pag. 5

<sup>6</sup> Para más información consultar [www.iso.ch/central@isocs.iso.ch](http://www.iso.ch/central@isocs.iso.ch)

Norma	Descripción
ISO-9001	Modelo para el aseguramiento de la Calidad aplicable al diseño, desarrollo, producción, instalación y servicio al cliente.
ISO-9002	Modelo para el aseguramiento de la calidad aplicable a la fabricación e instalación.
ISO-9003	Modelo para el aseguramiento de la calidad aplicable a la inspección y pruebas finales. Aplican a empresas distribuidoras o comercializadoras de productos, que no son fabricados por ellos.

**Tabla 1.1**<sup>7</sup> Modelos contractuales de ISO 9000:1994

### 1.4.1 ISO 9000 en México

En México se adoptó la normatividad ISO 9000 en 1990 con las normas NOM-CC, normas aplicables por sectores (Seguridad, Salud, Protección al Consumidor, Protección al Medio Ambiente). En ese tiempo el organismo regulador de normas y estándares internacionales era la Dirección General de Normas (DGN). Fue este organismo en conjunto con SECOFI<sup>8</sup>, que el primero de julio de 1992 decidió cambiar las primeras siglas por la nomenclatura NMX, siendo estas últimas de cumplimiento voluntario.

En este mismo año, es promulgada la LEY Federal sobre Metrología y Normalización, en la cual se describe el esquema mexicano de normalización y certificación, dando por primera vez el respaldo legal para que el sector privado pueda promoverlos.

La nomenclatura mexicana correspondiente a la norma ISO 9001 es NMX-CC-003, lo cual esta definido<sup>9</sup> como:

*NMX-CC-003 -Sistema de calidad- Modelo para el aseguramiento de la calidad en diseño, desarrollo, producción, instalación y servicio. Es aplicable cuando un proveedor debe asegurar la conformidad con los requisitos especificados durante el diseño, desarrollo, producción, instalación y servicio.*

Se enfatiza que los requisitos de los sistemas de calidad especificados en esta norma, y en las normas NMX-CC-004<sup>10</sup> Y NMX-CC-005<sup>11</sup> son complementarios (no alternativos) a los requisitos técnicos especificados (del producto<sup>12</sup>). Estos especifican los requisitos que tienen que ser cubiertos, pero no es el propósito de estas normas forzar la uniformidad

<sup>7</sup> TORRES Alvarez O., "Gestión Pública: Certificación de la calidad en el proceso de atención a clientes en la Delegación Regional Zona Sur del Instituto Mexicano del Petroleo", Memoria de desempeño profesional Para obtener el título de Licenciado en Ciencias Política y Administración Pública, Pag 63

<sup>8</sup> Son las siglas de Secretaría de Comercio y Fomento Industrial

<sup>9</sup> NMX-CC-003:1995 IMNC Pag. 1

<sup>10</sup> Modelo para el aseguramiento de la calidad de la producción, instalación y servicio

<sup>11</sup> Modelo para el aseguramiento de la calidad en inspección y pruebas finales

<sup>12</sup> Para efectos de esta norma, el término "producto" se aplica solamente a la oferta del producto intencionado y no a los sub-productos no intencionados que afectan al medio ambiente.

en los sistemas de calidad; ya que estos pueden ser genéricos e independientes de cualquier industria o sector económico específico. El diseño e implantación del sistema de calidad tiene necesariamente que estar influido por las diversas necesidades de una organización, por sus objetivos particulares, por los productos y servicios suministrados y los procesos y prácticas específicas que son empleadas.

Se pretende que estas normas se adopten en su forma presente, pero en ocasiones pueden necesitar adaptarse añadiendo o eliminando ciertos requisitos del sistema de calidad para situaciones específicas.

## 1.4.2 Proceso de Certificación ISO 9001

Obtener un certificado ISO 9001, implica que la empresa a la que se aplica una estandarización, en primera, ha logrado expresar su modo de organizarse y de como lleva a cabo el desarrollo de sus productos o servicios, determinando una serie de pasos a los que llamaremos *procedimiento*, así como, la definición de la documentación que respalde o evidencie la ejecución de dichas tareas, este procedimiento deberá incluir, además, ciertos requisitos de la norma ISO 9001. Lo siguiente será aplicarlo al desarrollo de sus productos o servicios. La propia empresa deberá llevar a cabo Auditorías Internas, que le ayuden a corroborar la buena implantación de su procedimiento. Ya comprobado que la empresa realiza adecuadamente sus procedimientos, alguna organización representante de ISO 9000 hará una "Auditoria Externa"; está calificará la calidad de la empresa decidiendo si es o no merecedora de la certificación solicitada.

Posteriormente ya otorgado el certificado ISO 9000, la organización representante, dará seguimiento, realizando periódicamente Auditorías Externas con el fin de verificar la calidad de la empresa certificada. Añadiendo que es responsabilidad de la empresa certificada, el buscar continuamente la mejora de sus procesos, lo cual conlleva a un ciclo continuo de cambios en sus procedimientos, formando parte de la *Planeación estratégica*<sup>13</sup> de la empresa.

Los beneficios<sup>14</sup> de contar con la certificación ISO 9001 son:

- ✓ Da ventaja contra la competencia
- ✓ Proporciona confianza a nuestros clientes
- ✓ Se obtiene consistencia en el suministro de productos y servicios
- ✓ Se incrementa el número de clientes satisfechos
- ✓ Se mejora la imagen de la compañía
- ✓ Se desarrollan planes adecuados de trabajo
- ✓ Se disminuyen los recursos de inspección
- ✓ Se incrementa la eficacia y eficiencia de las operaciones
- ✓ Proporciona un cambio de cultura positivo
- ✓ Se reducen desperdicios, retrabajos y desgastes
- ✓ Se incrementa la productividad y el lucro de la empresa
- ✓ Nos da un incremento en el negocio

---

<sup>13</sup> Más adelante se explicará el tema

<sup>14</sup> TORRES Alvarez O. Op cit. Pag. 65

## **1.5. Las Metodologías de Análisis de Sistemas de Información**

### **1.5.1. ¿Qué es una Metodología?**

La **Metodología** es un conjunto de métodos, reglas y postulados que permiten que la forma de producción de software sea más predecible. Tiene como objetivo el producir software de calidad, de fácil mantenimiento; proporcionar un control efectivo de desarrollo; facilitar la estimación de recursos a utilizar y el tiempo requerido.

La importancia de utilizar una metodología radica en la facilidad que proporciona para llevar a cabo el mantenimiento de una aplicación. En las ciencias computacionales como en la medicina, lo mejor es prevenir y Miller<sup>15</sup> definió el Mantenimiento preventivo, de software, como la aplicación de las metodologías actuales a sistemas de ayer para facilitar los requisitos de mañana.

Las metodologías que se utilizarán en el presente trabajo, están basadas en las dos filosofías de programación de mayor auge en nuestros días, que son: Estructurada y Orientada a Objetos. Aunque existen varias metodologías, aplicables a dichas filosofías, es de importancia especificar que se estudió únicamente dos de las más importantes, (sino es que las más importantes en el mercado), Youdon Structured Analysis/Structured Design (SA/SD) y Unified Modeling Language (UML) de Booch, Jacobson y Rumbaugh.

### **1.5.2. El Modelo en Sistemas de Información**

La definición común de Modelo es: "La simplificación de la realidad"<sup>16</sup>. Por lo que es de entenderse que el modelado aplicado a software, es la representación del sistema comunicando el que y cómo se hace, esto es ya como producto final; y tiene como principal objetivo el de ser la base de toda aplicación, además al realizar un modelo se desarrolla una etapa importante del "Enfoque de sistemas"<sup>17</sup>. La construcción de modelos de software, nos ayudan a controlar el riesgo .

El modelo del sistema es una visión de muy alto nivel del sistema en la que se identifican los servicios principales al usuario y se documentan sus relaciones<sup>18</sup>.

Otra definición muy acertada de modelo es: *El ciclo de vida del sistema es un modelo utilizado para explicar y ayudarnos a entender el proceso de desarrollo y mantenimiento del sistema*<sup>19</sup>. Como se observó en el apartado 1.2.2

---

<sup>15</sup> IBARGÜENGOITIA, Op cit, Pg.68

<sup>16</sup> BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. "El Lenguaje Unificado de Modelado". Addison Wesley Iberoamericana. Madrid, 1999. Pag.5

<sup>17</sup> CHURCHMAN, West C. "El Enfoque de Sistemas". Diana. México 1993. Pag.81

<sup>18</sup> SOMEERVILLE, Op Cit Pag.17

<sup>19</sup> RENDON, Aguilar Juan C. "Trabajo de memorias del desempeño profesional: Importancia de Utilizar una metodología en el Desarrollo de Sistemas". MAT 45. México 1993. Pag. 12

### **1.5.3. ¿Porqué una Metodología de Análisis en el Desarrollo de una Solución Informática?**

La mayoría de las personas que se dedican al desarrollo de software, en cualquier etapa, hacen un pequeño bosquejo de lo que se debe hacer para obtener los resultados solicitados, como lo son los diagramas de flujo, el pseudocódigo, las pruebas de escritorio; que son pequeñas herramientas que ayudan a visualizar el resultado, en la mayoría de los casos no tan profundamente; sin embargo esto dista de ser una gran ayuda aplicándolo a un sistema informático, ya que esto suele ser muy informal, y no proporcionan un lenguaje común que se pueda compartir fácilmente con otros.

El modelar permite comunicarnos con el usuario final de una manera clara, sin distraerse con asuntos y características ajenas al sistema. Al analizar se enfoca a ciertas propiedades relevantes del sistema; mientras que si se desarrolla (o se programa) a la par de cuando se analiza, se tiende a omitir aspectos importantes del software; otra de las ventajas es que la aplicación de un análisis hace que el cambio a los requerimientos del usuario sean a menor costo.

Para desarrollar software de calidad duradera hay que idear una base sólida (modelo) arquitectónica que sea flexible al cambio<sup>20</sup>. La razón fundamental de porque se modela es:

*para comprender mejor al sistema que se desarrolla o mejora; consiguiendo cuatro objetivos.<sup>21</sup>*

- 1. Los modelos nos ayudan a visualizar cómo es o queremos que sea un sistema.*
- 2. Los modelos nos permiten especificar la estructura o el comportamiento de un sistema.*
- 3. Los modelos nos proporcionan plantillas que nos guían en la construcción de un sistema.*
- 4. Los modelos documentan las decisiones que hemos adoptado.*

El modelar un sistema no es sólo para los sistemas grandes; la mayoría de las empresas dedicadas al desarrollo de software no utilizan ninguna metodología formal en sus sistemas grandes mucho menos en los pequeños. Sin embargo se ha observado que la aplicación de un modelado formal es necesario para los sistemas de mayor complejidad y robustez; Ya que: Construimos modelos de sistemas complejos porque no podemos comprender el sistema en su totalidad<sup>22</sup>. Se ha observado que cuanto más complejo sea un proyecto, de no hacer nada de modelado, lo más probable es que se fracase o no se construya el producto apropiado. Además el utilizar una metodología formal en el desarrollo de software, ayuda al equipo de trabajo a visualizar mejor el plano de su sistema, a permitirles desarrollar más rápidamente, obteniendo el producto apropiado.

---

<sup>20</sup> BOOCH, op cit Pag. 3

<sup>21</sup> BOOCH, op cit Pag. 6

<sup>22</sup> Idem



Los objetivos a lograr dentro de la calidad del software son: la *confiabilidad* y el *mantenimiento* de la misma aplicación. ¿Como se puede lograr estos objetivos? Los analistas de sistemas pueden llevar a cabo una serie de pasos (Metodología) que aseguren que el sistema es confiable cuando sea instalado y que la confiabilidad se mantenga después de la puesta en marcha.

La *Confiabilidad*. un sistema confiable es aquel que no produce peligros o fallas cuando se utiliza en forma razonable, es decir, de la manera que un usuario común espera que sea normal<sup>23</sup>. La confiabilidad depende de dos niveles<sup>24</sup>:

1. Que el sistema cubra los requerimientos correctos. Aquí el analista deberá realizar una completa y efectiva determinación de los requerimientos del sistema.
2. El trabajo efectivo proporcionado al usuario. Aquí se refiere al producto final que será entregado al usuario, depende de como se ha desarrollado el software.

Otro punto importante a trabajar para el logro de la confiabilidad, es el de evitar errores; se debe estar consciente de que a pesar de los mejores esfuerzos que se hagan, es imposible alcanzar este propósito en su totalidad; sin embargo en los programas se debe buscar este objetivo.

Los errores no sólo son de sintaxis, existen muchos tipos de ellos, aquí se identifica a los de mayor importancia que son los que se presentan en las etapas de Análisis y Diseño. A continuación se mencionan los errores que interesan en la etapa de análisis<sup>25</sup>:

- 1) No obtener requerimientos en forma correcta;
- 2) No obtener los requerimientos adecuados; y
- 3) No interpretar los requerimientos en forma clara y entendible de manera que los programadores los puedan poner en marcha en forma apropiada.

El *Mantenimiento de Sistemas*. Cuando existen equivocaciones y son detectados después de que el sistema ya se haya puesto en marcha, se recurre al mantenimiento. Este, siempre será vigente mientras el sistema se encuentre vigente. Aquí el trabajo del analista es tomar las precauciones para asegurar que la necesidad de mantenimiento sea controlada a través del diseño y prueba. También se incluyen las adaptaciones a las versiones iniciales del software.

Para las adaptaciones del software se tiene que: la mayor cantidad de trabajo de mantenimiento es para mejoras del usuario, de la documentación o del registro de componentes del sistema para una mayor eficiencia. El 60% de todo el mantenimiento es para este propósito; incluso muchas de las tareas se pueden evitar si se llevará una metodología formal de Análisis y Diseño apropiadamente, ver Tabla 1.2.

---

<sup>23</sup> SENN, op cit. Pag. 489

<sup>24</sup> Idem

<sup>25</sup> SENN. op cit. Pag. 492

<b>CATEGORIA</b>	<b>ACTIVIDAD</b>	<b>FRECUENCIA RELATIVA</b>
CORRECTIVO	Arreglos de emergencia. Depuración rutinaria.	20%
ADAPTATIVO	Adaptación de cambios a datos y archivos y al software y hardware del sistema.	20%
PERFECTIVO	Mejora para el usuario. Perfeccionar la documentación, nueva codificación para lograr eficiencia en el cómputo.	60%

**Tabla 1.2**<sup>26</sup> Tipo de mantenimiento de sistemas

Las conclusiones obtenidas de estudios hechos por universidades y el gobierno en los años 80's, las cuales no distan mucho de las observaciones actuales, sobre la solicitud de mantenimiento a los sistemas de información fueron las siguientes<sup>27</sup>:

1. El 60 a 90% del costo total del software durante la vida de un sistema se gasta en mantenimiento ver Fig. 1.2.
2. A menudo el mantenimiento no se hace en forma muy eficiente. En los casos documentados, el costo de mantenimiento es más de 50 veces el costo al desarrollarlo por primera vez (Boehm).
3. La demanda de software está creciendo a una tasa más rápida que la oferta. Muchos programadores gastan más tiempo en el mantenimiento de sistemas que en un nuevo desarrollo. Los estudios han informado que en algunos sitios dos tercios de los programadores emplean su tiempo en el mantenimiento de software. Existe un retraso en el trabajo del nuevo desarrollo; más aún, existe un retraso escondido, solicitudes para desarrollo de trabajo que los usuarios no se molestarán en introducir debido a que saben que pasarán años antes de que el desarrollo pueda iniciarse.

<sup>26</sup> SENN. op cit Pag.494

<sup>27</sup> SENN. op cit. Pag.493

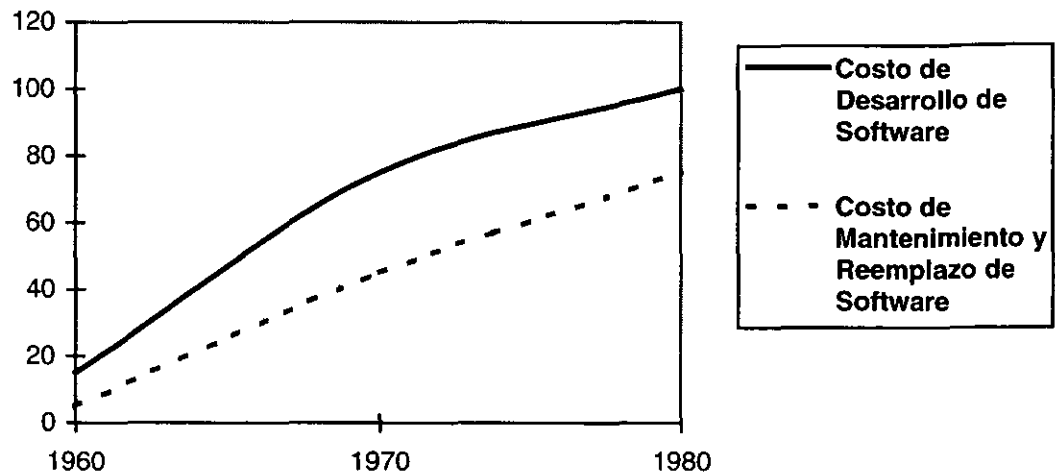


Fig. 1.2<sup>28</sup> Tendencia de los Costos de Software y de Mantenimiento

#### 1.5.4. El Origen de SA/SD

Edward Yourdon, es el creador de la nueva metodología de análisis estructurado moderno. Estudioso de las metodologías de sistemas, aplicó su experiencia sobre lo ya escrito a finales de los 70's por Tom DeMarco con su Structured Analysis and Systemas Specification, y de Chis Gane y Trish Sarson con su obra Structured System Analysis: Tools and Techniques.

El análisis estructurado es una prolongación de la herramienta de Flujo de datos, la cual es bien aceptada por cualquier persona con nociones básicas de programación; es llevada a cabo por análisis de aplicaciones desarrolladas bajo la filosofía de programación estructurada. Especificaremos más sobre esta metodología en el capítulo II

#### 1.5.5. El Origen de UML

La aparición de metodologías de análisis y diseño con la finalidad de desarrollar el modelado en la programación orientada a objetos data de mediados de los 70's y finales de los 80's. Iniciando con la presentación de numerosos tipos de ellas, las cuales de forma separada cubrían partes de sus necesidades, originando con esto una guerra de métodos. Siendo de los métodos más completos; los trabajos de Grady Booch, de Ivar Jacobson con su método Object-Oriented Software Engineering (OOSE), y el de James Rumbaugh con Object Modeling Technique (OMT).

En octubre de 1994 Booch, Jacobson y Rumbaugh deciden unificar esfuerzos en Rational ( [www.rational.com](http://www.rational.com) ), para crear un lenguaje que se utilice en el análisis y diseño orientados a objetos. Agregándole la retroalimentación de la comunidad internacional relacionada con la ingeniería de software (organizaciones externas). Apareciendo en octubre de 1995 el borrador de la versión 0.8 del llamado entonces Método Unificado. Y

<sup>28</sup> Idem

en enero de 1997 se produjo el UML 1.0, "un lenguaje de modelado bien definido, expresivo, potente y aplicable a un amplio espectro de dominios de problemas"<sup>29</sup>; a partir de entonces han aparecido versiones mejoradas. Especificaremos más de esta metodología en el capítulo II.

## 1.6. ¿Porque Implantar ISO-9000 en la Empresa Consultora?

Cuando la Empresa Consultora en estudio llegaba a los 10 años de estar en el mercado ofreciendo sus productos y servicios; procurado mantenerse a la vanguardia en cuanto a tecnología de computo se trata y con el fin de satisfacer las necesidades de sus clientes. Percibiendo que en el mundo actual, la calidad es cada día más importante, un claro ejemplo es: las empresas de consultoría de sistemas informáticos cada día son más, que se ha generado una guerra constante entre ellas, sobresaliendo las que presenten mejores productos, servicios y precios. Por estas razones la Empresa Consultora decide llevar a cabo dentro de su planeación estratégica a mediano plazo (1 año) para el crecimiento de la compañía y a través de la diversificación<sup>30</sup>, la estandarización de sus procesos y comprometerse con sus clientes al ofrecer un sistema de aseguramiento de calidad desarrollado bajo los estándares de ISO 9000.

Ya que la norma ISO 9001 ha demostrado, a nivel internacional, ser un mecanismo reconocido y confiable con lineamientos muy claros para estandarizar procesos, se toma la decisión de llevar a cabo un plan de acción para lograr la certificación bajo los estándares de esta norma.

Aquí presentamos una definición de calidad del software<sup>31</sup>:

*Es la concordancia de los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo y con las características implícitas que se espera tenga un software desarrollado profesionalmente.*

*La garantía de la calidad del software es un plan de acciones sistemáticas. Es responsabilidad de muchas personas en la organización: ingenieros de software, analistas, clientes, etc.*

De esto se obtiene una clara motivación para aplicar una estandarización de alineamientos de calidad al desarrollo de aplicaciones a la medida:

*La calidad no se impone a posteriori, es el resultado del uso de métodos y técnicas durante el proceso de desarrollo<sup>32</sup>*

---

<sup>29</sup> BOOCH op cit. Pag.XXIII

<sup>30</sup> Notas de clase del Seminario Taller "Análisis de la Planeación", módulo III - Planeación e Instrumentos de la Administración; con el Profesor Juan Torres Lovera, del día 24 de junio del 2000; La diversificación es un tipo de estrategia utilizado en la planeación para el crecimiento de una empresa.

<sup>31</sup> IBARGÜENGOTTIA op cit. Pag.51

<sup>32</sup> Idem

Además el implantar ISO 9000, proporciona los siguientes beneficios<sup>33</sup>

- ✓ Es una ventaja competitiva dentro del mercado globalizado y de intensa competencia
- ✓ Mejora la calidad de las operaciones
- ✓ Incrementa la eficiencia y la capacidad productiva, obteniendo mayores utilidades para la empresa
- ✓ Se cuenta con procesos controlados que evitan reprocesos, desperdicios y costos que no agregan valor
- ✓ Se cumple a los clientes que exigen una auditoría
- ✓ Se establece un mejor control de los proveedores, lo que permite que se reduzcan los reclamos, devoluciones falta de materiales.
- ✓ Se eliminan las improvisaciones
- ✓ Se consolida una cultura de trabajo basada en el orden y la consistencia

Estas fueron las motivaciones de la consultoría y como se ve bien relacionados con lo expuesto en el subtema 1.2.2:

## 1.7 Planeación Estratégica en el Estudio del Procedimiento DAM

**La PLANEACION.** Es el proceso de determinar objetivos y definir la mejor manera de alcanzarlos. Define los medios, de como se debe hacer, y los fines, qué es lo que se tiene que hacer. La planeación tiene dos conceptos la formal y la informal. Teniendo que en la informal nada está escrito y en pocas o ningunas ocasiones se comparten los objetivos con otros en la organización, es general y carece de continuidad; mientras que en la formal, existen objetivos específicos, que están escritos y disponibles para todos los miembros de la organización, también se encuentra bien definida la ruta que se debe tomar, de donde están a donde quieren llegar.

El *propósito* de la planeación es dar: dirección, reducir el impacto del cambio, minimizar el desperdicio y la redundancia y fijar los estándares para facilitar el control. La planeación estimula la coordinación, cooperación y el trabajo en equipo de todos los involucrados, ya que todos saben a donde se dirige la organización y lo que se espera que contribuyan para alcanzar los objetivos. La planeación es también una manera de reducir la incertidumbre a través de la previsión del cambio. La fuerzas de planeación miran hacia adelante, prevén los cambios, consideran el impacto de estos cambios y desarrollan respuesta apropiadas. Sin planeación no puede haber control.

Los planes Estratégicos son los que cubren a toda la organización, que establecen sus objetivos generales y que buscan colocarla en términos de su ambiente. Tienden a incluir un periodo extenso, abarcan un área más amplia, tratan menos los aspectos específicos, incluyen formulación de objetivos.

La planeación estratégica se desarrolla bajo nueve pasos<sup>34</sup> que son:

1. Definir la *misión* de la organización. Aquí se obtendrá la misión que define su propósito. Esto se logra haciéndose una serie de preguntas involucradas con lo que la organización realmente hace, obteniendo la finalidad del espacio de su producto o servicio.
2. Establecer *objetivos*. Señalando el fundamento de cualquier programa de planeación. Los objetivos trasladan la misión a términos concretos.

<sup>33</sup> TORRES Alvarez O. Op cit. Pag. 65

<sup>34</sup> Robbins Stephen P. "Administración: Teoría y Práctica". Prentice Hall Hispanoamericana, S.A. México D.F. Pag. 140-145

3. *Analizar los recursos* de la organización. Lo que la administración puede hacer está limitado por los recursos y capacidades que la organización posee. Los recursos pueden ser: humanos, financieros y físicos. Aquí también se busca la ventaja comparativa que tiene la organización con su competencia, así como también, sus debilidades.
4. *Examinar el ambiente*. Aquí se identifican los diversos factores políticos, sociales, económicos y de mercado que pudiesen incidir sobre la organización. Dividir el análisis en tres estados: 1) previsiones sectoriales. Aquí se hace la identificación y análisis de los sectores o áreas de interés, estudia la historia de cada uno, para prever el futuro 2) análisis cruzado de impacto. Se hace la asignación de probabilidades a las diversas tendencias, eventos y discontinuidades, encontradas en el punto anterior y 3) diseño de escenario. Crea escenarios con conciencia interna. Desarrollando cuatro derivados de las combinaciones del punto 2. Estos deben cubrir los extremos: desde un escenario sin sorpresas hasta uno que tenga la peor situación.
5. Hacer *predicciones*. Se hace un esfuerzo por encontrar la posible ocurrencia de eventos futuros. Incluye en el análisis los factores internos y externos. Las técnicas de predicción más conocidas son: Los modelos econométricos de predicción, la extrapolación de tendencia, los modelos de regresión, los paneles de expertos y las simulaciones por computadora.
6. *Analizar oportunidades y riesgos*. El análisis de recursos de la organización y las predicciones de factores internos y externos constituyen una base de datos con los cuales la administración analiza las oportunidades y riesgos. El mismo ambiente puede ser una oportunidad para una organización y una amenaza para otra en la misma industria debido a recursos diferentes.
7. Identificar y evaluar *estrategias alternativas*. Una vez identificada una oportunidad o riesgo, la administración debe buscar una serie de alternativas que pueden explotar la situación.
8. *Seleccionar estrategia*. Una vez que las estrategias alternativas han sido listadas y evaluadas, se seleccionará una. Existe la opción de continuar haciendo lo que la administración ha hecho en el pasado. Sin embargo, si se selecciona una nueva estrategia, debe ser consistente con la misión y objetivos de la organización y debe corresponder bien a sus capacidades.
9. *Instrumentar la estrategia*. La mejor estrategia puede fallar si la administración no es capaz de traducirla en programas, políticas, presupuestos y otros planes a corto y largo plazo necesarios para llevarla a cabo. En esta etapa se requiere de una comunicación exitosa de la estrategia a todos los niveles de la administración, para que cada uno de los involucrados haga su mejor esfuerzo en realizar lo que le corresponde hacer. Si algún área no lleva a cabo su parte de la estrategia, el esfuerzo total podría venirse abajo.

Los Tipos<sup>35</sup> de Estrategias que se contemplan son cuatro:

1. *Estabilidad*. Se caracteriza por la ausencia de cambios significativos, un ejemplo de ésta es, el continuar sirviendo a los mismos clientes ofreciendo el mismo producto o servicio, manteniendo la participación en el mercado o sosteniendo el nivel pasado de rendimiento sobre la inversión realizada.
2. *Crecimiento*. Aumentar el nivel de operaciones de la organización como lo son: más ingresos, más empleados y una mayor participación en el mercado. Existen tres maneras en las que se puede alcanzar el crecimiento de una organización: a través de la expansión directa, una fusión con firmas similares o diversificación.
3. *Contracción*. Esta se lleva a cabo reduciendo el tamaño o diversidad de sus operaciones como lo son: el recorte de departamentos, productos, servicios y personal.
4. *Combinación*. Aquí se refiere a la ejecución simultánea de dos o más de las estrategias anteriores en distintas partes de la organización o el uso de dos o más estrategias llevadas a cabo con el tiempo.

---

<sup>35</sup> ROBBINS Stephen. Op cit. Pag.145-148

Si se toma en cuenta estos conceptos es fácil asociar el cómo se llevó a cabo el desarrollo del presente trabajo con la utilización de la planeación estratégica.

## **1.8 Situación Actual del Procedimiento Desarrollo de Aplicaciones a la Medida: Etapa de Análisis**

El procedimiento "Desarrollo de Aplicaciones a la Medida", que de manera práctica se denomina DAM, es el procedimiento que respalda al desarrollo de Software dentro de la Empresa Consultora en estudio, ante la organización que otorgó el certificado ISO 9001.

Este procedimiento ha enriquecido a la organización de la empresa, obteniendo de este buenos frutos, sin embargo, aun se tienen deficiencias muy notorias, como lo son: La falta de entendimiento de algunos pasos por parte de los responsables y ejecutores; A los documentos a desarrollar les falta claridad de llenado; lo que sigue ocasionando pérdidas económicas; La etapa de análisis está incompleta, por lo que no ayuda al buen mantenimiento de la aplicación, perdiendo tiempo en el entender el lenguaje que se ha llevado a cabo.

El DAM fue certificado el 31 de agosto de 1999.

Su objetivo:

*Establecer las etapas a seguir en el desarrollo de proyectos, indicando Cómo, Cuándo y Quién realizará cada una de las actividades; especificando las herramientas (Guías, Ejemplos y Formatos) que se pueden utilizar durante el desarrollo del proyecto; así como los registros que se deben crear<sup>36</sup>.*

Sus etapas:

*Elaboración de propuesta, con 7 actividades;  
Inicio del Proyecto, con 2 actividades;  
Análisis Preliminar, con 6 actividades;  
Análisis, 8 actividades;  
Diseño de Bases de Datos, 8 actividades;  
Diseño, 4 actividades;  
Pruebas unitarias, 3 actividades;  
Pruebas integrales, 2 actividades;  
Entrega, 3 actividades.*

La etapa de análisis a mejorar tiene las siguientes actividades:

1. Elaboración del plan de trabajo de Análisis. Aquí se definirán las actividades necesarias para efectuar el análisis, así como, la asignación de tiempos a cada una de ellas. Las actividades son, como mínimo, las especificadas en el mismo procedimiento en esta etapa de análisis.
2. Obtener y documentar entidades involucradas, entradas y salidas y relaciones entre entidades. En esta actividad se obtendrá y documentará a partir de la información proporcionada: las entidades que tengan o vayan a tener relación directa o indirecta con el sistema; las entradas y salidas de información de las entidades y del sistema; y las relaciones que existen entre las entidades.
3. Elaborar Diagrama de Contexto General (Metodología SA/SD) o Diagrama de Negocio (Metodología UML). Realización de un paso utilizando ya la metodología de análisis y diseño seleccionada.

---

<sup>36</sup> "DAM", REV No.3. 25 Agosto 1999

4. Revisión y Aprobación del Diagrama de contexto (Metodología SA/SD) o Diagrama de Negocio (Metodología UML). Verificar y aceptar el (los) documento(s) generado(s), haciendo el registro del Vo.Bo.
5. Lista de Eventos (Metodología SA/SD). Realización de un paso ocupando la metodología de análisis y diseño seleccionada.
6. Elaborar Diagrama de Flujo ,DFD (Metodología SA/SD) o Casos de Uso (Metodología UML). Realización de un paso ocupando la metodología de análisis y diseño seleccionada
7. Revisión y Aprobación del DFD (Metodología SA/SD) o Casos de Uso (Metodología UML). Verificar y aceptar el (los) documento(s) generado(s), haciendo el registro del Vo.Bo.
8. Especificaciones del Proceso (Metodología SA/SD) o Documentación de Casos de Uso (Metodología UML). Realización de un paso ocupando la metodología de análisis y diseño seleccionada.

## **1.9 Mejora Continua**

Fundamentalmente el control de calidad se apoya en el concepto de Mejora Continua; refiriéndose aquí a la ejecución constante del proceso de aplicación de la calidad. Enfatizando que el proceso de Mejoramiento de la Calidad nunca termina. En el presente estudio se utilizó el ciclo Deming<sup>37</sup>, el cual consta de los siguientes cuatro pasos<sup>38</sup>:

- Paso 1: Estudiar el proceso, decidir que cambios podría mejorarlo, organizar el equipo apropiado.  
No proceder sin tener un plan.
- Paso 2: Efectuar las pruebas o cambios, preferentemente en pequeña escala.
- Paso 3: Observar los efectos ¿es exactamente lo que necesitamos?
- Paso 4: Repetir la prueba, si es necesario, tal vez en un ambiente diferente y dejar establecidos los Cambios.

Aquí se asocia este concepto de la mejora continua como parte de la actividad de control dentro de la planeación estratégica que se lleva a cabo.

## **1.10 Diagnóstico del Problema**

Un concepto importante para la realización correcta de la planeación es, como primer paso, el Diagnosticar la situación de nuestro problema.

Definiendo al diagnóstico como, la exposición de los resultados obtenidos de las evaluaciones realizadas al problema que se desea resolver.

A continuación se exponen los elementos considerados como parte de la Planeación, para la realización del diagnóstico del presente trabajo.

---

<sup>37</sup> W. Edwards Deming. Estadístico Estadounidense, socio fundador de American Society for Quality Control (ASQC); misionero de la búsqueda de las fuentes del mejoramiento de la calidad.

<sup>38</sup> TORRES Alvarez O. Op cit. Pag. 50



## 1.10.1 La Misión de Calidad de la Empresa Consultora

La misión de una empresa define su propósito; indicando qué es lo que la empresa desea realizar o su finalidad; tomando en cuenta que al crear una misión en una empresa, esta debe indicar que es lo que realmente se está haciendo.

Tenemos que la misión de calidad aplicada a la empresa en estudio es: " *Crear, ofrecer y proveer los productos y servicios que requieran nuestros clientes en tecnología de información de una manera oportuna, eficaz y eficiente*<sup>39</sup>, *en beneficio de nuestros clientes y personal, siempre orientados a incrementar la rentabilidad y la participación global de nuestra empresa, apegados a nuestros principios éticos*<sup>40</sup>"

Esta misión no ha obtenido buenos resultados, fallando en que no ha logrado proporcionar productos y servicios eficazmente; esto ha dado como resultado la pérdida de algunos clientes importantes, por lo tanto ha disminuido la continuidad de negocio en dichas empresas.

La eficacia es reflejada en la buena o mala aplicación de un plan estratégico, esto nos lleva a entender que actualmente la planeación del desarrollo de los servicios y productos está fallando. La aplicación incompleta de una metodología de análisis y diseño es uno de los puntos clave en los que la empresa ha presentado complicaciones.

Por lo que tenemos una misión que está intentando cumplirse, sin embargo aún no logra sus objetivos.

En todo proceso de mejoramiento de la calidad se recomienda previamente involucrar a los integrantes de la dirección en los siguientes conceptos<sup>41</sup>:

- ✓ **Comprensión** Todos deben saber que la Calidad se puede definir, medir y manejar. La comprensión comienza en el nivel directivo.
- ✓ **Compromiso** De igual manera comienza en el nivel directivo, el compromiso con los principios fundamentales de Calidad, una vez que la dirección ha aceptado el compromiso los empleados comenzarán a aceptar el suyo propio, adoptando el compromiso a Cero Defectos como estándar personal de realización.
- ✓ **Competencia** La compañía determina el método para garantizar que todos entienden y tenga oportunidad de participar en el Mejoramiento de la Calidad.
- ✓ **Comunicación** Para que haya un Mejoramiento de la Calidad debe existir una comunicación fuerte y eficaz.
- ✓ **Corrección** La corrección empieza con requisitos claros seguidos de medidas que ayudan a su cumplimiento, de no haber un sistema de eliminación de problemas no habrá Mejoramiento de la Calidad.
- ✓ **Continuidad** La continuidad asegura que el Proceso de Mejoramiento de la Calidad no termine nunca, la continuidad se basa en el hecho de que nunca es más económico, ni más rápido hacer algo bien por segunda vez.

<sup>39</sup> Notas de Clase del Seminario Taller "Análisis de la Planeación", módulo III – Planeación e Instrumentos de la Administración; con el Profesor Juan Torres Lovera, del día 20 de junio del 2000; La eficacia se evalúa en la competitividad estratégica que se lleva a cabo, mientras que la eficiencia corresponde a la operativa

<sup>40</sup> La misión de la empresa consultora hasta Agosto de 1999.

<sup>41</sup> TORRES Alvarez O. Op cit. Pag. 43

## 1.10.2 Definición de Componentes

Aquí definiremos los aspectos involucrados en el problema; como parte del análisis realizado. El tipo de problema que se presenta esta en función de sus componentes<sup>42</sup>. Se tiene que el problema esta compuesto por lo expuesto en la siguiente tabla Tabla 1.3.

Componente	Porcentaje (%)
Administrativo	35
Económico	20
Tiempo	20
Técnico	25

**Tabla 1.3<sup>43</sup>** Porcentajes subjetivos del tipo de problema expuesto

El problema es: administrativo porque el procedimiento y su ejecución es responsabilidad de los administradores, por lo tanto también de la organización de cómo se lleva a cabo el análisis; económico porque existen perdidas importantes con respecto a la rentabilidad de los proyectos (productos); de tiempo ya que es carácter urgente para la empresa; y técnico porque hay un desconocimiento notable de alguna herramienta de análisis en el personal involucrado.

## 1.10.3 Variables Controlables y No Controlables

Para poder explicar un sistema integral desde un enfoque de sistemas, es importante definir el medio ambiente en el cual se desenvuelve y como está apoyado por las actividades de sus partes<sup>44</sup>.

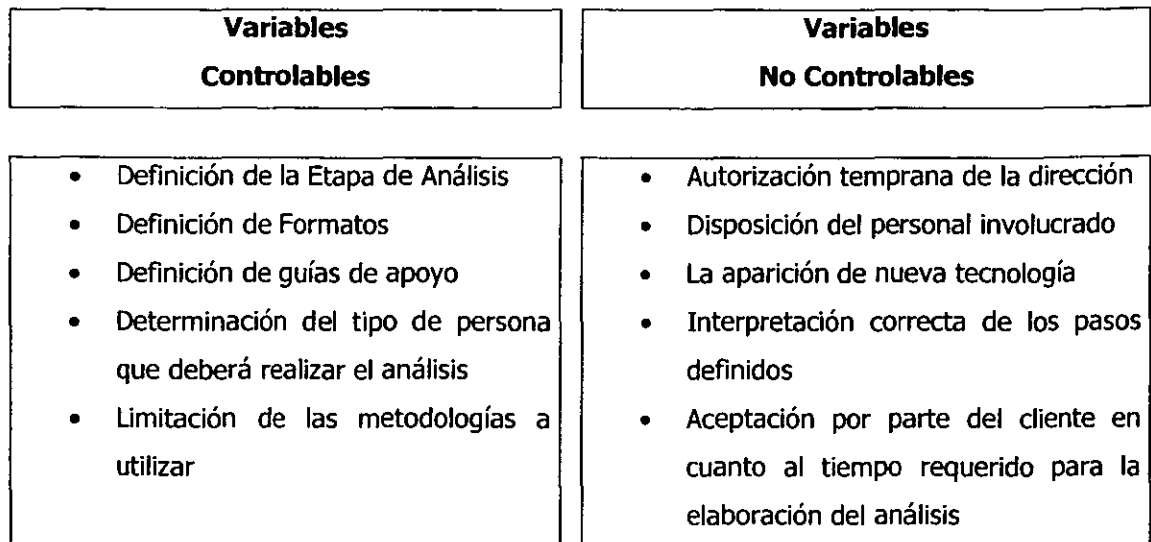
Por un lado tenemos que el medio ambiente de un sistema es representado por las restricciones fijas que tiene; es lo que esta afuera de él; agentes externos que definen como opera el sistema y no podemos hacer nada respecto a sus características o a su comportamiento teniendo entonces las variables no controlables. Y se define a las variables controlables como los recursos con los que cuenta el sistema; son las cosas que el sistema puede cambiar y utilizar para su propio provecho; también se deciden sobre cuales de ellas deben trabajar, como se gasta el dinero en las actividades y los limites de tiempo que pudiera tener.

Se puede determinar cada una de las variables controlables y no controlables contestando las siguientes preguntas ¿podría yo hacer algo acerca de ello? ¿influye en mis objetivos? Si la primer respuesta es que no y la segunda si entonces estamos hablando de una variable no controlable<sup>45</sup>. Resultando entonces lo expuesto en la Fig. 1.3.

<sup>42</sup> Notas de Clase del Seminario Taller "Análisis de la Planeación", módulo III – Planeación e Instrumentos de la Administración; con el Profesor Juan Torres Lovera, del día 20 de junio del 2000

<sup>43</sup> Como parte de la planeación se determinó por criterio propio los porcentajes en los que se considerará la problemática expuesta en el presente trabajo

<sup>44</sup> CHURCHMAN. op cit. Pag. 52-53



**Fig. 1.3<sup>46</sup>.** Variables controlables y no controlables del problema expuesto

## **1.11 Diagrama de Contexto y de Flujo de Datos de la Solución de la Problemática Actual**

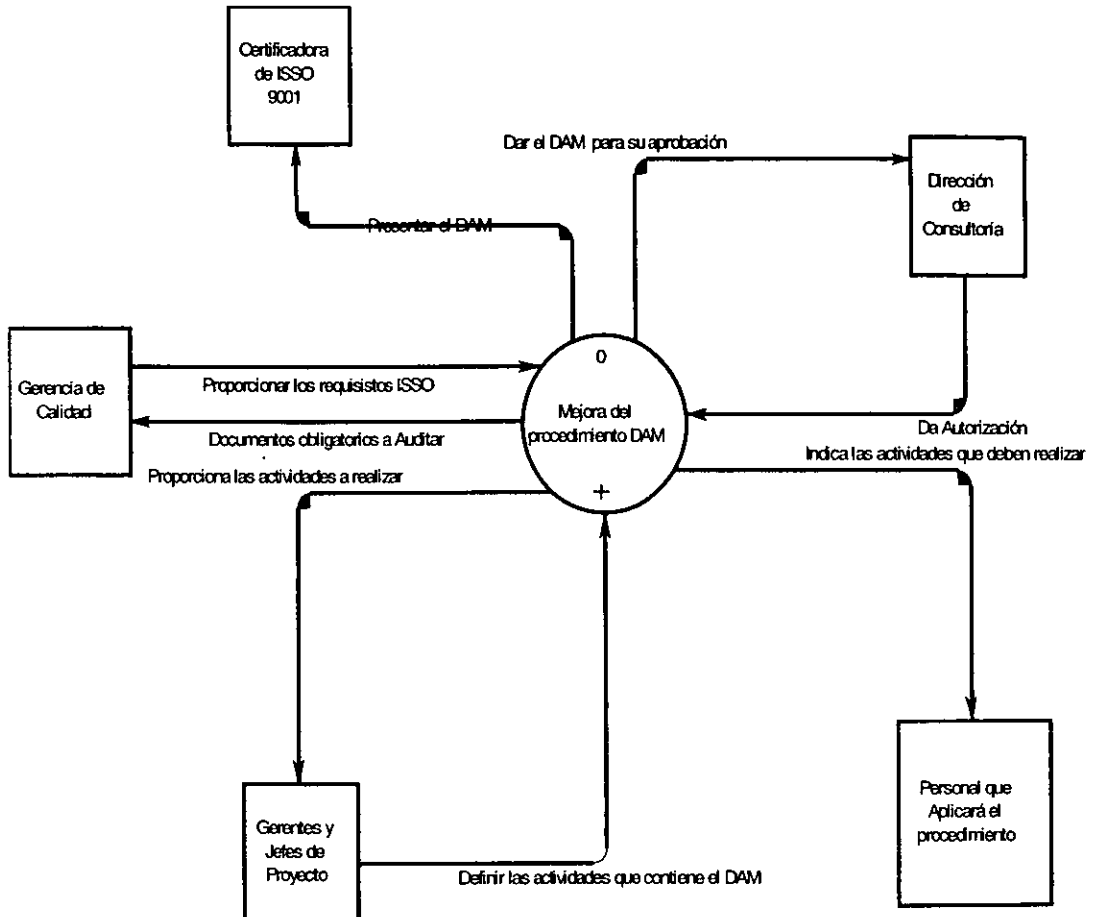
En el Diagrama de Contexto se representa la relación del proyecto, éste representado con un círculo, con sus entidades involucradas, representadas por cuadros grandes; las líneas que los unen indican la función que las relacionan, y la dirección de las flechas indican el flujo de información es decir el origen y destino.

El diagrama de contexto y de flujo de datos forma parte del análisis de la planeación ejecutada en el presente estudio.

Este diagrama es importante ya que expresa como se dará solución al problema ver la Fig. 1.4

<sup>45</sup> CHURCHMAN. Op cit. Pag. 55

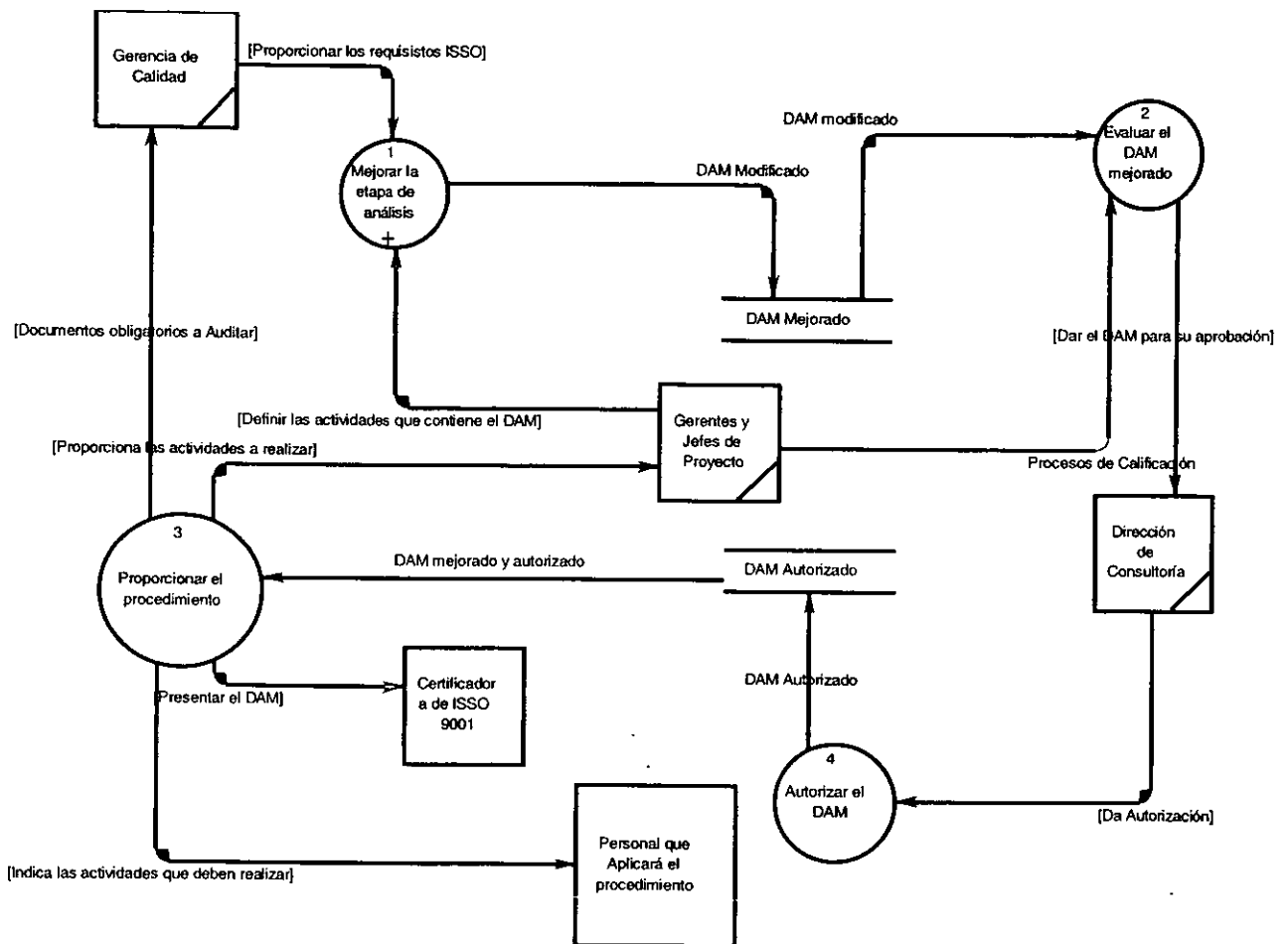
<sup>46</sup> Fuente: Creación Propia



**Fig. 1.4<sup>47</sup> Diagrama de Contexto**

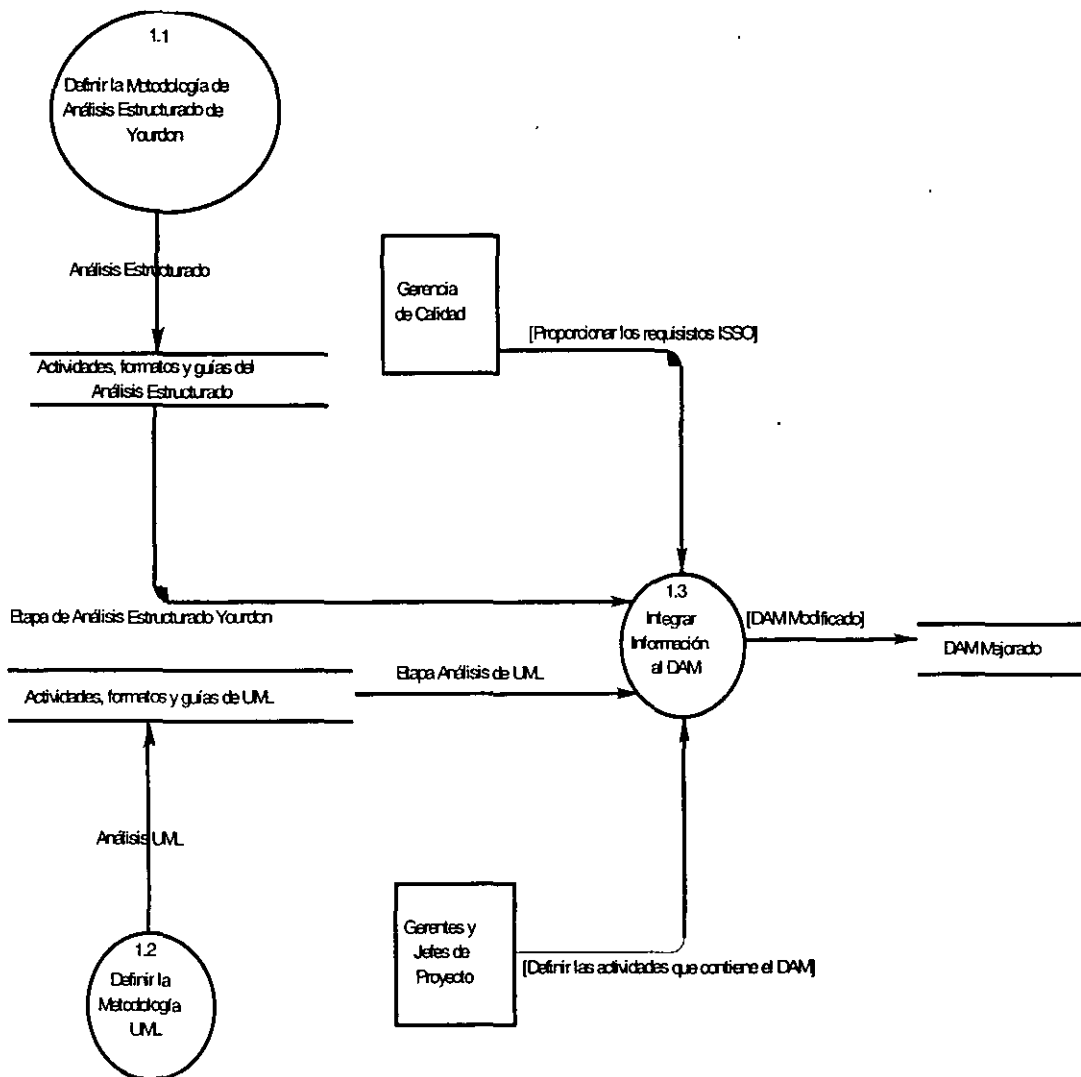
<sup>47</sup> Fuente: Creación Propia

En el diagrama de flujo de datos se indica, de manera sencilla y de fácil entendimiento, los procesos que integra el estudio; los datos que emplea cada proceso; los datos que son almacenados; y los datos que integran y abandonan el sistema. Entendiendo que los datos son la guía de las actividades del estudio. Para nuestro estudio tenemos los diagramas de flujos de datos presentados en las Figuras 1.5 y 1.6 .



**Fig. 1.5<sup>48</sup> DFD-Nivel 1 de Mejorar la etapa de análisis**

<sup>48</sup> Fuente: Creación Propia



**Fig. 1.6<sup>49</sup> DFD-Nivel 2 de Mejorar la etapa de análisis**

Se limita al presente estudio a la ejecución del proceso de mejorar la etapa de análisis (Fig. 1.5 proceso 1).

Para más información sobre las herramientas de diagrama de contexto y diagrama de flujo de datos consultar el capítulo II sección 2.4.1 del presente trabajo.

<sup>49</sup> Fuente: Creación Propia

## **Conclusiones del Capítulo I:**

A pesar de que la existencia de metodologías de análisis y diseño de sistemas de información lleva ya varias décadas, la industria de la computación no se ha aplicado adecuadamente a la utilización de alguna de ellas, resultando, como se pudo apreciar en este capítulo en los apartados 1.3, 1.5.3, y 1.8, a una serie de consecuencias como lo son: un alto gasto, por supuesto innecesario, en mantenimiento, corrección y desgastes infrahumanos. Esta problemática se encuentra presente en la empresa consultora en estudio lo que impulsó a la realización del presente trabajo.

Se observó la relación que existe entre el concepto de metodología de análisis y diseño de sistemas con el de Ingeniería de Software en los apartados 1.2.1 y 1.5.1; verificando en el apartado 1.2.2 que la etapa de análisis de sistemas pertenece al ciclo de vida de la ingeniería de software; así como la ingeniería de software a su vez utiliza de la planeación como se vio en el apartado 1.2.1 y 1.7. Teniendo así la relación que existe entre lo que es una metodología de análisis y diseño de sistemas con el uso de la planeación.

El uso adecuado de alguna metodología de análisis es aplicable a varios tipos de problemas, no exclusivamente computacionales, ayuda a entender cualquier situación a la que nos enfrentemos, como el caso de lo mostrado en este capítulo apartado 1.11, en el que se plantea la solución a la presente investigación utilizando la metodología de análisis estructurado Yourdon. De esto se obtiene que al hacer uso de alguna metodología en la aplicación de la *Planeación* de cualquier problema que se desee resolver, genera un buen resultado.

Es importante recomendar al que planea obtener un certificado ISO 9000 en sus desarrollos tome la decisión de llevar a cabo el uso de alguna metodología de análisis y diseño, como se observó en los apartados 1.6, 1.8, 1.9, 1.10 y 1.10.1; y para los que ya tienen ese certificado será bueno mantenerse a la vanguardia en las que utilicen y en las que aparezcan, esto es, para la mejora de sus procedimientos.

Por último se tiene que la correcta aplicación de una metodología de análisis y diseño de sistemas de información es una utilidad esencial para garantizar la calidad de cualquier tipo de software; utilizándolo cualquier persona, grupo de personas y/o compañías dedicadas al desarrollo de aplicaciones, inclusive el departamento interno de sistemas de cualquier tipo de industria. Con esto se cumple el objetivo de este capítulo: el señalar la importancia de utilizar una metodología de análisis y diseño de sistemas de información.

## FUENTES DE CONSULTA

### Administración Teoría y Práctica

Robbins Stephen P.  
Prentice- Hall Hispanoamericana, S.A.  
México

### Análisis y Diseño de Sistemas de Información

Senn, James A.  
1a. Edición  
Mc Graw-Hill  
México 1990

### El Enfoque de Sistemas

Churchman, West C.  
Diana  
México 1993

### El Lenguaje Unificado de Modelado

Booch Grady, Rumbaugh James, Jacobson Ivar  
Addison Wesley  
Madrid España 1999

### Ingeniería de Software

Segunda Edición  
Sommerville Ian  
Addison Wesley Iberoamericana  
México D.F. 1988

### Notas de clase: Ingeniería de Software

Ibargüengoitia, González Ma. Guadalupe  
Maestría en Ciencias de la Computación  
Facultad de Ciencias, UNAM

### Trabajo de Memorias del Desempeño Profesional: Importancia de Utilizar una Metodología en el Desarrollo de Sistemas

Rendón Aguilar Juan Carlos  
MAT 45  
1993

### Trabajo de Memorias del Desempeño Profesional: Gestión Pública: Certificación de la calidad en el proceso de atención a clientes en la Delegación Regional Zona SUR del Instituto Mexicano del Petróleo

Torres Alvarez Ivan  
2000

Procedimiento: Desarrollo de Aplicaciones a la Medida (DAM)

[www.iso.ch/central@isocs.iso.ch](http://www.iso.ch/central@isocs.iso.ch)

### Sistema de Calidad- Modelo para el aseguramiento de la calidad en diseño, desarrollo, producción, instalación y servicio

NMX-CC-003: 1995 IMNC  
ISO 9001:1994



## **Capítulo II:**

# ***Consideraciones en el Análisis de las Metodologías Estructuradas "Yourdon" y Orientado a Objetos "UML"...***

*Objetivo: Se llevará a cabo el estudio y se  
presentarán los pasos requeridos, en el análisis de  
una solución informática, de cada una de las dos  
metodologías mencionadas.*

**"CUANDO UN HOMBRE SABE A DONDE VA  
EL MUNDO ENTERO SE APARTA PARA DARLE PASO"**

**Thomas Jefferson**

## **2.1. Definición de Programación Estructurada**

En principio se define que un sistema estructurado es aquel que se desarrolla en forma descendente y modular. Cada uno de los módulos son simples, lo que significa que tienen un efecto mínimo sobre otros módulos dentro del sistema por lo que las conexiones entre ellos se limitan y la interacción de los datos es mínima<sup>50</sup>.

Entendiendo la definición anterior entonces ahora se define a la programación estructurada como una programación de tipo modular; trata también otros aspectos ligados al lenguaje de programación y a la organización del trabajo, es fácil tanto para la producción de los programas como para su documentación y mantenimiento. La creación de los programas aplica el criterio de subdividir el problema dado en partes de tal manera que se agilice el proceso de entender por completo tanto el problema como la solución<sup>51</sup>. Esta última es una técnica para construir los esquemas de encadenamiento de las sentencias del programa.

Los módulos o subdivisiones deseadas deben cumplir los siguientes requisitos:

- a) Deberán estar jerarquizados; aplicando una programación de "arriba a abajo" (top-down) de acuerdo a una descomposición arborescente, modular, que permita el fraccionamiento de los programas en módulos o segmentos. Se interpreta que el programa consta de un módulo principal y de varios módulos de nivel más bajo (gobernados por el principal) que se encargan de ejecutar las órdenes dadas por él.
- b) Deberán ser pequeños y sencillos de fácil seguimiento (no más de un par de páginas de listado).
- c) Deberán esconder los detalles poco importantes a módulos superiores en la jerarquía; que facilite el mantenimiento de los programas, al permitir modificar un "sub-árbol", submódulo, del esquema general sin necesidad de tocar el que está arriba de él o a su lado.
- d) Deberán, a su vez, usar tantos módulos de más baja jerarquía como sea necesario para cumplir con el punto b).
- e) Deberán usar las estructuras de datos y control adecuadas para cumplir con el punto b).
- f) Dar a los programas una estructura suficientemente fuerte para que los organigramas no sean tan rígidos, proporcionando la característica de dinámicos.
- g) Deberán ser legibles; esto es, que no sólo su autor sea capaz de entenderlos, sino cualquiera que tenga acceso a ellos y a un conocimiento elemental de programación.

El resultado deberá ser una relación de estructuras óptimas, módulos funcionales, que deben trabajar armónicamente entre ellos.

---

<sup>50</sup> SENN. Op cit. Pag.495

<sup>51</sup> LENNE G. Guillermo. "Introducción a la Computación y a la Programación Estructurada". MC. Graw-Hill. 2a. Edición. México, D.F. 1990. Pag.213

## **2.2. Definición de Programación Orientada a Objetos (OOP)**

La Programación Orientada a Objetos, identificada comúnmente como OOP, no se debe tratar como un significado sino más bien como un fin. A diferencia con la Programación Estructurada que va orientada a procesos, la OOP se enfoca a la definición de objetos; esta orientado a datos donde un dato es encapsulado en objetos y mensajes, estos son utilizados para la manipulación de datos.

La definición que da BOOCH para un objeto es la siguiente:

*Un objeto tiene estado, comportamiento e identidad, la estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancia y objetos son intercambiables. Un objeto se distingue de otro cuando existen diferencias u fronteras bien definidas.*

El estado de un objeto abarca todas las propiedades (normalmente estáticas) del mismo más los valores actuales (normalmente dinámicos) de cada una de esas propiedades. Y el comportamiento, es cómo actúa y reacciona un objeto, en términos de sus cambios de estado (eventos) y paso de mensajes (recepción y/o envío). Estos conceptos son de gran utilidad en el desarrollo de la metodología orientada a objetos UML, que se detallará en la sección 2.4.2.

Todos los objetos que tienen características similares pertenecen a una misma clase; la clase es uno de los conceptos primordiales de la OOP; una definición sobre lo que es una clase es que; se puede entender como una definición abstracta de las características de objetos que tienen la misma apariencia y que permiten la ejecución de operaciones similares en datos encapsulados. Un objeto es una instancia de una clase particular y es distinto de otra instancia de la misma clase.

Hay cuatro elementos que la OOP debe cumplir<sup>52</sup>: Abstracción, Encapsulamiento, Modularidad y Jerarquía.

Una *abstracción* es aquella que se enfoca en los detalles significativos para el usuario, eliminando los que son irrelevantes. Berzins, Gray y Naumann<sup>53</sup> recomiendan que:

*Un concepto merece el calificativo de abstracción sólo si se puede describir, comprender y analizar independientemente del mecanismo que vaya a utilizarse eventualmente para realizarlo*

Booch, define a la abstracción como:

*Una abstracción denota las características esenciales de un objeto que lo distinguen de todos los demás tipos de objeto y proporciona así fronteras conceptuales nítidamente definidas respecto a la perspectiva del observador*

El *encapsulamiento* oculta los detalles de la implementación de un objeto<sup>54</sup>, definiéndolos como datos privados, permite que los cambios hechos en los programas sean fiables con el menor esfuerzo, este elemento asegura que el objeto ejecute sus operaciones adecuadamente. Un objeto encapsulado puede ser accesado o manipulado por funciones o características genéricas, las cuales también deben estar definidas dentro del mismo objeto.

<sup>52</sup> BOOCH Grady. "Análisis y Diseño Orientado a Objetos". Segunda Edición. Addison-Wesley/Diaz de Santos, E.U.A. 1996. Pag.45

<sup>53</sup> BOOCH Grady. Op cit. Pag 46

<sup>54</sup> BOOCH Grady. Op cit. Pag 55

La *modularidad* empaqueta las abstracciones en unidades discretas<sup>55</sup>, se utiliza como organizador de elementos ya sea clases, objetos, operaciones, etc. En cualquier lenguaje de programación lo mejor es agrupar sus elementos que se relacionen lógicamente en módulos.

La *jerarquía* conocida también como herencia simple y herencia múltiple, es una clasificación u ordenación de abstracciones. Permite la creación de una nueva clase usando una o varias clases existentes como modelo o base. Es decir que la nueva clase estará definida en términos de la(s) clase (s) existente(s)<sup>56</sup>.

Existen otros tres elementos que son útiles pero no esenciales, los cuales son: Tipos (tipificación), Concurrencia y Persistencia.

Un *tipo* es una caracterización precisa de propiedades estructurales o de comportamiento que comparten una serie de entidades. El uso de los términos de tipo y clase suelen ser el mismo. No se puede hacer operaciones con elementos de diferentes tipos.

La *concurrencia* permite a diferentes objetos actuar al mismo tiempo<sup>57</sup>, es una especie de controlador de acciones de manera sincronizada.

La *persistencia* conserva el estado de un objeto en el tiempo y en el espacio<sup>58</sup>. Se almacena los resultados transitorios que suelen perderse entre la ejecución de un programa y la base de datos. Booch, proporciona la siguiente definición sobre persistencia:

*La persistencia es la propiedad de un objeto por la que su existencia trasciende en el tiempo (es decir, el objeto continúa existiendo después de que su creador deja de existir) y/o el espacio (es decir, la posición del objeto varía con respecto al espacio de direcciones en el que fue creado)*

Para aprovechar de buena forma la OOP, es necesario apoyarse con otras herramientas que son<sup>59</sup>:

- 1) Análisis de requerimientos Orientados a Objetos
- 2) Diseño Orientado a Objetos
- 3) Análisis de dominios Orientado a Objetos
- 4) Sistema de Base de Datos Orientado a Objetos
- 5) Ingeniería de software Orientado a Objetos
- 6) Programación Orientado a Objetos

Las cinco primeras herramientas mencionadas, son parte de la metodología orientada a objetos UML; se limita el presente estudio únicamente en la herramienta 1, el análisis de requerimientos orientados a objetos, definiendo este tema en el apartado 2.4.2.

La OOP facilita la interoperabilidad, genera código reusable; genera software de fácil modificación, crecimiento y mantenimiento; el resultado de la comparación con varias técnicas de programación ha sido favorable a la OOP, además de ser el tipo de programación donde se dirige la tecnología.

---

<sup>55</sup> BOOCH Grady. Op cit. Pag 61

<sup>56</sup> RAO Bindu R. "C++ and the OOP Paradigm". Cap Gemini America Series. Mc Graw Hill. USA 1993. Pag.4

<sup>57</sup> BOOCH G. Op Cit. Pag 82

<sup>58</sup> BOOCH G. Op Cit. Pag 85

<sup>59</sup> RAO Bindu R. Op cit. Pag.5

## **2.3. Características y Definiciones Básicas de la Metodología...**

De acuerdo a lo descrito en el capítulo uno inciso 1.11, en donde se planea como se lleva a cabo el estudio del DAM; ahora se procede a la etapa de ejecución dentro de lo planeado, continuando con la definición de las metodologías de Análisis Estructurado de Yourdon y UML.

### **2.3.1. Estructurada "Yourdon"**

En el capítulo I se vio una definición de lo que es metodología y el origen del SA/SD ahora se expone una definición de lo que es la metodología Estructurada.

El análisis estructurado tiene relación con los aspectos de reconocimiento de la naturaleza dinámica, no estática, de los sistemas en las organizaciones. Se trata de describir de manera correcta el sistema que se desea mejorar, se dice mejorar puesto que siempre la generación de un cambio o la creación de un nuevo sistema de información tiene como objetivo el de mejorar la situación actual de una empresa.

El Análisis Estructurado tiene como finalidad :

- a) Aprender los detalles y procedimientos del sistema en uso.
- b) Tener un pensamiento prospectivo sobre sus necesidades y su medio ambiente (situación en el mercado, avances tecnológicos, etc).
- c) Crear la documentación necesaria para la comprensión de los procesos, componentes y la interrelación entre ellos, tanto los actuales como los propuestos.
- d) Evaluar la eficiencia y efectividad del sistema actual en cuanto al impacto sobre las demandas anticipadas
- e) Justificar la solución propuesta tanto los cambios como la creación de un nuevo sistema.
- f) Fomentar la participación de todo el personal involucrado en los procesos para aprovechar su experiencia y conocimiento del sistema actual.

El análisis estructurado permite al analista conocer un sistema o proceso (módulo) en una forma lógica y manejable al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle importante<sup>60</sup>.

Algunas de sus ventajas son: presentan herramientas de notación sencilla fáciles de entender, permite la diagramación de actividades paralelas, es decir, representar actividades con mayor exactitud al mostrarlas cuando ocurren al mismo tiempo; nos lleva a la creación de diagramas de entidad relación (ER), tal tecnología a servido correctamente por más de una década, en el que se presenta el modelo de una organización a través de entidades y las relaciones que existen entre ellas; es una metodología práctica, que requiere de actividades sencillas, así como de menos tiempo que algunas de las otras metodologías de análisis y diseño; la ingeniería de software esta enfocada a procesos y no a objetos.

Sin embargo tiene limitaciones; no es posible utilizarla para programación orientada a objetos; como esta metodología lleva al diseño de diagramas de entidad relación (ER), éste último presenta un vocabulario limitado para poder presentar todas las reglas relacionadas con datos del problema que estemos intentando resolver pues no puede describir el número infinito de relaciones que pueden surgir en un modelo de datos complejo, existen problemas

---

<sup>60</sup> SENN. op cit. Pag. 176

de cardinalidad; Es compleja la representación de subtipos "fijos" ya que requiere un producto cartesiano de todos los subtipos posibles; en el caso de que se desee representar la relación "varios a varios" se tiene que crear una tabla de intersección con dos relaciones dependientes del tipo "1 a varios"; tampoco se puede incluir información de procesos en el modelo de datos; no se puede distinguir algunas relaciones con el tiempo, es decir, que el tiempo forme parte del identificador único; no existe el concepto de clasificación múltiple, refiriéndose a la definición de clases compuestas de dos o más clases.

## **2.3.2. Orientada a Objetos "UML"**

El UML, Lenguaje Unificado de Modelado<sup>61</sup>, es la herramienta seleccionada para la modelación de sistemas complejos, de gran extensión de software. Como se detalló anteriormente, existen diferencias esenciales entre el pensamiento estructurado y el orientado a objetos, sin embargo esto no quiere decir que si se tiene desarrollo estructurado y se desea cambiar a un modelo orientado a objetos, se deba de desechar lo realizado para implantar lo nuevo, si se requerirá de un arduo trabajo pero se tendrán buenos resultados. El modelado de objetos además de tomar los principios relevantes del análisis estructurado, presenta una extensión o elementos novedosos que el anterior no tiene. Aunque UML es un lenguaje expresivo, no es fácil aprenderlo ni mucho menos utilizarlo.

Durante el análisis orientado a objetos se procura ante todo identificar y describir los objetos –o conceptos- dentro del dominio del problema que se desea resolver.

El UML se señala como un lenguaje que presenta las siguientes características para el modelado de software: Visualizar, Especificar, Construir y Documentar

*Visualizar* porque ayudará a pensar, definir y comunicar los algoritmos para el desarrollo de programas de manera sencilla, ya que contiene formas gráficas respaldadas con textos bien definidos.

*Especificar* ya que construye modelos precisos, no ambiguos y completos.

*Construir* porque sus modelos pueden conectarse directamente a diferentes lenguajes de programación<sup>62</sup>, es decir, que el código que genera el UML también es un lenguaje de programación que puede ser entendido por otros lenguajes.

Es para *Documentar*, debido a que cubre la documentación de la arquitectura de un sistema y todos sus detalles. También proporciona un lenguaje para expresar requisitos y pruebas; así como un lenguaje para modelar las actividades de planificación de proyectos y control de versiones<sup>63</sup>.

La buena utilización del UML en el desarrollo de sistemas de información generará<sup>64</sup>:

*Un proceso bien definido guiará a sus usuarios al decidir qué producir, qué actividades y que personal se emplea para crearlos y gestionarlos, cómo usar esos artefactos para medir y controlar el proyecto de forma global.*

<sup>61</sup> Se le llama lenguaje porque es una forma de comunicación, útil para el diálogo entre los participantes del desarrollo de sistemas.

<sup>62</sup> BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. Op cit. Pag 13

<sup>63</sup> BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. Op cit. Pag 14

<sup>64</sup> BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. Op cit. Pag 12

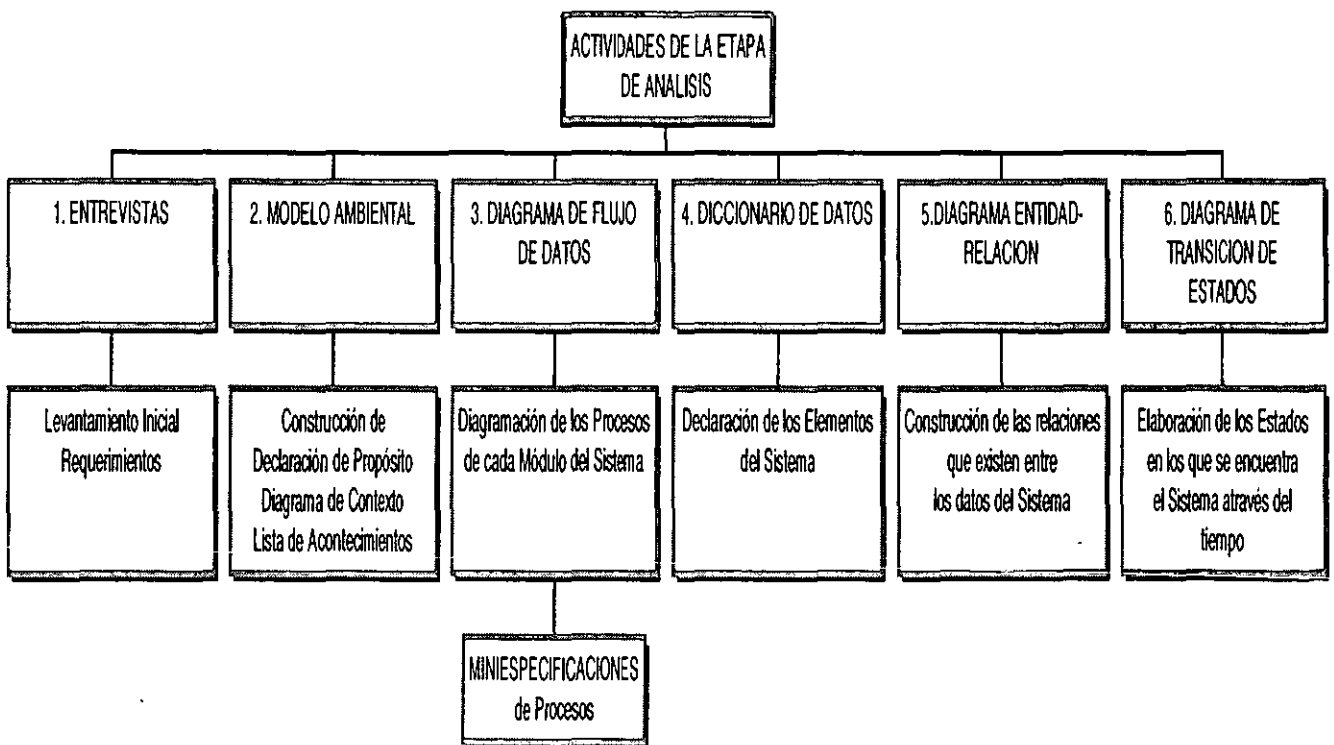
Las ventajas de utilizar UML son: se puede utilizar esta metodología para llegar a un diseño no exclusivamente orientado a objetos; es mucho más adaptable a los cambios que otros; permite especificaciones abiertas en algunas de sus herramientas; es capaz de modelar más relaciones en un diseño de base de datos que las relacionales;

Las desventajas son: es una metodología complicada de entender; existe el problema de que no hay muchas bases de datos orientas a objetos en el mercado, por lo que se recurre al adecuación de las base de datos relacionales para diseñar las orientadas objetos; permanece el problema de expresar las entidades relacionadas con el tiempo.

## 2.4. Etapas y Componentes del Análisis...

### 2.4.1. Metodología Estructurada "Yourdon"

Esta metodología esta compuesta como lo muestra la figura 2.1



**Fig. 2.1**<sup>65</sup> Organización del análisis estructurado Yourdon

<sup>65</sup> Fuente: Creación Propia

El Proceso de Análisis consta de las siguientes herramientas, las que se entenderán como pasos o actividades:

1. **Las Encuestas, Entrevistas y Recolección de Datos.** Las primeras también conocidas como el estudio de factibilidad o como el estudio inicial de negocios, sus objetivos son: identificar a los usuarios responsables y crear un "campo de actividad" inicial del sistema; establecer metas y objetivos para un sistema nuevo; desarrollar la estimación de costo-beneficio de la automatización del sistema<sup>66</sup>; crear el esquema que se usará para guiar el resto del proyecto. Esta actividad ocupa sólo del 5 al 10 por ciento del tiempo y recursos de todo el proyecto.

Las Entrevistas y Recolección de Datos se realizan a los usuarios involucrados, para reunir la información necesaria sobre el comportamiento de un sistema actual o de los nuevos requerimientos; verificar que el analista realmente haya interpretado correctamente las necesidades.

2. El **Modelo Ambiental.** Define la interfase entre el sistema y el resto del universo, es decir, el ambiente. Consta de tres componentes:
  - a) Declaración de propósito.- Es una declaración breve y concisa del propósito del sistema
  - b) El diagrama de contexto.- Es el Diagrama de Flujo de datos a nivel 0. Donde una sola burbuja representa todo el sistema y su relación con el mundo externo
  - c) La lista de acontecimientos.- Es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder.
3. El **Diagrama de Flujo de Datos (DFD).** Ilustra en forma gráfica las funciones que el sistema debe realizar; su interacción; las entradas; y las salidas. Estos diagramas no sólo se pueden utilizar para modelar sistemas de procesos de información, sino también como manera de modelar organizaciones enteras, es decir, como una herramienta para la planeación estratégica y de negocios<sup>67</sup>.

Sus componentes son:



Los *procesos* (o funciones) representados por medio de círculos o burbujas etiquetados. Son partes del sistema que transforman las entradas en salidas. Deben estar enumerados de acuerdo al número de proceso y el nivel (jerarquía) que representa. El proceso debe tener un nombre que deberá describir lo que hace, de manera que tenga significado para el usuario, en una frase del tipo "verbo infinitivo-objetivo"; por ejemplo Validar Entrada



Los *flujos* se muestran por medio de flechas. Son las conexiones entre los procesos y representan la información que dichos procesos requieren como entrada o la información que generan como salida dependiendo hacia donde se dirige la flecha; el flujo de dos cabezas representa un diálogo de tipo pregunta y respuesta. El nombre del flujo representa el significado del conjunto de

<sup>66</sup> Para más detalle del proceso de cálculo de costo-beneficio de un sistema consultar el libro de YOURDON E. "Análisis Estructurado Moderno", en su apéndice "C"

<sup>67</sup> YOURDON Edward. "Análisis Estructurado Moderno". Prentice Hall. México 1993. Pag.158



información que se mueve a lo largo del flujo. Si los flujos salen o entran de un almacenador puede no etiquetarse o tener la misma etiqueta que el almacenador cuando el paquete de datos corresponde a todos los datos que guarda el almacenador.



Los *almacenadores* también representados como una elipse. Modela una colección de paquetes de datos en reposo. Pueden representar formas computarizadas como son los archivos, bases de datos, discos etc., o formas no computarizadas como lo son fichas de papel en cajas de cartón, directorios, etc. Su nombre debe ser el plural del que se utilizan para los paquetes de entrada y salen del almacén por medio de flujos.



Los *Terminadores* son entidades externas con las que se comunica el sistema, las cuales no pueden ser controladas desde el sistema; en algunos casos pueden ser otros sistemas computacionales. Se debe considerar que las relaciones entre terminadores no deben modelarse en el DFD, a menos que sean relevantes en el sistema, aquí entonces los terminadores deberán modelarse como procesos.

Para la buena construcción de un DFD se deben tener las siguientes consideraciones:

- a) Cualquier flujo de datos que abandone un proceso debe estar basado en los datos que le entran.
- b) Sólo deben entrar al proceso los datos necesarios para llevarlo a cabo.
- c) Un proceso no debe saber nada de ningún otro en el sistema, es decir debe ser independiente; la única dependencia que debe existir es aquella que esté basada en sus propios datos de entrada y salida.
- d) Todos los flujos de datos que explican el proceso en el diagrama previo deben incluirse en el diagrama del siguiente nivel inferior.
- e) Los flujos y almacenes de datos nuevos se añaden si son utilizados internamente por el proceso para eslabonar otros procesos introducidos por primera vez en la expansión de éste nivel.
- f) Ninguna entrada debe contradecir las descripciones de los diagramas de flujo de datos de niveles más altos.
- g) Deberá ser de fácil lectura, dada nivel desarrollado contendrá 7 más menos 2 procesos.
- h) Determinar hasta que nivel de expansión se debe llevar, depende de la complejidad del sistema, sólo hay que llegar hasta que se comprendan sus detalles y la forma en que trabaja; teniendo cuidado de verificar todos los aspectos con usuarios que conocen el sistema. Por lo general el número de niveles oscila de 3 a 7 jerarquizados.
- i) Se deben expandir los procesos que incluyen varias tareas para las que es necesario el flujo de datos entre diferentes personas o localidades.
- j) No requieren de expansión aquellas tareas que son realizadas por una persona o en un escritorio, donde no existe flujo de datos.

- k) Evitar burbujas (procesos) que tienen entradas pero no salidas.
- l) Evitar burbujas (procesos) que tienen salidas sin tener entradas, aunque este tipo de proceso se puede aplicar en un generador de números aleatorios.
- m) Tener cuidado con los flujos y procesos no etiquetados
- n) Con excepción de almacenadores externos, tener cuidado con los almacenadores de sólo lectura o escritura.
- o) Asegurar que cada proceso, flujo de datos y almacenador se encuentre definido en el diccionario de datos.

Para modelar sistemas de tiempo real se utilizan los flujos de control, procesos de control y almacenadores de control; que son representados de la misma forma que los normales la diferencia es que sus líneas son punteadas. Dichos controles representan señales, interrupciones, que despiertan otros procesos normales.

Para evaluar que un DFD es correcto se recomienda hacer las siguientes preguntas, algunas tienen relación con las consideraciones mencionadas anteriormente:

- a) ¿Existen en el diagrama de flujo de datos componentes que no tienen nombre (flujos de datos, procesos, almacenamientos, entradas o salidas)?
  - b) ¿Existen almacenes de datos que son entradas y a los que nunca se hace referencia?
  - c) ¿Existen procesos que no reciben entradas?
  - d) ¿Existen procesos que no generan salidas?
  - e) ¿Existen procesos que tienen varias finalidades? (si es así, entonces se tienen que simplificar extendiéndolos en varios procesos para poder estudiarlos mejor)
  - f) ¿Existen almacenadores de datos a los que nunca se hace referencia?
  - g) ¿Es el flujo de datos que llega a un proceso, adecuado para realizarlo?
  - h) ¿Existen demasiados datos en el almacenador (más que los detalles necesarios)?
  - i) ¿El flujo de datos que llega a un proceso es demasiado extenso para la salida que éste produce?
  - j) ¿Se introducen alias en la descripción del sistema? ¿Aparecen en el diccionario de datos? (si no es así entonces pueden presentarse inconsistencias al describir y comprender el sistema)
  - k) ¿Los procesos son independientes entre sí? ¿Dependen sólo de los datos que reciben como entrada?
4. El **Diccionario de Datos**; ya que los diagramas de flujo de datos (DFD) por si solos no describen el objeto de la investigación se tiene el diccionario de datos que proporciona más información relacionada con el sistema.

Como lo menciona el autor en estudio Edward Yourdon<sup>68</sup>

*El diccionario de datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios*

---

<sup>68</sup> YOURDON. Op cit. Pag. 212

Es un catálogo de los elementos que forman parte del flujo de datos, almacenadores y procesos de todo el sistema, se refieren a los datos y la forma en que están estructurados para satisfacer los requerimientos de los usuarios y las necesidades de la organización. Permitiendo: comunicar un significado común para todos los involucrados, facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar dónde efectuar cambios en el sistema, y localizar errores y omisiones en el sistema

El detalle de lo que debe contener un diccionario de datos son:

Los *elementos básicos*, también conocidos como campos, su nombre debe de distinguirse, debe ser un nombre comprensible y significativo; debe contener una descripción que indique de manera breve y comprensible para el lector lo que representa en el sistema; indicar si tiene interacción, es decir, ocurrencias repetidas de un componente se lee como cero o más ocurrencias; definir el tipo de información que almacenará así como la cantidad de espacios, longitud, requeridos; especificar, en caso necesario, los valores permitidos; y si el dato es opcional o no.

Las *estructuras de datos*, es un grupo de datos elementales, con las definiciones anteriormente señaladas; que están relacionados con otros y que en conjunto describen un componente del sistema. Los flujos y los almacenadores son estructuras de datos, que están formados por elementos relevantes que describen la actividad o entidad bajo estudio.

La notación utilizada por el diccionario de datos y sencilla para su lectura se lista en la tabla 2.1.

Símbolo	Significado
=	Está compuesto de
+	Y
{ }	Iteración, se puede definir su límite inferior y/o superior a los lados derecho e izquierdo de las llaves, según corresponda.
( )	Optativo (puede estar presente o ausente)
[ ]	Seleccionar una de varias alternativas
**	Comentarios
@	Identificador (campo clave) para un almacenador
	Separa opciones alternativas en la construcción

**Tabla 2.1**<sup>69</sup> Notación simbólica del diccionario de datos

5. Las **especificaciones de procesos** (también conocidas como miniespecificaciones) indican lo que sucede en cada burbuja primitiva de nivel más bajo en un DFD, define lo que debe hacerse para transformar entradas en salidas. Es una descripción detallada de la política de negocios del usuario que cada burbuja lleva a cabo<sup>70</sup>

Existen varias herramientas que se pueden utilizar para producir una especificación de un proceso como lo son: tablas de decisiones, lenguaje estructurado (español, inglés, etc.), pre/post condiciones, diagramas de flujo, diagramas de Nassi/Shneiderman. Se debe tomar en cuenta que para la realización de las miniespecificaciones se debe satisfacer que: La especificación del proceso debe expresarse de una manera que puedan verificar

<sup>69</sup> YOURDON. Op cit. Pag 214.

<sup>70</sup> YOURDON. Op cit. Pag. 227

tanto el usuario como el analista; pueda ser comunicada efectivamente al público amplio que esté involucrado.

Sin embargo el presente estudio se enfoca en la técnica de preferencia y que ha tenido resultados exitosos que es la utilización del lenguaje estructurado. Este es un subconjunto de todo el idioma con importantes restricciones sobre el tipo de frases que pueden utilizarse y la manera en que pueden juntarse dichas frases<sup>71</sup>, esto es definir el propio lenguaje en la organización<sup>72</sup>, Su propósito es hacer un balance razonable entre la precisión del lenguaje formal de programación<sup>73</sup> y la informalidad y legibilidad del lenguaje cotidiano. Y como reglas se tiene lo siguiente:

- a) Los objetos a utilizar deben estar definidos en el diccionario de datos o ser términos locales (estos sólo son conocidos dentro de dicha especificación, también conocidas como variables locales).
- b) El nombre de los verbos, estructuras y lenguajes de programación deberán distinguirse siendo escritas con mayúsculas.
- c) Los datos y las variables locales deberán ser escritas en minúsculas.

Otras herramientas auxiliares son:

- 1) Las Pre/Post condiciones describen la función que debe realizar el proceso, sin decir mucho acerca del algoritmo o procedimiento que se utilizará<sup>74</sup>. Son útiles cuando un algoritmo ha sido especificado ya muchas veces que ya es obvia su descripción, también cuando se sabe que existen muchas formas de realizarse el proceso especificado que se puede realizar como el programador lo desee.

Las precondiciones describen todas las cosas que deben darse antes de que el proceso pueda comenzar, lo que pueden describir es: las entradas que llegan mediante un flujo conectado al proceso; la relación que existe entre las entradas, esto es cuando llegan dos entradas, o bien cuando una entrada debe estar dentro de un rango o intervalo; las relaciones que deben existir entre las entradas y los almacenadores, es decir condiciones de existencia; y las relaciones que deben existir entre diferentes almacenes o dentro de un almacenador específico.

Por otra parte las postcondiciones describen lo que procede cuando el proceso ha concluido, lo que pueden describir es: las salidas que genera el proceso; las relaciones de las salidas con los datos de entrada; las relaciones que existirán entre las salidas y los valores en uno o varios de los almacenadores, cuando la información debe recuperarse de un almacén y utilizarse como parte de la salida del proceso; y por último indicar los cambios que deberán ocurrir en los almacenadores.

- 2) Las Tablas de Decisión. Estas se utilizan cuando el lenguaje estructurado y las pre/post condiciones no son suficientes para expresar un algoritmo de un proceso, cuando existen decisiones complejas de expresar.

---

<sup>71</sup> YOURDON. Op cit. Pag. 231

<sup>72</sup> Se recomienda el uso de los siguientes verbos: OBTENER, ACEPTAR, LEER, MOSTRAR, ESCRIBIR, ENCONTRAR, BUSCAR, SUMAR, RESTAR, MULTIPLICAR, DIVIDIR, CALCULAR, BORRAR, ELIMINAR, VALIDAR, MOVER, REEMPLAZAR, FIJAR, ORDENAR, SELECCIONAR, etc.

<sup>73</sup> Aquí se refiere a las formas de tipo: HACER-MIENTRAS, FIN HACER, SI, FIN SI, HACER CASO, FIN HACER, SI-ENTONCES-OTRO, REPITE-HASTA, etc.

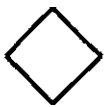
<sup>74</sup> YOURDON. Op cit. Pag. 240

Estas tablas se crean en primera listando todas las condiciones o entradas y las acciones relevantes, a la izquierda; después se lista en columnas separadas cada combinación posible de valores de las variables, conocidas como regla. Una regla describe una acción o acciones que deben llevarse a cabo para una combinación específica de valores de las variables. Se debe considerar que si existe N variables con valores binarios (verdadero-falso), entonces existirán  $2^N$  reglas distintas que deberán listarse.

6. El **Diagrama de Entidad-Relación (ER)**. Se hace énfasis en las relaciones entre los datos. Generalmente corresponden con los almacenadores de los DFD's. Describe la distribución de datos almacenados del sistema. Para la definición de sus componentes se tiene una relación estrecha con el diccionario de datos. La definición de un ER representa la forma física de como se almacenarán y organizarán los datos que componen el sistema. Sus componentes son:

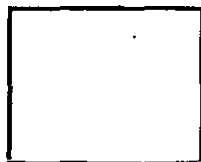


Conjunto o colección de objetos en donde cada uno de sus miembros se identifican como únicos.



Relaciones nos indican la conexión entre los objetos. Representan la memoria del sistema. Sus notaciones alternativas son por flechas dirigidas donde se indica la cardinalidad o asociación entre ellas.

7. El **Diagrama de Transición de Estados (DTE)**. Se enfoca al comportamiento dependiente del tiempo del sistema<sup>75</sup>. Se definen los estados de manera que se debe decir qué sucede cuando. En la mayoría de los casos el DTE representa una especificación de proceso para una burbuja (proceso) de control en un DFD, donde las condiciones son los flujos entrantes y las acciones los flujos de salida. Sus componentes son:



Representa los Estados en los que se puede encontrar el sistema en un tiempo dado. El estado se encuentra esperando que algo ocurra; representa algún comportamiento del sistema que es observable y que perdura durante algún periodo finito. Es bueno, aunque no es necesario, definir los estados inicial y final que normalmente son conocidos.



Representa los cambios de estado que ocurren de manera repentina



Las condiciones y acciones. Las primeras causan un cambio de estado, es decir, un acontecimiento en el ambiente externo que el sistema es capaz de detectar; las segundas son las que el sistema toma cuando cambia un estado generando un dato de salida (almacenador)

Para la construcción de un diagrama de transición de estados se debe comenzar por identificar todos los posibles estados del sistema y representar cada uno como una caja separada en una hoja de papel. Luego, se pueden explorar todas las conexiones, los cambios de estado, entre las cajas. Ya realizado el DTE se deben tener las siguientes consideraciones:

<sup>75</sup> YOURDON. Op cit. Pag 288

- a) Verificar que se hayan especificado todos los estados.
- b) Verificar que es posible producir cada uno de los estados.
- c) Verificar que cada estado tenga salida, excepto en el o los estados finales.
- d) Verificar que en cada estado se hayan especificado todas las condiciones posibles, tanto las normales como las inesperadas.

Se puede ver que las gráficas proporcionan una manera fácil de leer y entender, para que el analista pueda mostrarle a los usuarios los principales componentes del modelo al igual que las conexiones (o interfaces) entre componentes. Y las herramientas de modelado textuales proporcionan definiciones precisas del significado de los componentes y conexiones.

## 2.4.2. Metodología Orientada a Objetos "UML"

La esencia del análisis orientado a objetos es el identificar muchos objetos o conceptos, que representan cosas del mundo real o dominio del problema que se desea resolver, es importante señalar que: en una etapa de análisis se le da prioridad al conocimiento de los requerimientos, los conceptos y las operaciones relacionadas con el sistema<sup>76</sup>. Aquí no se especifican componentes del software, mostrando únicamente: conceptos, asociaciones entre conceptos y atributos de conceptos.

El proceso de análisis UML tiene la composición mostrada en la figura 2.2. Cada uno de los pasos señalados son definidos en esta sección.

1. Se empieza con el **levantamiento de requerimientos**, al igual que en el análisis estructurado se tiene un paso o acción en el que el analista debe captar, asimilar y aclarar cuales son las necesidades del usuario o cliente, identificando con esto la justificación del porque se desarrollará un sistema de información.

Esta actividad no es propia de UML, el requisito es que exprese claramente cuales son la necesidades entendibles tanto para el usuario como el desarrollador.

Proponiéndose las actividades<sup>77</sup>:

- a) Panorama general.- Aquí se definen los objetivos del sistema.
- b) Clientes.- Define el nombre del cliente, y las personas involucradas con el nuevo sistema.
- c) Metas.- Que se desea que haga el sistema.
- d) Funciones del Sistema.- Es una lista de operaciones que indica que va a hacer el sistema, categorizándolas por su significado, proponiendo las siguientes: *Evidente* cuando la función debe realizarse y el usuario debería saber que se ha realizado; *Oculto* cuando debe realizarse pero no se requiere que el usuario tenga conocimiento de ella; y *Superflua* u *opcional* es una función que se puede o no realizar, que no afecta la operación.

---

<sup>76</sup> LARMAN Craig. "UML y Patrones Introducción al análisis y Diseño orientado a objetos". 1a. edición, edición en español. Prentice-Hall Pearson. México 1999. Pag. 161.

<sup>77</sup> LARMAN Craig. Op cit. Pag. 41

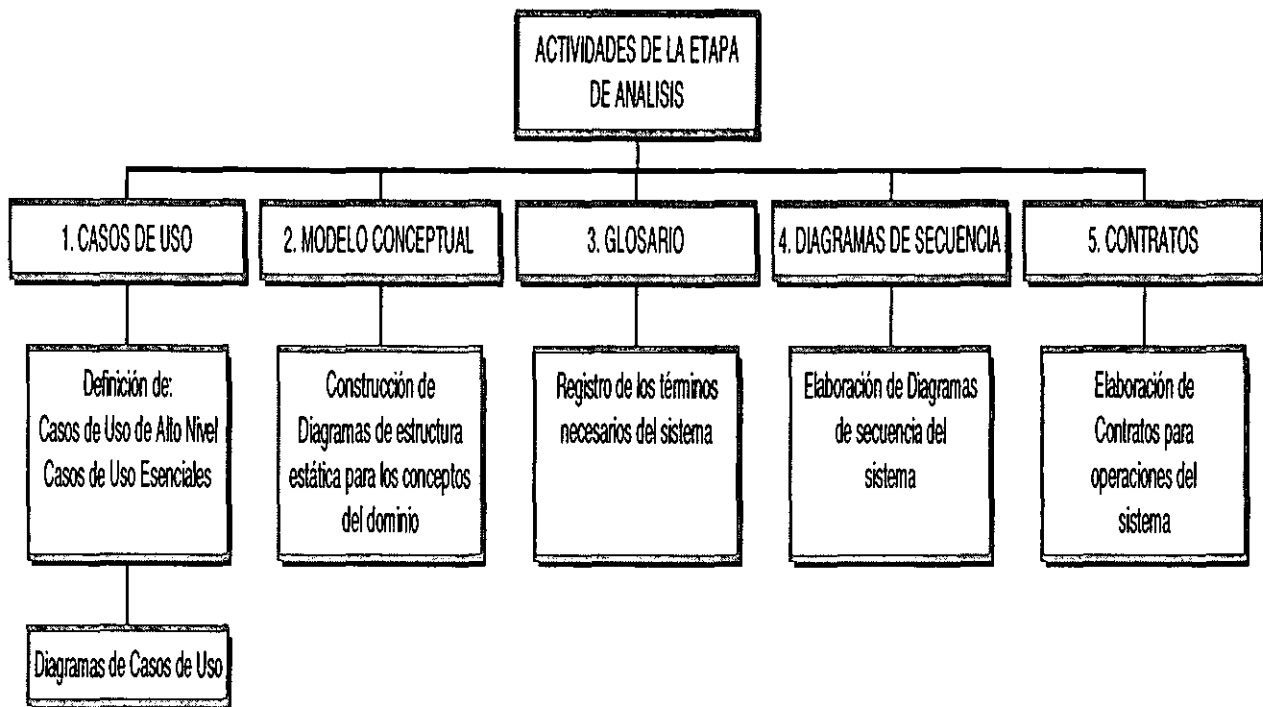


Fig. 2.2<sup>78</sup> Organización del análisis UML

2. Los **Casos de Uso** que son descripciones narrativas, de principio a fin, de los procesos del dominio; señalando la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso<sup>79</sup>; ejemplificando e incluyendo los requerimientos en las historias que narran. Aquí también se establecen las *fronteras*<sup>80</sup> del sistema, estas últimas pueden ser las fronteras hardware/software de un dispositivo o sistema de cómputo; el departamento de una organización; o bien la organización completa.

Los Caso de Uso de *alto nivel* son los primeros en desarrollarse ya que explican de manera clara y concisa los principales procesos globales. Son de carácter esencial<sup>81</sup>, debido a su brevedad y abstracción, describiendo los conceptos en la tabla 2.2

Los conceptos se pueden especificar en cualquier formato especificado por el analista que lo documenta, lo único que se requiere es que transmita claramente lo que se desea interpretar y que se utilice el lenguaje de la demás documentación.

<sup>78</sup> Fuente: Creación Propia

<sup>79</sup> LARMAN Craig. Op cit. Pag. 49

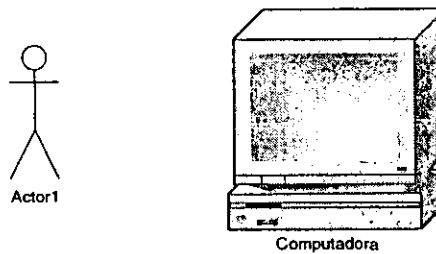
<sup>80</sup> De acuerdo a LARMAN, aquí se identifica lo que es interno o externo, así como responsabilidades del sistema. El ambiente externo está representado únicamente por actores

<sup>81</sup> Adelante se explica los casos de uso esenciales

Concepto	Descripción
Nombre del Caso de Uso	Nombre de preferencia que la primer palabra sea un verbo.
Actores	Lista de actores (agentes externos), en la cual se indica quién inicia el caso de uso.
Tipo <sup>82</sup>	1. Primario, secundario u opcional 2. Esencial o real
Descripción	Descripción breve y precisa de lo que realiza el caso de uso

**Tabla 2.2<sup>83</sup>** Conceptos descriptivos de los casos de uso de alto nivel

El *Actor* es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso; estimulando al sistema a que reaccione ante su petición o que devuelva alguna información. Se debe escribir el nombre del actor con mayúsculas en la narrativa del caso. En el lenguaje UML el actor es representado por alguno de los símbolos de la figura 2.3



**Fig. 2.3<sup>84</sup>** Símbolos Representativos para el Actor de UML

En los Casos de Uso *extendidos* se tiene más detalle que los anteriores, son útiles para obtener un conocimiento más detallado y profundo de los procesos y requerimientos. Presentando los conceptos especificados en la tabla 2.3.

<sup>82</sup> Estos conceptos serán definidos más adelante, en la misma sección

<sup>83</sup> LARMAN Craig. Op cit. Pag. 49

<sup>84</sup> LARMAN G. Op cit. Pag 52



<b>Concepto</b>	<b>Descripción</b>
Caso de Uso	Nombre del caso de uso; de preferencia que la primer palabra sea un verbo en infinitivo.
Actores	Lista de actores (agentes externos), en la cual se indica quién inicia el caso de uso.
Propósito	Intención u objetivo del caso de uso.
Resumen	Descripción sintetizada y precisa de lo que realiza el caso de uso; similar a la descripción que se realiza en los caso de uso de alto nivel.
Tipo	1. Primario, secundario u opcional 2. Esencial o real (ídem del tipo señalado en los caso de uso de alto nivel).
Referencias Cruzadas	Casos relacionados de uso y funciones también relacionadas del sistema.
**** Conceptos esenciales para este tipo de caso de uso:	
Curso normal de los eventos	<p>Descripción de los detalles de la conversión interactiva entre los actores y el sistema; explicando la secuencia más común de los eventos; es decir la historia normal de las actividades y la terminación exitosa de un proceso.</p> <p>Presentando las columnas de: Acción del actor y Respuesta del sistema. En donde en la primera se enumeran las acciones de los actores, (esta enumeración de acuerdo al proceso completo); y en la segunda se describen, igualmente enumeradas, las respuestas del sistema (de acuerdo a la acción del actor).</p> <p>En caso de que alguna acción conduzca a opciones estas deberán listarse en la acción que las ocupe, haciendo su especificación en un sección posterior.</p>
Curso alterno de los eventos	Descripciones importantes de las opciones o excepciones que pueden presentarse en relación con el curso normal. Alternativas que pueden ocurrir.

**Tabla 2.3**<sup>85</sup> Lista de conceptos definidos en los Casos de Uso extendidos

Se debe poner especial atención en no representar como casos de uso a los pasos, las operaciones o las transacciones individuales.

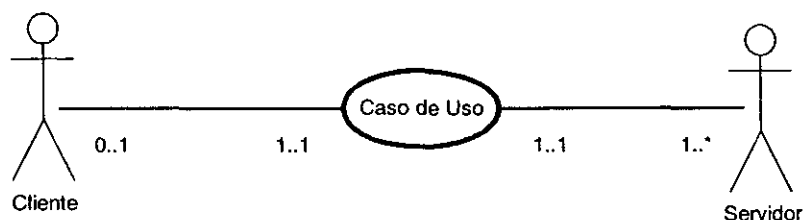
<sup>85</sup> LARMAN G. Op cit. Pag 51

Para identificar los casos de uso, se puede utilizar cualquiera o los dos métodos siguientes<sup>86</sup>:

- En base a los actores:
  1. Se identifican los actores relacionados con un sistema o empresa.
  2. En cada actor, se identifican los procesos que inician o en que participan.
- En base a los eventos:
  1. Se identifican los eventos externos a los que un sistema ha de responder.
  2. Se relacionan los eventos con los actores y con los casos de uso.

Los **Diagramas de los Casos de Uso** explican gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre éstos y los casos de uso. Estos diagramas son comparados con los diagramas contextuales del análisis estructurado, ya vistos en la sección anterior.

Los Casos de uso son representados por óvalos como en la fig. 2.4; y las líneas representan las comunicaciones entre los casos y los actores, con flechas se representa el flujo de información.



**Fig. 2.4**<sup>87</sup> Representación de Casos de Uso

Los **tipos** de casos de uso son<sup>88</sup>:

- 1) *Primarios*. Representan los procesos comunes más importantes.
- 2) *Secundarios*. Representan procesos menores o raros.
- 3) *Opcionales*. Representan procesos que pueden no abordarse.

También se distinguen por:

- 1) *Esenciales*. Son casos expandidos, hacen escasa referencias a tecnología, describiendo el proceso a partir de sus actividades y motivos esenciales.
- 2) *Reales*. Describen concretamente el proceso haciendo uso de la tecnología especificando las entradas y salidas<sup>89</sup>.

<sup>86</sup> LARMAN G. Op cit. Pag. 54

<sup>87</sup> Fuente: Creación Propia

<sup>88</sup> LARMAN G. Op cit. Pag. 58

<sup>89</sup> Su desarrollo se lleva a cabo en la etapa de diseño la cual no se aborda en el presente trabajo, para más información se recomienda el libro de LARMAN parte IV, la bibliografía se encuentra al final del presente proyecto. O cualquier otro libro de UML

3. El **Modelo Conceptual** representa los conceptos<sup>90</sup> en los que se ha dividido el sistema, nunca componentes de software; se crea este modelo para los casos de uso ya generados. Siendo éste ilustrado con diagramas estructurales estáticos donde no se define ninguna operación, mostrando únicamente conceptos<sup>91</sup>, asociaciones entre conceptos y atributos de conceptos. Las estrategias que ayudan a obtener los conceptos son<sup>92</sup>:
- 1) A partir de *una lista de categorías de conceptos* ya definidos en la lista como se tiene en la tabla 2.4, asociándole a cada uno de ellos el concepto que sea pertinente del dominio estudiado.
  - 2) A partir de la *identificación de frases nominales*, consiste en distinguir los conceptos clave a partir de las descripciones textuales incluidas en los casos de uso extendidos revisando el *curso normal de los eventos y cursos alternos*.

CATEGORIA DEL CONCEPTO	ASOCIADO AL PROBLEMA EN ESTUDIO
Objetos físicos o tangibles	<b>CONTENIDO</b>
Especificaciones, diseño o descripciones de cosas	
Lugares	
Transacciones	
Línea o renglón de elemento de transacciones	
Papel de las personas	
Contenedores de otras cosas	
Cosas dentro de un contenedor	
Otros sistemas de cómputo o electromecánicos externos al sistema	
Conceptos de nombres abstractos	
Organizaciones	
Eventos	
Procesos (usualmente no se representan como conceptos)	
Reglas y políticas	
Catálogos	
Registros de finanzas, de trabajo, de contratos de asuntos legales	
Instrumentos y servicios financieros	
Manuales, libros	

**Tabla 2.4<sup>93</sup>** Lista de categorías para la detección de conceptos

<sup>90</sup> El concepto es una idea, cosa u objeto

<sup>91</sup> El termino concepto se relaciona con el termino clase.

<sup>92</sup> LARMAN G. Op cit. Pag. 91

<sup>93</sup> LARMAN G. Op cit. Pag 92

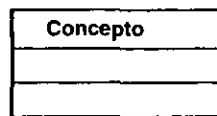
Nota: Se pueden excluir los conceptos que no se relacionen con los requerimientos y que no se encuentren en el dominio.

Los pasos sugeridos para construir un modelo conceptual son<sup>94</sup>:

- 1) Detectar y listar los conceptos
- 2) Dibujarlos
- 3) Incorporar las asociaciones entre los componentes<sup>95</sup>
- 4) Agregar los atributos de los componentes

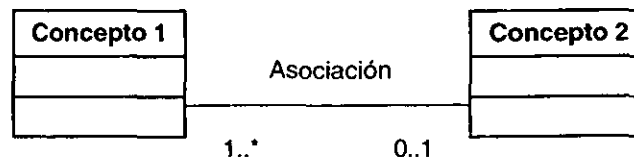
Los nombres de los conceptos y atributos deberán ser bajo el vocabulario del dominio del problema.

La representación gráfica de un concepto es como se presenta en la figura 2.5



**Fig. 2.5<sup>96</sup>** Representación gráfica de un concepto

Larman define a la *Asociación* como una relación entre dos conceptos que indica alguna conexión significativa e interesante entre ellos. Y es representada por una línea bidireccional que puede tener en los extremos algún indicador que especifique el tipo de relación numérica, de multiplicidad<sup>97</sup>, que existe entre los conceptos. La descripción de la asociación puede tener una pequeña flecha que indica la dirección de cómo se debe leer la asociación ver Fig. 2.6



**Fig. 2.6<sup>98</sup>** Muestra de Asociación utilizada en UML

Para encontrar las asociaciones es preciso hacer una lista de las relaciones que existen entre los conceptos. Las asociaciones que siempre deben contemplarse son las del estilo: El Concepto 1 es una parte física o lógica del Concepto 2; el Concepto 1 está física o lógicamente contenido en el Concepto 2; y el Concepto 1 está registrado en el Concepto 2. Se debe de poner atención en las asociaciones en que el conocimiento de la relación ha de preservarse durante algún tiempo, es decir que se requiere conocer de acuerdo a los requerimientos.

<sup>94</sup> LARMAN G. Op cit. Pag 96

<sup>95</sup> Más adelante se hace la definición a los conceptos de asociaciones y atributos

<sup>96</sup> LARMAN G. Op cit. Pag 89

<sup>97</sup> La Multiplicidad define cuántas instancias de un tipo A pueden asociarse a una instancia del tipo B en determinado momento

<sup>98</sup> Fuente: Creación Propia

Un *Atributo* es un valor lógico, o propiedad, de un dato que está contenido en un objeto, en este caso en el concepto como lo muestra la Fig. 2.7. En un modelo conceptual es preferible que los atributos sean atributos simples o valores puros de datos. Se debe tener cuidado en la definición de atributos y conceptos, ya que esto se puede confundir, una buena técnica para diferenciarlos es: detectar si un atributo no es considerado como número o texto, entonces podría ser que este atributo deberá ser un concepto.

Los atributos *simples* son los tipos primitivos de datos como los son: Boolean, fecha, número, texto, hora, etc. Los valores *puros de datos* también conocidos como tipos de datos, los cuales no tienen identidad, para los cuales la identidad única no es significativa. Los *No primitivos* es aquel tipo de dato que es compuesto, se le asocian operaciones, posee otros atributos o es una cantidad con una unidad de medición que puede cambiar.

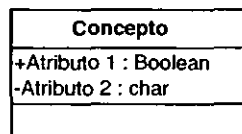


Fig. 2.7<sup>99</sup> Representación de atributos de un objeto

Un buen modelo conceptual capta las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos actuales; nos facilita además conocer el dominio: sus conceptos, su terminología y sus relaciones<sup>100</sup>.

4. El **Glosario** es similar al diccionario de datos manejado en el análisis estructurado<sup>101</sup>. Su realización es durante toda la fase de desarrollo de un sistema, pero es común generarlo en la fase de análisis, junto con la especificación de requerimientos<sup>102</sup>, los casos de uso y el modelo conceptual. En el glosario se registran las reglas del dominio de la empresa, las restricciones, los casos de uso y los contratos.
5. El **Diagrama de Secuencia del Sistema** es la representación gráfica de los eventos que fluyen de los actores<sup>103</sup> al sistema, es decir, las interacciones del actor y de las operaciones que da origen, en determinado escenario<sup>104</sup>. El aislar y explicar gráficamente las operaciones que un actor solicita a un sistema ayuda al entendimiento del comportamiento del sistema.

Los eventos pueden tener parámetros y son acciones externas de entrada que un actor genera en un sistema y la operación es una acción que se ejecuta en respuesta al evento generado. Los eventos y las operaciones deberán tener el mismo nombre.

<sup>99</sup> Idem

<sup>100</sup> LARMAN G. Op cit. Pag 129

<sup>101</sup> El concepto de Diccionario de datos se encuentra especificado en la sección anterior

<sup>102</sup> Los requerimientos están en un documento que describe lo que hace un sistema de software: sus funciones y sus atributos generalmente están escritas desde el punto de vista del usuario.

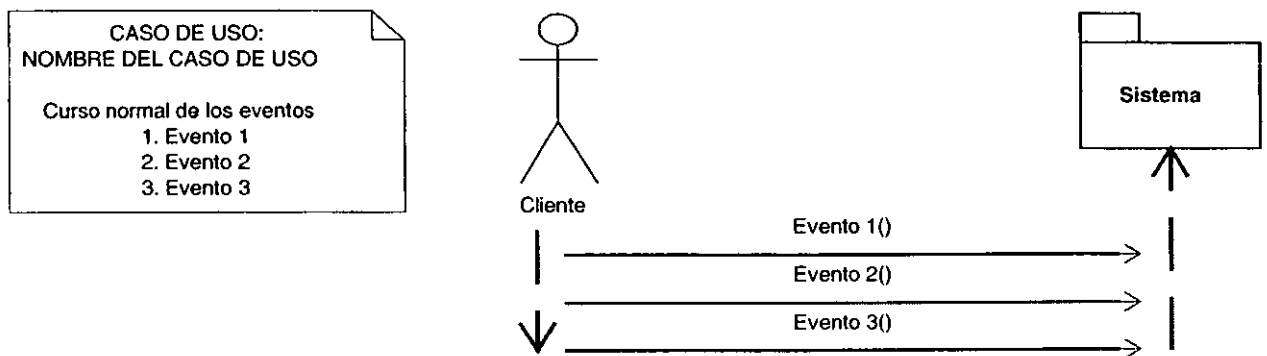
<sup>103</sup> Concepto definido previamente

<sup>104</sup> El escenario es un caso de uso es una instancia o trayectoria realizada por medio del uso: un ejemplo real de su ejecución.

El procedimiento para generar un diagrama de secuencia es el siguiente<sup>105</sup>:

- 1) Trace una línea que represente el sistema como una caja negra.
- 2) Identifique los actores que operan directamente sobre el sistema. Trace una línea para cada uno de ellos.
- 3) A partir del curso normal de los eventos del caso de uso identifique los eventos ("externos") del sistema que son generados por los actores. Muéstrellos gráficamente en el diagrama.
- 4) A la izquierda del diagrama puede incluir o no el texto del caso de uso.

Esperando resulte semejante a lo que muestra en la Fig. 2.8



**Fig. 2.8**<sup>106</sup> Representación del Diagrama de secuencia

6. Los **Contratos** describen el efecto que tienen las operaciones sobre él. El comportamiento de un sistema es una descripción de lo que hace, sin explicar cómo lo hace<sup>107</sup>. Siendo los contratos la herramienta que especifica el comportamiento de un sistema a partir de cómo cambia de estado cuando ha sido invocada una operación de él.

Debe redactarse en un estilo declarativo, enfatizando lo que sucederá y no cómo se conseguirá. Teniendo la estructura como se muestra en la tabla 2.5.

Vale la pena detallar los conceptos de *Poscondiciones* y *Precondiciones*. Las primeras estipulan cómo cambió el sistema tras una operación, son declaraciones de lo que pasa ya concluida una operación. Se deben señalar las que son de tipo: creación y eliminación de las instancias; Las modificación de atributos; y Las asociaciones formadas y canceladas. El UML no define cómo expresar las poscondiciones así que se sugiere elegir el formato que más le agrade y que concuerde con la redacción de los demás documentos. Lo importante es que sea en forma declarativa, orientada al cambio de estado y no a la acción; deberán ser declaraciones sobre los estados o resultados, no una descripción de acciones a realizar. También se deben expresar en tiempo pasado.

<sup>105</sup> LARMAN G. Op cit. Pag 140

<sup>106</sup> Fuente: Creación Propia

<sup>107</sup> LARMAN G. Op cit. Pag 147

Las Precondiciones definen las suposiciones sobre el estado del sistema al iniciarse la operación<sup>108</sup>. Las precondiciones que se requieren ser expuestas son: las cosas de importancia para probar en el software en algún momento de su ejecución y las cosas que no serán sometidas a prueba pero de las cuales depende el éxito de la operación.

<b>Concepto</b>	<b>Descripción</b>
Nombre	Nombre de la operación y parámetros.
Responsabilidades	Descripción informal de las responsabilidades que debe cumplir la operación.
Tipo	Nombre del tipo (concepto, clase de software, interfaz)
Referencias Cruzadas	Número de referencia de las funciones del sistema, casos de uso, etc.
Notas	Notas de diseño, algoritmos e información afín.
Excepciones	Casos excepcionales.
Precondiciones	Suposiciones acerca del estado del sistema antes de ejecutar la operación.
Poscondiciones	Lista de estados del sistema después de la operación.

**Tabla 2.5<sup>109</sup>** Estructura de Contratos

Para la creación de un contrato se debe aplicar los siguientes pasos<sup>110</sup>:

1. Identifique las operaciones del sistema a partir de los diagramas de secuencia.
2. Elabore un contrato en cada operación del sistema.
3. Comience redactando la sección de Responsabilidades; después describa informalmente el propósito de la operación.
4. Complete luego la sección de Poscondiciones, describiendo en forma declarativa los cambios de estado de los objetos en el modelo conceptual.
5. Para describir las poscondiciones utilice las siguientes categorías:
  - Creación y eliminación de instancias.
  - Modificación de los atributos.
  - Asociaciones formadas y canceladas.
6. Utilice la sección de Notas para explicar los detalles del diseño, algoritmos.
7. Use la sección de Excepciones para explicar la reacción ante situaciones raras o especiales.
8. Ponga atención en la inclusión de la formación de asociaciones.

<sup>108</sup> LARMAN G. Op cit pag. 153

<sup>109</sup> LARMAN G. Op cit. Pag 155

<sup>110</sup> LARMAN G. Op cit. Pag 149

## 2.5 Seleccionando una Metodología

Como se vio en las secciones anteriores, las dos metodologías son una gran ayuda para el entendimiento de una aplicación. Por lo que el realizar una correcta elección de alguna de ellas para su implantación, es de suma importancia para el éxito de un sistema de información. Presentando resumidamente en la tabla 2.6 las consideraciones que sirven para una buena elección.

<b>Estructurada Yourdon</b>	<b>Orientada a Objetos UML</b>
<ol style="list-style-type: none"> <li>1. Es la de mayor uso en la ingeniería de software.</li> <li>2. Es de más fácil entendimiento para alguien nuevo en el tema.</li> <li>3. Conduce al diseño de base de datos relacionales las cuales son las más comerciales.</li> <li>4. Existen varias herramientas que automatizan el desarrollo de esta metodología.</li> </ol>	<ol style="list-style-type: none"> <li>1. Es una metodología más completa .</li> <li>2. Lleva más tiempo en su desarrollo.</li> <li>3. Se puede aplicar también para un diseño de base de datos relacionales.</li> <li>4. Como requisito se debe tener conocimientos en programación orientada a objetos, así como de diseño de base de datos orientado a objetos.</li> <li>5. Si se desea aprovechar al máximo sus capacidades se necesita una base de datos orientada a objetos y asegurar que la programación sea de igual manera orientada a objetos.</li> <li>6. Tiene mayor flexibilidad hacia los cambios.</li> <li>7. Da mejor cobertura, en caso de que se desee transportar de una documentación estructurada a una orientada a objetos.</li> <li>8. Reducción de código salvo el buen uso del código reusable.</li> </ol>

**Tabla 2.6<sup>111</sup>** Resumen de consideraciones para la elección de una metodología

Cualquier metodología que se elija correctamente en base a sus necesidades y posibilidades dará un buen resultado, logrando disminuir el tiempo de desarrollo y agrandar las probabilidades de éxito. Se reducen los atrasos en los compromisos fechados hasta en un 85% asignando el 15% restante a los inconvenientes fuera del alcance del uso correcto de alguna de las metodologías.

<sup>111</sup> Fuente: Creación propia



Para hacer una buena elección de qué metodología utilizar se propone tomar en cuenta las limitaciones de:

*Tiempo*; aquí se debe determinar si el tiempo proporcionado es el suficiente para desarrollar una metodología u otra;

*Costos*; al realizar una metodología se invierte tiempo lo que esta directamente relacionado con el dinero que se puede invertir en el desarrollo de una u otra metodología;

*Recursos humanos*; esto es ¿se cuenta con el personal, especifiquemos analista, con el suficiente conocimiento en la metodología que deseamos desarrollar, así como de programación orientada a objetos?;

*Infraestructura*; ver si se cuenta con la tecnología apropiada, refiriéndonos a herramientas computacionales para el desarrollo de cada una de las metodologías; al manejador de base de datos adecuado; y al lenguaje de programación apropiado para la metodología de análisis y diseño seleccionada.

## **Conclusiones del Capítulo II:**

El entender cada uno de los elementos y significados de los pasos requeridos por una metodología de análisis y diseño de sistemas de información es elemental para su buen uso; por lo que en este capítulo se detallaron estos elementos; exponiendo gráfica y textualmente sus componentes para el mejor entendimiento de ambas metodologías, mostrándolos en los puntos 2.3 y 2.4 con todos sus apartados.

La aplicación de UML resulta confusa para aquellos que no tienen conocimiento de la programación orientada a objetos, es por esto que en el presente trabajo apartado 2.2, se dio una explicación sobre éste tipo de programación y se obtuvo que para lograr tener una visión del concepto "objeto" se debe cambiar la manera de ver la situación o el problema en estudio, esto es un objeto se define por lo que es y no por lo que hace.

Por otra parte el análisis estructurado de Yourdon es adaptable para cualquier persona no importando su especialidad, ya que la forma de enfrentar una situación, es similar al pensamiento cotidiano, enfocándose a la creación de procesos o funciones a la que estamos acostumbrados, visto este punto en el apartado 2.1.

Se concluye que ambas metodologías son fáciles de entender pues las actividades que se indican siguen una secuencia lógica, exponiendo los resultados obtenidos en el apartado 2.5. Se cumple entonces el objetivo de éste capítulo el de especificar los pasos requeridos para cada una de las metodologías de análisis y diseño en estudio.

## **FUENTES DE CONSULTA**

### Análisis Estructurado Moderno

Yourdon Edward  
Prentice- Hall Hispanoamericana, S.A.  
México 1993

### Análisis y Diseño Orientado a Objetos Con Aplicaciones

Booch Grady  
Segunda Edición  
Addison-Wesley / Diaz de Santos  
E.U.A. 1996

### Análisis y Diseño de Sistemas de Información

Senn James A.  
Segunda Edición  
Mc Graw Hill  
México, D.F. 1992

### C++ and the OOP Paradigm

Rao Bindu R.  
Cap Gemini America Series  
Mc. Graw Hill  
E.U.A. 1993

### El Lenguaje Unificado de Modelado

Booch Grady, Rumbaugh James, Jacobson Ivar  
Addison Wesley  
Madrid España 1999

### Introducción a la Computación y a la Programación Estructurada

Lenne G. Guillermo  
Segunda Edición  
MC. Graw-Hill  
México, D.F. 1990

### Object-Oriented Modeling and Design

Rumbaugh, James, Michael Blaha  
Prentice Hall  
E.U.A 1991

Proyecto de Investigación I y II: Sistema Integral Genérico Contable

Módulo de Contabilidad

Licenciatura en computación

UAM- Iztapalapa

Tomo I

Olvera Carreño Fernando

Marzo de 1995

UML y Patrones "Introducción al análisis y diseño orientado a objetos"

Larman G.

Prentice-Hall Pearson

México 1999

## **Capítulo III:**

# ***Propuesta de Mejora a la Etapa de Análisis del Procedimiento de Desarrollo de Aplicaciones a la Medida***

*Objetivo: Se aplicará las metodologías de análisis estudiadas, en el capítulo dos, a un caso práctico; se analizarán los resultados; y se presentarán los cambios sugeridos a la etapa de análisis del DAM*

**"UNA VISIÓN SIN ACCIÓN ES UN SUEÑO  
UNA ACCIÓN SIN VISIÓN ES UN PASATIEMPO  
UNA VISIÓN CON ACCIÓN PUEDE CAMBIAR AL MUNDO"**

**-Autor desconocido-**

## **3.1. Aplicación de las Metodologías en un Caso Práctico...**

Continuando con la ejecución de lo planeado en el capítulo 1 inciso 1.11, de como realizar el presente estudio, ahora se realiza la aplicación de cada una de las actividades de la fase de análisis definidas en el capítulo 2, para cada una de las metodologías en estudio, a una *aplicación a la medida* aquí nos referimos al caso práctico definido en el apartado 3.1.1; obteniendo de esto las guías necesarias para su integración al DAM.

### **3.1.1. Caso práctico: Módulo de Verificación de Estructura y Carga de Información de los Archivos Dinámicos que son Reportados a la Comisión Nacional del SAR**

La Comisión Nacional del SAR, nacionalmente conocida como CONSAR, es la entidad federal encargada de regular las entidades involucradas con la inversión y uso de los fondos ahorrados para el retiro, del trabajador mexicano.

¿Como es que la CONSAR regula a las entidades involucradas? empecemos por identificar dichas entidades que son<sup>112</sup>:

- 1) El *trabajador* es la persona que deposita sus fondos de ahorro para el retiro y tiene como característica principal estar afiliado al Instituto Mexicano del Seguro Social.
- 2) El *patrón* es el ente que realiza las aportaciones quincenales obligatorias como lo son: de seguridad social, de vivienda, de ahorro para el retiro.
- 3) La *Afore* que es la Administradora de Fondos para el Retiro, es privada y se caracteriza por tener un alto respaldo financiero, esta entidad debe cumplir con determinados lineamientos especificados a nivel federal algunos de estos son: contar con un mínimo monto capital que pueda soportar posibles irregularidades, tener el respaldo de algún(as) empresa(s) de algún sector financiero, entre otros.
- 4) *Procesar* es la entidad no gubernamental dedicada a la concentración de información correspondiente a cada uno de los trabajadores que tienen una cuenta de ahorro para el retiro; también es responsable de validar la correcta distribución de fondos a las Afores, ganancias a los trabajadores, la validación de los procesos de registro de afiliados en alguna Afore, de aportaciones monetarias al fondo de ahorro, de traspasos de fondos, de retiros y de unificación de cuentas.
- 5) *Instituto Mexicano del Seguro Social (IMSS)*, esta entidad administra las aportaciones realizadas a los trabajadores bajo el motivo de seguridad social.
- 6) El *Instituto del Fondo Nacional de la Vivienda para los Trabajadores (Infonavit)*, es la entidad que administra las aportaciones realizadas a los trabajadores con el concepto de vivienda.
- 7) *Banco Nacional de México* esta entidad es la encargada del resguardo de los fondos concentrados por las aportaciones que hace el patrón hasta que se le solicite una distribución de estos a las Afores.

---

<sup>112</sup> Para mayor información consultar la nueva ley del seguro social, Diario oficial en los temas que hacen referencia este sistema de ahorro para el retiro a partir de 1997

- 8) *Banco Liquidador* es la entidad intermediaria entre Banco de México y las Afores y es el encargado de hacer la transferencia monetaria de los fondos requeridos por la Afore por los conceptos de recaudación, retiro y traspasos.
- 9) *SIEFORE* son sociedades de inversión asociadas a alguna Afore, dedicadas a la inversión de los fondos de ahorro, siempre con el fin de incrementar las ganancias de los trabajadores. El manejo seguro de estos fondos es supervisado mediante un "régimen de inversión" establecido federalmente.

El proceso de validación que hace la CONSAR es mediante la cuadratura de la información que recibe periódicamente de las entidades que regula. Estos datos son transferidos mediante archivos planos que contienen una estructura determinada dependiendo del tipo de información que reporten. Estos archivos deberán cumplir con :

- 1) La estructura, refiriéndose aquí a las posiciones de campos, longitudes, tipo de datos, alineaciones, caracteres válidos, existencia de valores en catálogos; definidos anticipadamente por esta comisión y publicados en el Diario Oficial de la Federación. En caso de no cumplir con los lineamientos, las entidades que incurran a alguna falta serán sancionadas económicamente por la comisión.
- 2) La veracidad de información aquí la comisión se encarga de validar la información reportada, por las entidades relacionadas, verificando que coincidan contable, financiera, y verazmente la información recibida por todos los conceptos que la comisión desee evaluar.

El proceso de cómo se transfiere dicha información es representado por la fig. 3.1

Limitando el presente trabajo al enfoque únicamente de la Verificación y Carga de información reportada en los archivos a Consar. Donde la verificación se refiere a la validación de estructura del archivo, validación contra catálogos, validar que los tipos de datos reportados correspondan a los definidos; y la carga se refiere al almacenamiento de la información reportada en los archivos en una Base de datos.

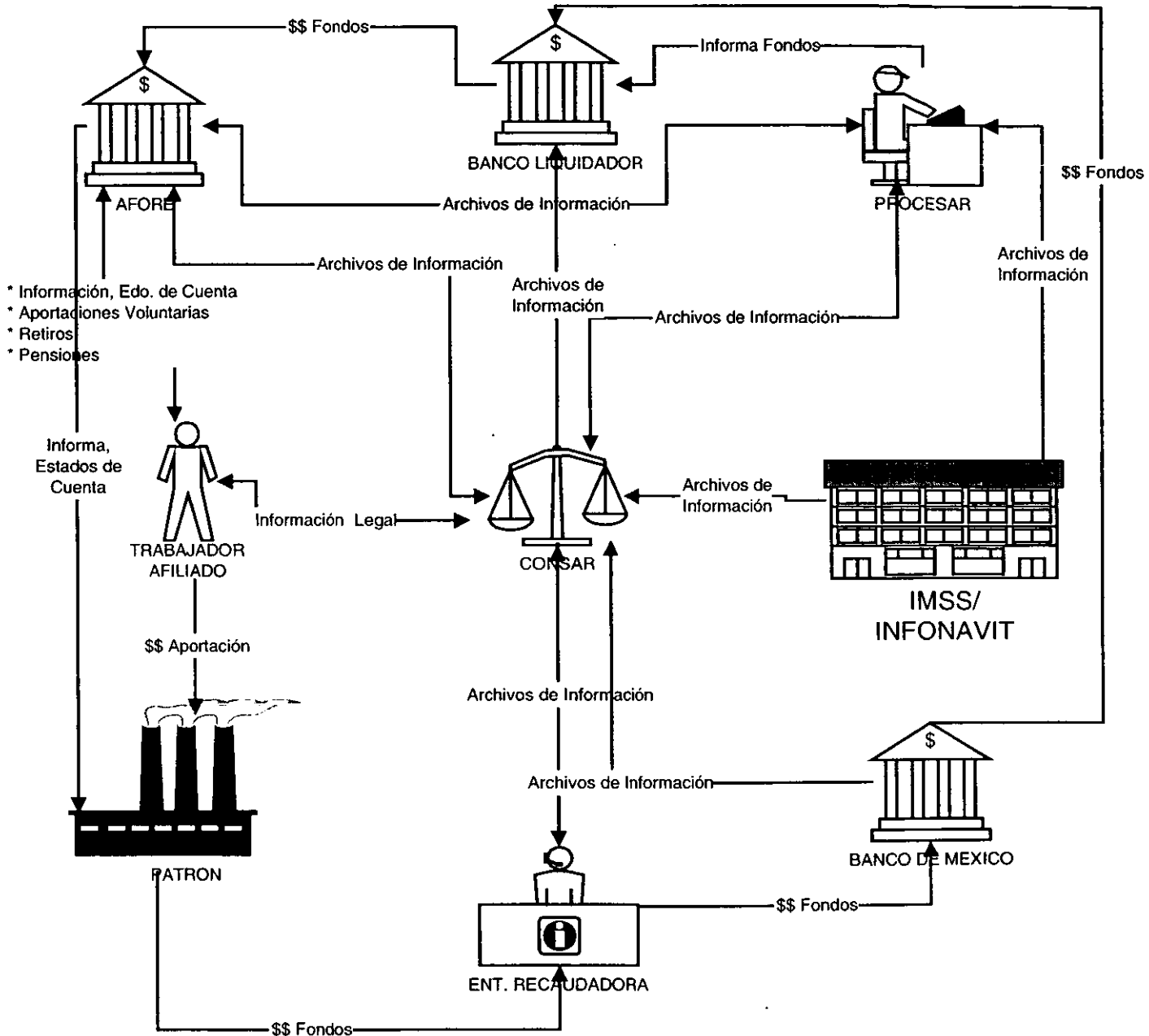


Fig. 3.1<sup>113</sup> Representación gráfica del flujo de información

<sup>113</sup> Fuente: Creación propia



### **3.1.2. Desarrollo de las Etapas del Análisis Estructurado "Yourdon"**

#### **1er. Paso LAS ENCUESTAS**

El resultado de las encuestas y entrevistas se ve reflejado en el primer paso de la metodología UML, "Requerimientos", debido a que en ninguna de las dos metodologías manejan un formato formal.

#### **2do. Paso EL MODELO AMBIENTAL**

Como se definió en el capítulo II el modelo ambiental esta compuesta por: el Propósito, la Lista de Acontecimientos y el Diagrama de Contexto (Fig. 3.2), entonces el modelo ambiental es:

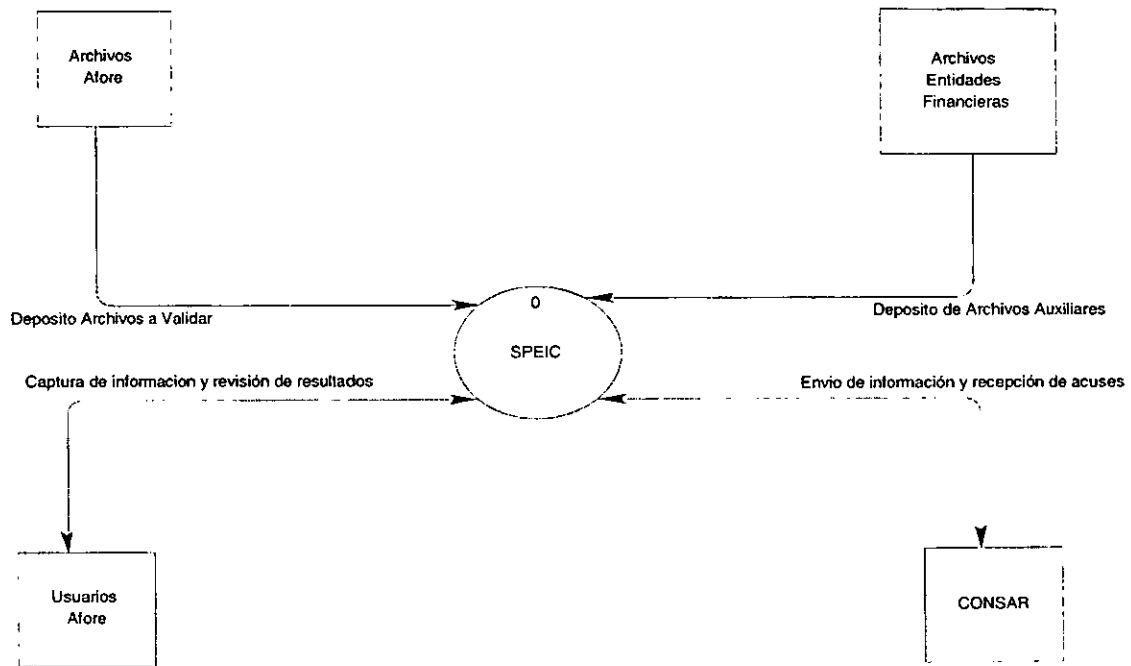
**PROPOSITO:** Realizar la verificación de estructura e información reportada en los archivos que son enviados a CONSAR.

#### **LISTA DE ACONTECIMIENTOS:**

1. Recepción de archivos Aforc
2. Recepción de Archivos Procesar
3. Recepción de Archivos Auxiliares
4. Validación de estructura de los archivos según su tipo
5. Integración de información a la Base de Datos
6. Consulta de Bitácora de Carga de Archivos

**DIAGRAMA DE CONTEXTO...**

Process Model		
Project	: Sistema de Preverificación de Información y Envío a CONSAR	
Model	: DIAGRAMA DE CONTEXTO	
Author	Version	18/07/100



**Fig. 3.2<sup>114</sup>** Diagrama de Contexto del caso práctico

**3er. Paso      DIAGRAMA DE FLUJO DE DATOS**

Los diagramas de flujo de datos obtenidos para el caso práctico se muestran en las figuras 3.3, 3.4 y 3.5. Se remarca que el proceso involucrado con el caso práctico, se limita al proceso 1 del DFD de nivel 1 y los subniveles de este.

<sup>114</sup> Fuente: Creación propia

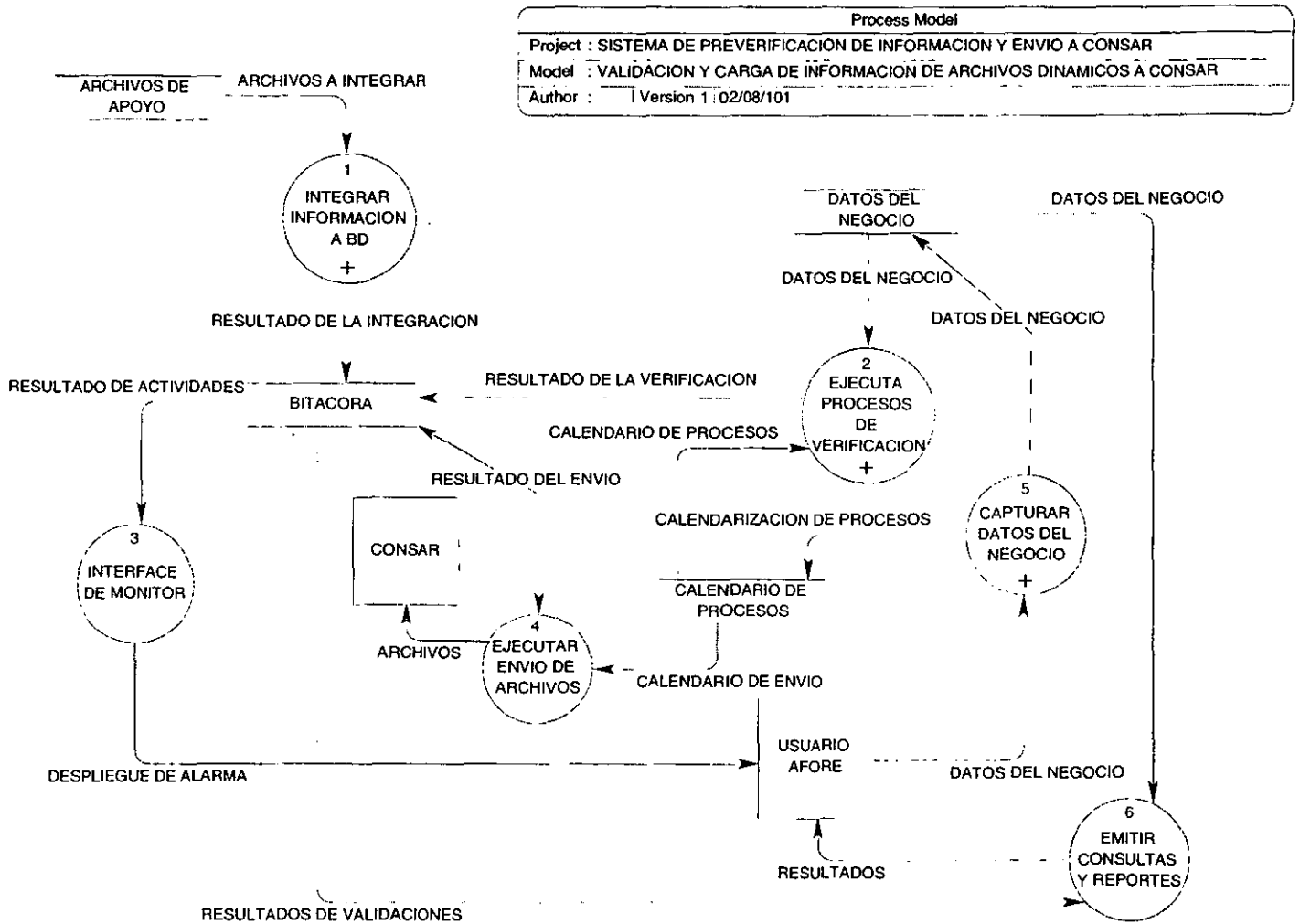


Fig. 3.3<sup>115</sup> Diagrama de Flujo de Datos de nivel 1

<sup>115</sup> Fuente: Creación propia

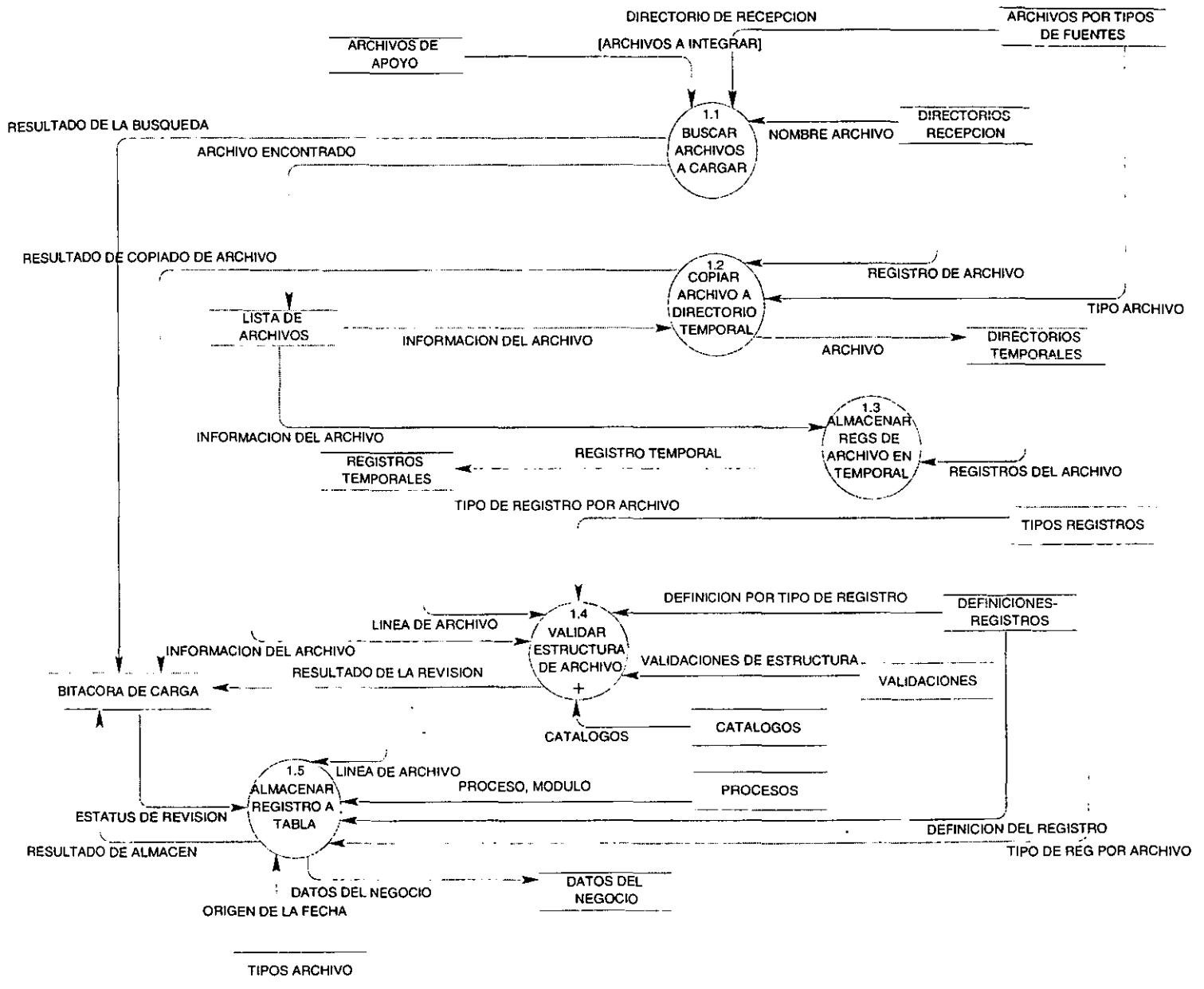


Fig. 3.4<sup>116</sup> Diagrama de Flujo de Datos de nivel 2

<sup>116</sup> Idem

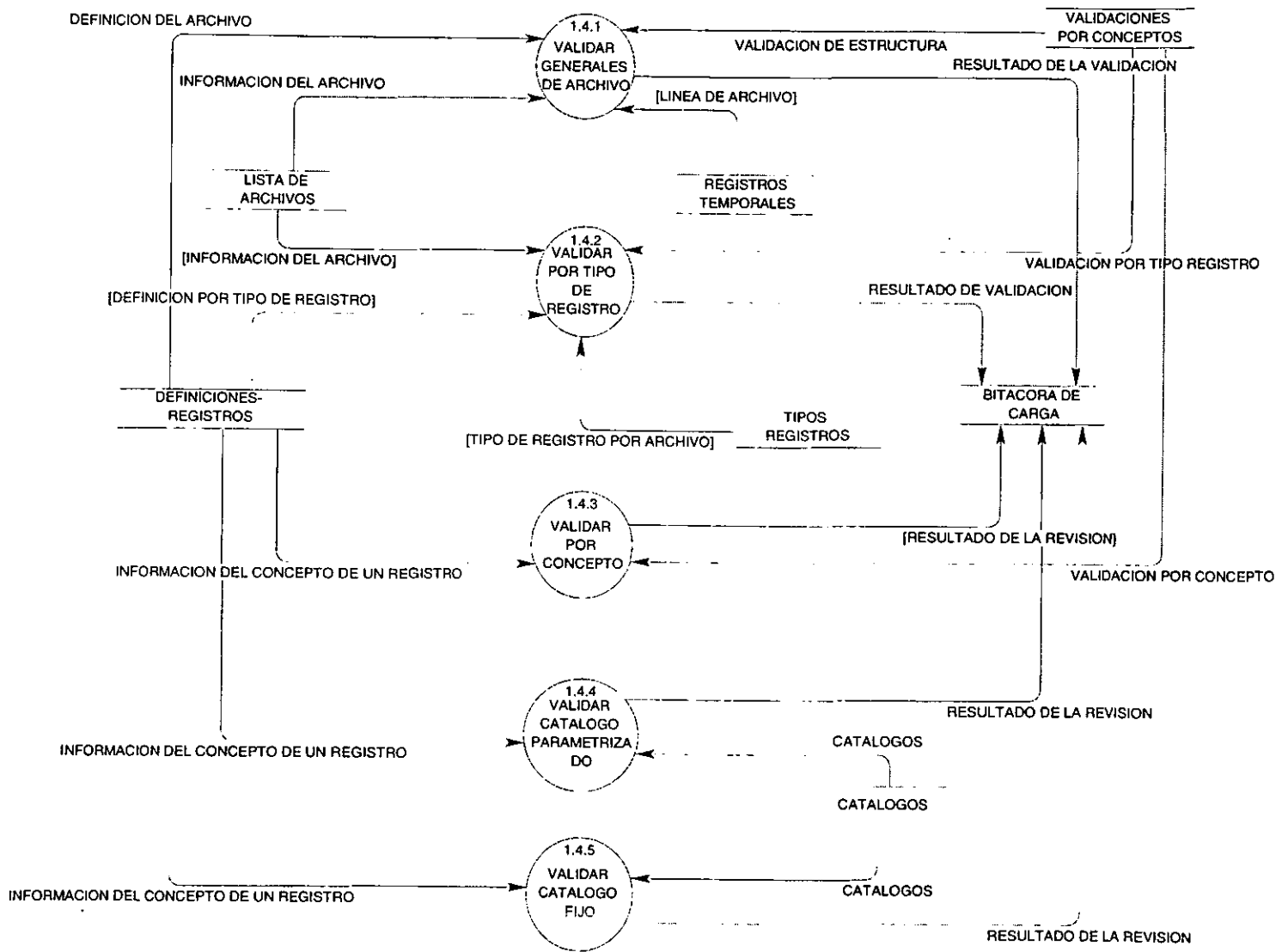


Fig. 3.5<sup>117</sup> Diagrama de Flujo de Datos de Nivel 3

**4to. Paso EL DICCIONARIO DE DATOS**

El Diccionario de datos resultante del presente desarrollo se muestra en la tabla 3.1

<b>Componente</b>	<b>Descripción</b>
ARCHIVOS DE APOYO	*Tiene la relación de los tipos de archivos que auxilian a la validación de la información reportada en los archivos a CONSAR* {Archivos}
DATOS DEL NEGOCIO	*Tiene la información reportada en los archivos * @Tipo de Archivo + @Fecha de Aplicación + @Tipo de Fuente + @Tipo de Registro + @Concepto + @Fecha Inicio de Información + @Fecha Fin de Información + @ClaveDato + DatoString + DatoNumerico
BITACORA DE CARGA	*Tiene el historial de las acciones ocurridas durante el proceso * @Nombre del Archivo + @Fecha + @Hora + Tipo de Archivo + Mensaje + Status
ARCHIVOS POR TIPOS DE FUENTES	*Tiene la relación de los tipos de archivos con los tipos de fuente que se reportan * @Tipo de Fuente + @Tipo de Archivo + @Fecha de Aplicación + Directorio Recepción + Directorio Temporal
DIRECTORIOS RECEPCION	*Tiene las rutas de los directorios donde podrán llegar los archivos * @Tipo de Archivo + @Directorio Recepción + @Tipo de Fuente + Nombre del Archivo + Fecha + Hora
DIRECTORIOS TEMPORALES	*Tiene las rutas de los directorios donde se copiarán los archivos recibidos * @Archivo + @Directorio

**Tabla 3.1**<sup>118</sup> Diccionario de datos de la metodología estructurada Yourdon

<sup>118</sup> Idem

Componente	Descripción
LISTA DE ARCHIVOS	*Tiene los datos de los archivos que se encontraron en el proceso de Buscar Archivos a Cargar * @Tipo de Archivo + @Fecha de Aplicación + @Nombre del Archivo + @Fecha + @Hora + @Directorio Recepción + @Tipo de Fuente + Estatus + Directorio + Consecutivo
REGISTROS TEMPORALES	*Es una entidad que sirve de puente en el proceso de valida archivo y almacena la información de los registros reportados en los archivos * @Consecutivo + Registro
TIPOS DE ARCHIVOS	*Tiene la información general de los tipos de archivos definidos * @Tipo de Archivo + @Fecha de Aplicación + Descripción Archivo + Proceso + Origen de la Fecha [0 1 2 3 4] + Longitud de Archivo
TIPOS DE REGISTROS	*Tiene la información de los tipos de registros por tipo de archivo * @Tipo de Archivo + @Tipo de Registro + @Fecha de Aplicación + Obligatorio [0 1] + Cantidad Obligada + Tabla Fija [0 1] + Tabla Destino
DEFINICIONES REGISTROS	*Tiene la definición de los campos de los tipos de registros por tipo de archivo * @Tipo de Archivo + @Tipo de Registro + @Fecha de Aplicación + @Concepto de Registro + Descripción del Concepto + Tipo de Dato + Posición Inicial + Posición Final + Longitud + Decimales + Clave de Catálogo + Campo de Catálogo + Clave de Subcatálogo + Campo de Subcatálogo + Tabla de Catálogo + Campo de la Tabla + Agrupación + Campo Destino

**Tabla 3.1** Continuación de Diccionario de datos de la metodología estructurada Yourdon

<b>Componente</b>	<b>Descripción</b>
VALIDACIONES (VALIDACIONES POR CONCEPTOS)	*Tiene las validaciones correspondientes por cada campo del tipo de archivo * @Tipo de Archivo + @Tipo de Registro + @Fecha de Aplicación + @Concepto de Registro + Validación + Descripción de la Validación
CATALOGOS	*Representa todos los archivos posibles de validación * @Clave de Catálogo + @Campo de Catálogo + @Clave de Subcatálogo + @Campo de Subcatálogo + @NumRegistro + Dato Catálogo
PROCESOS	*Es el catálogo de Procesos clasificado por módulos* @Modulo [Contable Financiero Operativo] + @Proceso
Archivos a Integrar	*Contiene los archivos que serán validados e integrados a una Base de datos, también contiene a los archivos que auxilian el proceso de validación como son los catálogos*
Directorio de Recepción	*Contiene las rutas del servidor donde se buscan los archivos a integrar* Directorio Recepción
Resultado de la Búsqueda	*Tiene la información de algún archivo encontrado para su integración* "Archivo:" + Nombre del Archivo + "Encontrado creado:" + Fecha + Hora
Archivo Encontrado	*Son los datos de un archivo encontrado para su integración* Tipo de Archivo + Nombre del Archivo + Fecha + Hora + Directorio = Directorio Recepción + Estatus = 1

**Tabla 3.1 Continuación** de Diccionario de datos de la metodología estructurada Yourdon



<b>Componente</b>	<b>Descripción</b>
Nombre Archivo	*Es el nombre del archivo que ha sido depositado en algún directorio de recepción* Nombre del Archivo
Resultado de Copiado de Archivo	*Es la información de si el archivo encontrado ha sido copiado o no al directorio temporal* Tipo de Archivo + Fecha de Aplicación + Tipo de Fuente + Archivo Consecutivo+ Nombre del Archivo + Fecha + Hora + Directorio = Directorio Temporal + Directorio Recepción = Directorio Recepción + Estatus = 2
Registro de Archivo	*Es el archivo encontrado y que es copiado al directorio temporal* Nombre de Archivo
Tipo Archivo	*Indica el tipo de archivo de acuerdo a su entidad origen y su directorio recepción* Tipo de Archivo
Archivo	*Es el archivo que es copiado al directorio temporal*
Información del Archivo	*Es la información del archivo que es encontrado y será validado para su integración* Tipo de Archivo + Nombre del Archivo + Fecha de Aplicación + Fecha + Hora + Directorio * Tipo de Fuente
Registro Temporal	*Es un registro (renglón o línea) del archivo y es almacenado temporalmente* Registro
Línea de Registro	*Es una línea (renglón) del archivo a validar* Registro
Registros del Archivo	*Son cada una de las líneas del archivo a integrar* Registro

**Tabla 3.1 Continuación** de Diccionario de datos de la metodología estructurada Yourd

<b>Componente</b>	<b>Descripción</b>
Tipo de Registro por Archivo	*Tiene los tipos de registros definidos para el tipo de archivo que se va a integrar* Tipo de Registro + Cantidad Obligada + Tabla Fija + Tabla Destino
Definición por tipo de registro	*Es la definición de los campos o conceptos que deben reportarse en los tipos de registro definidos en el tipo de archivo a integrar* Tipo de Registro + Concepto de Registro + Posición Inicial + Tipo de Dato + Longitud + Decimales + Tabla Fija + Tabla Destino + Campo Destino
Resultado de la Revisión	*Tiene la información detalle resultante de la validación de la estructura del archivo a integrar* Mensaje + Nombre del Archivo + Fecha + Hora + Estatus [5 6]
Validación de Estructura	*Es la validación general del archivo a integrar* Tipo de Registro + Concepto de Registro + Posición Inicial + Longitud + [Caracteres de Registro   Encabezado   Cantidad de Registros   Tipo de Archivo   Sumario]
Catálogos	*Contiene la información de los catálogos que auxilian a la validación del archivo a integrar, esta información puede ser de catálogos "fijos" o "parametrizados" *
Estatus de Revisión	*Tiene el estatus del archivo resultante de la validación del archivo* Tipo de Archivo + Nombre del Archivo + Fecha de Aplicación + Fecha + Hora + Tipo de Fuente + Estatus = [5 6]
Proceso, Modulo	*Tiene los datos del Proceso y Módulo al que pertenece el Tipo de Archivo a Integrar*

**Tabla 3.1 Continuación** de Diccionario de datos de la metodología estructurada Yourd

<b>Componente</b>	<b>Descripción</b>
Resultado de Almacén	*Tiene el resultado del almacenamiento de información del archivo a integrar* Mensaje + Nombre del Archivo + Fecha + Hora + Estatus [8 9]
Definición del Registro	*Es la definición de los campos o conceptos que deben reportarse en el tipo de registro a validar* Tipo de Registro + Concepto de Registro + Posición Inicial + Tipo de Dato + Longitud + Decimales + Tabla Fija + Tabla Destino + Campo Destino
Origen de la Fecha	*Indica de donde se toma la fecha de información reportada en el archivo a integrar* Origen de la Fecha = [Fecha de Encabezado   Fecha de Sumario   Periodo Información Encabezado   Periodo Información Sumario]
Datos del Negocio	*Es la información clasificada, reportada en el archivo a integrar*
Definición del Archivo	*Es la definición de los campos o conceptos que deben reportarse en el tipo de registro a validar* Tipo de Registro + Concepto de Registro + Posición Inicial + Tipo de Dato + Longitud + Decimales + [Caracteres de Registro   Encabezado   Cantidad de Registros   Tipo de Archivo   Sumario]
Resultado de la Validación	*Tiene la información detalle resultante de la validación de la estructura del archivo a integrar* Mensaje + Nombre del Archivo + Fecha + Hora + Estatus [5 6]
Validación por Tipo Registro	*Contiene las validaciones propias al tipo de registro (renglón) correspondiente a cada una de las líneas del archivo a revisar* Tipo de Registro + Cantidad Obligada

**Tabla 3.1 Continuación** de Diccionario de datos de la metodología estructurada Yourd

<b>Componente</b>	<b>Descripción</b>
Validación por Concepto	*Contiene las validaciones propias de cada uno de los conceptos (campos) del archivo a validar* Tipo de Registro + Concepto de Registro + Posición Inicial + Tipo de Dato + Longitud + Decimales + {Validación}
Información del Concepto de un Registro	*Es la valor reportado en un campo o concepto del archivo a validar* Tipo de Archivo + Fecha de Aplicación + Fecha + Hora + Tipo de Fuente + Tipo de Registro + Concepto + Tipo de Dato + Posición Inicial + Longitud + Decimales + Fecha Inicio + Fecha Fin + Clave de Dato + Tabla Fija + Tabla Destino + Campo Destino

**Tabla 3.1 Continuación** de Diccionario de datos de la metodología estructurada Yourd

## **5to. Paso            ESPECIFICACIONES DE PROCESOS**

Vale la pena comentar que la enumeración puesta en los siguientes procesos es independiente a la enumeración general, ya que aquí se asocia la descripción de los procesos con los diagramas de flujo desarrollados con anterioridad en el paso 3.

### **1.1            BUSCAR ARCHIVOS A CARGAR**

Descripción: Almacenar en LISTA DE ARCHIVOS los nombres de los archivos que fueron encontrados en los diferentes directorios de recepción.

1. Buscar en DIRECTORIOS RECEPCION Archivo
2. Si se encontró alguno  
 Obtener del Archivo el Tipo de Archivo, Nombre del Archivo, Fecha, Hora  
 Fijar Directorio a Nombre del Directorio Recepción  
 Fijar Estatus a Encontrado  
 Guardar en LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha, Hora, Estatus, Directorio  
 Guardar en BITACORA DE CARGA un mensaje de "Archivo: **Nombre del Archivo** encontrado  
 Creado: **Fecha, Hora"**

#### **NOTAS:**

1. Encontrado = 1

## **1.2 COPIAR ARCHIVO A DIRECTORIO TEMPORAL**

Descripción: Mover el Archivo encontrado de DIRECTORIOS RECEPCION a DIRECTORIOS TEMPORALES.

1. Leer de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha, Hora, Directorio con

Estatus = Encontrado y

Mínima Fecha y

Mínima Hora

Para Cada Archivo encontrado

1.1. Buscar en ARCHIVOS POR TIPOS DE FUENTES Fecha de Aplicación, Tipo de Fuente, Directorio Recepción, Directorio Temporal con

Tipo Archivo = Tipo de Archivo

Directorio Recepción = Directorio

1.2. Si se encontró

Obtener de LISTA DE ARCHIVOS Consecutivo con

Nombre = Nombre del Archivo

Acumular Consecutivo + 1

Renombrar Archivo a Archivo y Consecutivo

Mover de Directorio Recepción a Directorio Temporal

el Archivo y Consecutivo con

Nombre = Nombre del Archivo y

Fecha = Fecha y

Hora = Hora

Actualizar de LISTA DE ARCHIVOS Estatus = Archivo copiado,

Directorio = Directorio Temporal,

Fecha de Aplicación = Fecha de Aplicación,

Tipo de Fuente = Tipo de Fuente

Consecutivo = Consecutivo con

Tipo de Archivo y

Nombre del Archivo y

Fecha y

Hora y

Directorio Recepción y

Estatus = Encontrado

1.3. Si no

Guardar en BITACORA DE CARGA un mensaje de "Archivo: **Nombre del Archivo Fecha y Hora** no válido, encontrado en el Directorio: **Directorio Recepción**"

### **NOTAS:**

1. Encontrado = 1

2. Archivo copiado = 2

### 1.3 ALMACENAR REGS DE ARCHIVO EN TEMPORAL

Descripción: Guardar en la entidad REGISTROS TEMPORALES la información contenida en un archivo

1. Leer de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha, Hora, Directorio con

Estatus = Archivo copiado y

Mínima Fecha y

Mínima Hora

Para Cada Archivo encontrado

Fijar Consecutivo a 0

Repetir

Obtener de DIRECTORIOS TEMPORALES Registro Con

Archivo = Nombre del Archivo y

Directorio

Acumular Consecutivo + 1

Almacenar en REGISTROS TEMPORALES Consecutivo, Registro

Hasta Fin de Archivo

Actualizar de LISTA DE ARCHIVOS Estatus = Archivo Almacenado en Temporal con

Tipo de Archivo y

Nombre del Archivo y

Fecha y

Hora y

Directorio y

Estatus = Archivo copiado

#### NOTAS:

- |                                   |     |
|-----------------------------------|-----|
| 1. Encontrado                     | = 1 |
| 2. Archivo copiado                | = 2 |
| 3. Archivo Almacenado en Temporal | = 3 |

#### **1.4 VALIDAR ESTRUCTURA DE ARCHIVO**

Descripción: Validar la estructura de un Archivo, de acuerdo a su definición

##### **1.4.1 VALIDAR GENERALES DE ARCHIVO**

Descripción: Realizar las validaciones generales de estructura de un archivo

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación,  
Fecha y Hora, Directorio, Tipo de Fuente con  
Estatus = Archivo Almacenado en Temporal
2. Actualizar de LISTA DE ARCHIVOS Estatus = Validando Estructura del Archivo con  
Tipo de Archivo y  
Nombre del Archivo y  
Fecha y  
Hora
3. Guardar en BITACORA DE CARGA un mensaje de "Archivo: **Nombre del Archivo Fecha y Hora** se encuentra con estatus: **Validando estructura del archivo**"
4. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS  
Tipo de Registro, Posición Inicial, Longitud con  
VALIDACIONES POR CONCEPTO.Tipo de Registro =  
DEFINICIONES-REGISTROS.Tipo de Registro y  
VALIDACIONES POR CONCEPTO.Concepto de Registro =  
DEFINICIONES-REGISTROS.Concepto de Registro  
Tipo de Archivo y  
Fecha de Aplicación y  
Validación = Encabezado
5. Si No Existe Tipo de Registro  
Fijar EstatusArchivo a Validación Estructura con Error  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora** no tiene asignado el encabezado para su tipo de Archivo"
6. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
7. Leer de REGISTRO TEMPORAL Consecutivo con  
Primeros Longitud de Tipo Registro- Caracteres de Registro = Tipo de Registro  
Acumular Cantidad de Encabezados + 1
8. Si Cantidad de Encabezados <> 1  
Fijar EstatusArchivo a Validación Estructura con Error  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo:  
**Nombre del Archivo Fecha y Hora** tiene más de un  
encabezado o no tiene encabezado"
9. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS  
Tipo de Registro, Posición Inicial, Longitud con  
VALIDACIONES POR CONCEPTO.Tipo de Registro =  
DEFINICIONES-REGISTROS.Tipo de Registro y  
VALIDACIONES POR CONCEPTO.Concepto de Registro =  
DEFINICIONES-REGISTROS.Concepto de Registro  
Tipo de Archivo y  
Fecha de Aplicación y  
Validación = Cantidad de Registros

10. Si No Existe Tipo de Registro
  - Fijar EstatusArchivo a Validación Estructura con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora** no tiene asignado Cantidad de Registros para su tipo de Archivo"
11. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
12. Obtener de REGISTROS TEMPORALES Registro con
  - Primeros Longitud de Tipo Registro- Caracteres de Registro = Tipo de Registro
13. Fijar Cantidad de Registros a Extraer de Registro los Longitud- caracteres desde Posición
14. Obtener de REGISTROS TEMPORALES el Máximo de Consecutivo
15. Si Cantidad de Registros <> Máximo de Consecutivo
  - Fijar EstatusArchivo a Validación Estructura con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora**, no corresponde la cantidad de registros, los esperados: **Cantidad de Registros**, los reportados: **Máximo de Consecutivo**"
16. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS
  - Tipo de Registro, Posición Inicial, Longitud con
  - VALIDACIONES POR CONCEPTO.Tipo de Registro =
  - DEFINICIONES-REGISTROS.Tipo de Registro y
  - VALIDACIONES POR CONCEPTO.Concepto de Registro =
  - DEFINICIONES-REGISTROS.Concepto de Registro
  - Tipo de Archivo y
  - Fecha de Aplicación y
  - Validación = Tipo de Archivo
17. Si No Existe Tipo de Registro
  - Fijar EstatusArchivo a Validación Estructura con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora** no tiene asignado el Tipo de Archivo para su tipo de Archivo"
18. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
19. Obtener de REGISTROS TEMPORALES Registro con
  - Primeros Longitud de Tipo Registro- Caracteres de Registro = Tipo de Registro
20. Fijar Tipo Archivo a Extraer de Registro los Longitud- caracteres desde Posición
21. Si Tipo Archivo <> Tipo de Archivo
  - Fijar EstatusArchivo a Validación Estructura con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora**, no corresponde el tipo de Archivo reportado: **Tipo Archivo**"
22. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS
  - Tipo de Registro, Posición Inicial, Longitud con
  - VALIDACIONES POR CONCEPTO.Tipo de Registro =
  - DEFINICIONES-REGISTROS.Tipo de Registro y
  - VALIDACIONES POR CONCEPTO.Concepto de Registro =
  - DEFINICIONES-REGISTROS.Concepto de Registro y
  - Tipo de Archivo y
  - Fecha de Aplicación y
  - Validación = Longitud de Registro
23. Si No Existe Tipo de Registro
  - Fijar EstatusArchivo a Validación Estructura con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo Fecha y Hora** no tiene asignado la Longitud de Registros para su tipo de Archivo"



24. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
25. Obtener de REGISTROS TEMPORALES Registro con  
    Primeros Longitud de Tipo Registro- Caracteres de Registro = Tipo de Registro
26. Fijar Longitud de Registro a Extraer de Registro los Longitud- caracteres desde Posición
27. Leer REGISTROS TEMPORALES Registro  
    Para cada Registro hacer:  
        Calcular Longitud Registro = Longitud del Registro  
        Si Longitud Registro <> Longitud de Registro  
            Fijar EstatusArchivo a Validación Estructura Con Error  
            Guardar en BITACORA DE CARGA un mensaje de "El Archivo:  
                **Nombre del Archivo, Fecha y Hora** no corresponde  
                en longitud en el registro: **Consecutivo**, los esperados:  
                **Longitud de Registro**, los reportados: **Longitud  
                Registro"**
28. Si EstatusArchivo = Validación Estructura con Error  
    Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura con Error con  
        Tipo de Archivo y  
        Nombre del Archivo y  
        Fecha y  
        Hora  
    Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del  
        Archivo, Fecha y Hora**, Presento las validaciones generales de  
        Archivo Con Error"
- Si no  
        Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura Sin Error con  
            Tipo de Archivo y  
            Nombre del Archivo y  
            Fecha y  
            Hora  
        Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del  
            Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo  
            sin Error"

#### **1.4.2 VALIDAR POR TIPO DE REGISTRO**

Descripción: Realizar las validaciones por Tipo de Registro

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación,  
    Tipo de Fuente, Fecha y Hora , Directorio con  
    Estatus = Archivo Almacenado en Temporal
2. Leer TIPOS REGISTROS Tipo Registro, Cantidad Obligada con  
    Tipo de Archivo y  
    Fecha de Aplicación y  
    Obligatorio = 1  
    Hacer lo siguiente para Cada registro
- 2.1. Seleccionar de REGISTROS TEMPORALES Acumular Cantidad de Registros +1 con  
    Primeros Longitud de Tipo Registro- Caracteres de Registro = Tipo Registro

- 2.2. Si Cantidad de Registros < Cantidad Obligada
  - Fijar EstatusArchivo a Validación Estructura Con Error
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** no cumple con la cantidad de: **Cantidad Obligada**, Registros Obligatorios"
3. Leer REGISTROS TEMPORALES Consecutivo, Registro
  - Hacer lo siguiente para cada Registro
  - 3.1. Fijar Tipo de Registro a Primeros Longitud de Tipo Registro- Caracteres de Registro
  - 3.2. Buscar en TIPOS REGISTROS Tabla Fija, Tabla Destino con
    - Tipo de Archivo y
    - Fecha de Aplicación Y
    - Tipo de Registro
  - 3.3. Si no existe
    - Fijar EstatusArchivo a Validación Estructura Con Error
    - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** tiene un Tipo de Registro no válido: **Tipo de Registro**, en el Registros: **Consecutivo**"
4. Si EstatusArchivo = Validación Estructura con Error
  - Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura con Error con
    - Tipo de Archivo y
    - Nombre del Archivo y
    - Fecha y
    - Hora
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo Con Error"
- Si no
  - Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura Sin Error con
    - Tipo de Archivo y
    - Nombre del Archivo y
    - Fecha y
    - Hora
  - Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo sin Error"

#### **1.4.3 VALIDAR POR CONCEPTO**

Descripción: Realizar las validaciones por Concepto

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación,
  - Tipo de Fuente con
  - Estatus = Archivo Almacenado en Temporal
2. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS
  - Tipo de Registro, Posición Inicial, Longitud con
  - VALIDACIONES POR CONCEPTO.Tipo de Registro =
  - DEFINICIONES-REGISTROS.Tipo de Registro y
  - VALIDACIONES POR CONCEPTO.Concepto de Registro =
  - DEFINICIONES-REGISTROS.Concepto de Registro y
  - Tipo de Archivo y
  - Fecha de Aplicación y
  - Validación = Encabezado

3. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
4. Leer de REGISTROS TEMPORALES Consecutivo, Registro  
Hacer lo siguiente para cada Registro
  - 4.1. Fijar Tipo de Registro a los Primeros Longitud de Tipo Registro- Caracteres de Registro
  - 4.2. Leer de DEFINICIONES REGISTROS Concepto de Registro, Tipo de Dato, Longitud, Decimales, Posición Inicial con Tipo de Archivo y Fecha de Aplicación y Tipo de Registro  
Hacer lo siguiente para cada Registro  
Fijar Dato a los Longitud-caráctes de Registro desde Posición Inicial  
Validar Dato de acuerdo a su Tipo de Dato  
Si No es Valido  
Fijar EstatusArchivo a Validación Estructura con Error  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** registro: **Consecutivo** no cumple con lo especificado para su Tipo de Dato"  
Leer de VALIDACIONES POR CONCEPTO Validación con Tipo de Archivo y Fecha de Aplicación y Tipo de Registro y Concepto de Registro  
Hacer lo siguiente para cada Registro  
Realizar la Validación a Dato  
Si No es Valido  
Fijar EstatusArchivo a Validación Estructura con Error  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** registro: **Consecutivo** no cumple con lo especificado para la validación: **Validación**"
5. Si EstatusArchivo = Validación Estructura con Error  
Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura con Error con Tipo de Archivo y Nombre del Archivo y Fecha y Hora  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo Con Error"  
Si no  
Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura Sin Error con Tipo de Archivo y Nombre del Archivo y Fecha y Hora  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo sin Error"

#### **1.4.4 VALIDAR CATALOGO PARAMETRIZADO**

Descripción: Realizar las validaciones por Concepto que tiene asociado un catalogo de tipo parametrizado (catálogos dinámicos que pueden ser definidos por el usuario)

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación, Tipo de Fuente con Estatus = Archivo Almacenado en Temporal
2. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS Tipo de Registro, Posición Inicial, Longitud con VALIDACIONES POR CONCEPTO.Tipo de Registro = DEFINICIONES-REGISTROS.Tipo de Registro y VALIDACIONES POR CONCEPTO.Concepto de Registro = DEFINICIONES-REGISTROS.Concepto de Registro y Tipo de Archivo y Fecha de Aplicación y Validación = Encabezado
3. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
4. Leer de REGISTROS TEMPORALES Consecutivo, Registro Hacer lo siguiente para cada Registro
  - 4.1. Fijar Tipo de Registro a los Primeros Longitud de Tipo Registro- Caracteres de Registro
  - 4.2. Leer de DEFINICIONES REGISTROS Concepto de Registro, Tipo de Dato, Longitud, Decimales, Posición Inicial, Clave de Catalogo, Campo de Catalogo, Clave de SubCatálogo, Campo de SubCatálogo con Tipo de Archivo y Fecha de Aplicación y Tipo de Registro Hacer lo siguiente para cada Registro Fijar Dato a los Longitud-caráctes de Registro desde Posición Inicial Si Clave de SubCatálogo No Existe BUSCAR en CATALOGOS NumRegistro con Clave de Catalogo y Campo de Catalogo Y Dato Catálogo = Dato Si no existe Fijar EstatusArchivo a Validación Estructura con Error Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** registro: **Consecutivo** el concepto: **Concepto de Registro** no existe en catálogo"
    - Si No BUSCAR en CATALOGOS NumRegistro con Clave de Catalogo y Campo de Catalogo y Clave de SubCatálogo y Campo de SubCatálogo y Dato Catálogo = Dato Si no existe Fijar EstatusArchivo a Validación Estructura con Error Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora** registro: **Consecutivo** el concepto: **Concepto de Registro** no existe en Subcatálogo"

5. Si EstatusArchivo = Validación Estructura con Error  
Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura con Error con  
Tipo de Archivo y  
Nombre del Archivo y  
Fecha y  
Hora  
Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo Con Error"

Si no

- Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura Sin Error con  
Tipo de Archivo y  
Nombre del Archivo y  
Fecha y  
Hora

#### **1.4.5 VALIDAR CATALOGO FIJO**

Descripción: Realizar las validaciones por Concepto

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación,  
Tipo de Fuente con  
Estatus = Archivo Almacenado en Temporal
2. Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS  
Tipo de Registro, Posición Inicial, Longitud con  
VALIDACIONES POR CONCEPTO.Tipo de Registro =  
DEFINICIONES-REGISTROS.Tipo de Registro y  
VALIDACIONES POR CONCEPTO.Concepto de Registro =  
DEFINICIONES-REGISTROS.Concepto de Registro y  
Tipo de Archivo y  
Fecha de Aplicación y  
Validación = Encabezado
3. Calcular Longitud de Tipo Registro = Longitud de Tipo de Registro
4. Leer de REGISTROS TEMPORALES Consecutivo, Registro  
Hacer lo siguiente para cada Registro
  - 4.1. Fijar Tipo de Registro a los Primeros Longitud de Tipo Registro-  
Caracteres de Registro
  - 4.2. Leer de DEFINICIONES REGISTROS Concepto de Registro, Descripción del  
Concepto, Tipo de Dato, Longitud, Decimales, Posición Inicial,  
Tabla de Catalogo, Campo de la Tabla, Agrupación con  
Tipo de Archivo y  
Fecha de Aplicación y  
Tipo de Registro y  
Que Tabla de Catalogo exista  
Ordenado por Concepto de registro, Tabla de Catalogo, Agrupación  
Hacer lo siguiente para cada Concepto de Registro  
Fijar Dato a los Longitud-caráctes de Registro desde Posición Inicial

Si Tabla de Catalogo = Tabla Auxiliar y Agrupación > 1  
Acumular QuerySelect + "y" + Campo de la Tabla  
+ "=" + Dato

Si no

Si QuerySelect No esta vacío

Ejecuta QuerySelect

Si la ejecución no fue exitosa

Fijar EstatusArchivo a Validación Estructura con Error

Guardar en BITACORA DE CARGA un mensaje de "El

Archivo: **Nombre del Archivo, Fecha y Hora** registro:

**Consecutivo** el concepto: **Descripción del Concepto de Registro** no existe en el Catálogo: **Tabla de Catalogo"**

Acumular QuerySelect + "Contar los registros de: " + Tabla de Catalogo + "con" + Campo de la Tabla + "=" + Dato

Fijar Tabla Auxiliar a Tabla de Catalogo

5. Si EstatusArchivo = Validación Estructura con Error

Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura con Error con

Tipo de Archivo y

Nombre del Archivo y

Fecha y

Hora y

Directorio

Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento las validaciones generales de Archivo Con Error"

Si no

Actualizar de LISTA DE ARCHIVOS Estatus = Validación Estructura Sin Error con

Tipo de Archivo y

Nombre del Archivo y

Fecha y

Hora

**NOTAS:**

- |                                       |      |
|---------------------------------------|------|
| 1. Archivo Almacenado en Temporal     | = 3  |
| 2. Validando estructura del archivo   | = 4  |
| 3. Validación de Estructura con Error | = 5  |
| 4. Validación de Estructura sin Error | = 6  |
| 5. Longitud de Registro               | = 9  |
| 6. Cantidad de Registros              | = 10 |
| 7. Encabezado                         | = 4  |
| 8. Sumario                            | = 6  |

## 1.5 ALMACENAR REGISTRO A TABLA

Descripción: Guardar la información ya validada, en cuanto a su estructura, de un Archivo en la Entidad destino definida

1. Obtener de LISTA DE ARCHIVOS Tipo de Archivo, Nombre del Archivo, Fecha de Aplicación,  
Fecha, Hora, Tipo de Fuente con  
Estatus = Validación Estructura sin error
2. Obtener de TIPOS ARCHIVOS Origen de la Fecha con  
Tipo de Archivo y  
Fecha de Aplicación
3. En caso de que Origen de la Fecha = Fecha de Encabezado ó Fecha de Sumario ó Periodo Información Encabezado ó Periodo Información Sumario  
Seleccionar de VALIDACIONES POR CONCEPTO y DEFINICIONES-REGISTROS  
Tipo de Registro, Posición Inicial, Longitud con  
VALIDACIONES POR CONCEPTO.Tipo de Registro =  
DEFINICIONES-REGISTROS. Tipo de Registro y  
VALIDACIONES POR CONCEPTO.Concepto de Registro =  
DEFINICIONES-REGISTROS.Concepto de Registro y  
Tipo de Archivo y  
Fecha de Aplicación y  
Validación = Fecha de Encabezado ó Fecha de Sumario ó  
Periodo Información Encabezado ó Periodo Información Sumario  
Calcular Longitud del Registro = Longitud de Tipo de Registro  
Obtener de REGISTRO TEMPORAL Consecutivo, Registro con  
Primeros Longitud del Registro- Caracteres de Registro =  
Tipo de Registro  
Fijar Dato a Extraer de Registro los Longitud caracteres a partir de Posición Inicial  
Si longitud de Dato = 8  
Fijar Fecha Inicio a convertir Dato a fecha  
Fijar Fecha Fin a Fecha Inicio  
Si no  
Fijar Fecha Inicio a convertir los primeros 8 caracteres de Dato a fecha  
Fijar Fecha Fin a convertir los últimos 8 caracteres de Dato a fecha
4. En caso de que Origen de la Fecha = Fecha de Archivo  
Fijar Fecha Inicio a Convertir Nombre del Archivo a fecha  
Fijar a Fecha Fin a Fecha Inicio
5. Actualizar de LISTA DE ARCHIVOS Estatus = Subiendo a Base de Datos con  
Tipo de Archivo y  
Nombre del Archivo y  
Fecha y  
Hora y  
Estatus = Validación Estructura sin error
6. Obtener de PROCESO Módulo  
con Proceso
7. En Caso de que Módulo = Contable
  - 7.1. En Caso de que Tipo de Fuente = Afore  
Obtener de REGISTROS TEMPORALES Clave de la Afore con  
Tipo Registro = Encabezado
  - 7.2. En Caso de que Tipo de Fuente = Siefore  
Obtener de REGISTROS TEMPORALES Clave de la Siefore con  
Tipo Registro = Encabezado

**ESTA TESIS NO SALE  
DE LA BIBLIOTECA**

- 7.3. Leer de REGISTROS TEMPORALES Consecutivo, Registro con  
Tipo Registro <> Encabezado  
Armar Inserción a Balanza Diaria  
Leer de DEFINICIONES-REGISTROS Concepto, Tipo de Dato, Longitud,  
Posición, Campo Destino con
- Tipo de Archivo y
  - Fecha de Aplicación y
  - Tipo de Registro
  - En caso de que Concepto = Cuenta
    - Extraer Cuenta = Extraer los Longitud-Caracteres
    - A partir de Posición Inicio
  - En caso de que Concepto = SubCuenta
    - Extraer SubCuenta = Extraer los Longitud-Caracteres
    - A partir de Posición Inicio
  - En caso de que Concepto = Saldo Inicial
    - Extraer Saldo Inicial = Extraer los Longitud-Caracteres
    - A partir de Posición Inicio
  - En caso de que Concepto = Cargos
    - Extraer Cargos = Extraer los Longitud-Caracteres
    - A partir de Posición Inicio
  - En caso de que Concepto = Abonos
    - Extraer Abonos = Extraer los Longitud-Caracteres a
    - partir de Posición Inicio
  - En caso de que Concepto = Saldo Final
    - Extraer Saldo Final = Extraer los Longitud-Caracteres
    - A partir de Posición Inicio
    - Obtener de Cuenta Fecha Inicial con Cuenta
    - Obtener de SubCuenta Fecha Inicial de Subcuenta
    - con Cuenta y SubCuenta
    - Armar Valores de Inserción
    - Ejecutar la Inserción de Valores
8. En otro caso
- Fijar Clave de Dato a 0
  - Leer de REGISTROS TEMPORALES Consecutivo, Registro
  - Ordenar por Registro
  - Para cada uno hacer:
    - Acumular Clave de Dato + 1
    - Fijar Tipo Registro Temp a
      - Primeros Longitud del Registro- Caracteres de Registro
    - Si Tipo Registro Temp <> Tipo Registro Anterior
      - Si QueryInsert <> Valor Nulo y QueryValues <> Valor Nulo
      - Ejecutar QueryInsert + QueryValues
    - Fijar Tipo Registro Anterior a Tipo Registro Temp
    - Obtener de TIPOS REGISTROS Tabla Fija, Tabla Destino con
      - Tipo de Archivo y
      - Fecha de Aplicación y
      - Tipo de Registro = Tipo Registro Temp
      - En Caso de Tabla Fija = 0
      - Buscar en Tabla Destino Dato con
        - Tipo de Archivo y
        - Fecha de Aplicación y
        - Fecha Inicio de Información = Fecha Inicio y
        - Fecha Fin de Información = Fecha Fin



- Si Existe
  - Borrar de Tabla Destino con
    - Tipo de Archivo y
    - Fecha de Aplicación y
    - Fecha Inicio de Información =
    - Fecha Inicio y
    - Fecha Fin de Información = Fecha Fin
  - Leer de DEFINICIONES-REGISTROS Concepto, Tipo de Dato, Longitud, Posición Inicio, Campo Destino con
    - Tipo de Archivo y
    - Fecha de Aplicación y
    - Tipo de Registro
  - En Caso de Tabla Fija = 0
    - Fijar Dato a Extraer los Longitud-Caracteres a partir de Posición Inicio
  - En Caso de que Tipo de Dato = Numérico
    - DatoString = NULL
    - DatoNumerico = Dato
  - En Caso de que Tipo de Dato = Alfanumérico
    - DatoString = NULL
    - DatoNumerico = Dato
  - Insertar en Tabla Destino Tipo de Archivo, Fecha de Aplicación, Fecha Inicio de Información, Fecha Fin de Información, Clave de Dato, DatoString, DatoNumerico
  - Si la Inserción no fue exitosa
    - Actualizar de LISTA DE ARCHIVOS
      - Estatus = Carga con Error con
      - Tipo de Archivo y
      - Nombre del Archivo y
      - Fecha y
      - Hora y
      - Directorio
  - En Caso de Tabla Fija = 1
    - Fijar Dato a Extraer los Longitud-Caracteres a partir de Posición Inicio
    - Acumular QueryInsert + Campo Destino
    - Acumular QueryValues + Dato
- 9. Si QueryInsert <> Valor Nulo y QueryValues <> Valor Nulo
  - Ejecutar QueryInsert + QueryValues
  - Si la ejecución no fue exitosa
    - Actualizar de LISTA DE ARCHIVOS
      - Estatus = Carga con Error con
      - Tipo de Archivo y
      - Nombre del Archivo y
      - Fecha y
      - Hora y
      - Directorio
- 10. Obtener de LISTA DE ARCHIVOS Estatus con
  - Tipo de Archivo y
  - Nombre del Archivo y
  - Fecha y
  - Hora y
  - Directorio

11. Si Estatus = Carga con Error

Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Presento errores en la Carga de Información"

Sino

Actualizar de LISTA DE ARCHIVOS Estatus = Carga Exitosa con

Tipo de Archivo y  
Nombre del Archivo y  
Fecha y  
Hora y  
Directorio

Guardar en BITACORA DE CARGA un mensaje de "El Archivo: **Nombre del Archivo, Fecha y Hora**, Se integro a la Base de Datos Exitosamente"

**NOTAS:**

1. Archivo Almacenado en Temporal	= 3
2. Validación Estructura sin error	= 6
3. Subiendo a Base de Datos	= 7
4. Tipo Registro	= 5
5. Fecha de Archivo	= 0
6. Fecha de Encabezado	= 1
7. Fecha de Sumario	= 2
8. Periodo de Información Encabezado	= 3
9. Periodo de Información Sumario	= 4
10. Fecha de Archivo	= 3
11. Numérico	= 1
12. Alfanumérico	= 2
13. Carga Exitosa	= 8
14. Carga con Error	= 9

6to. Paso **DIAGRAMA ENTIDAD RELACION (ER)**

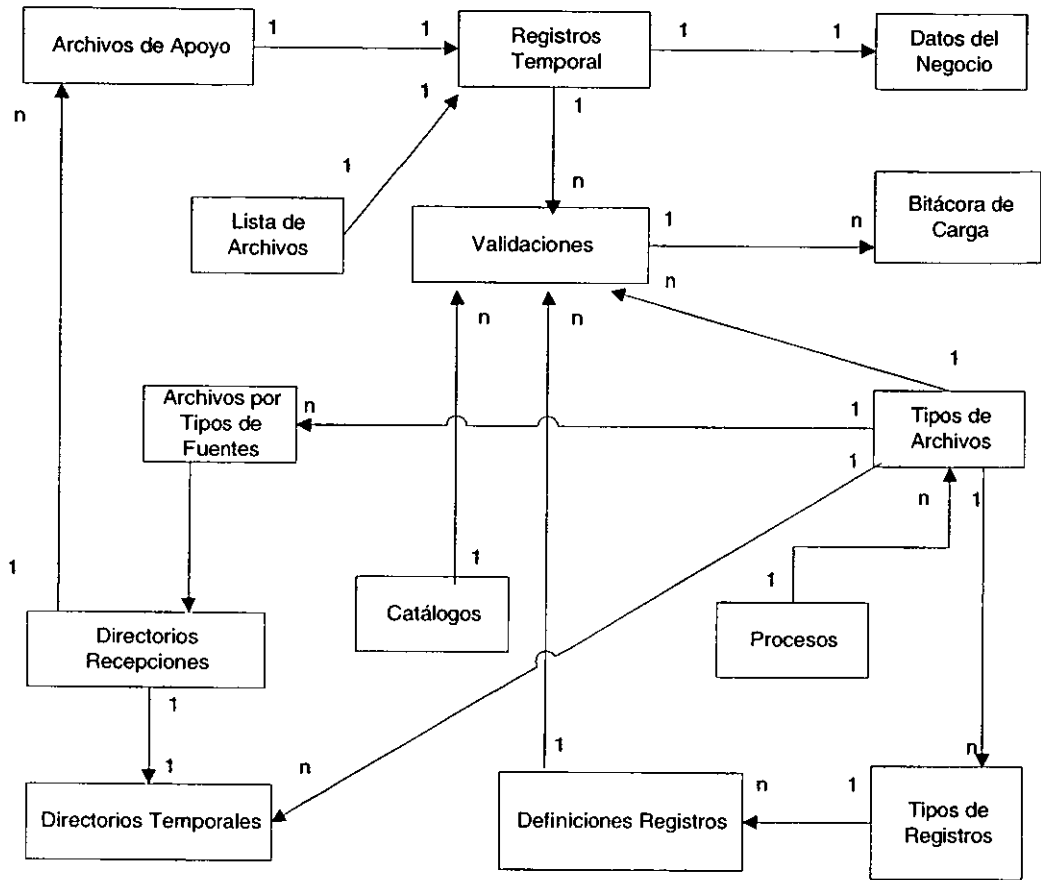
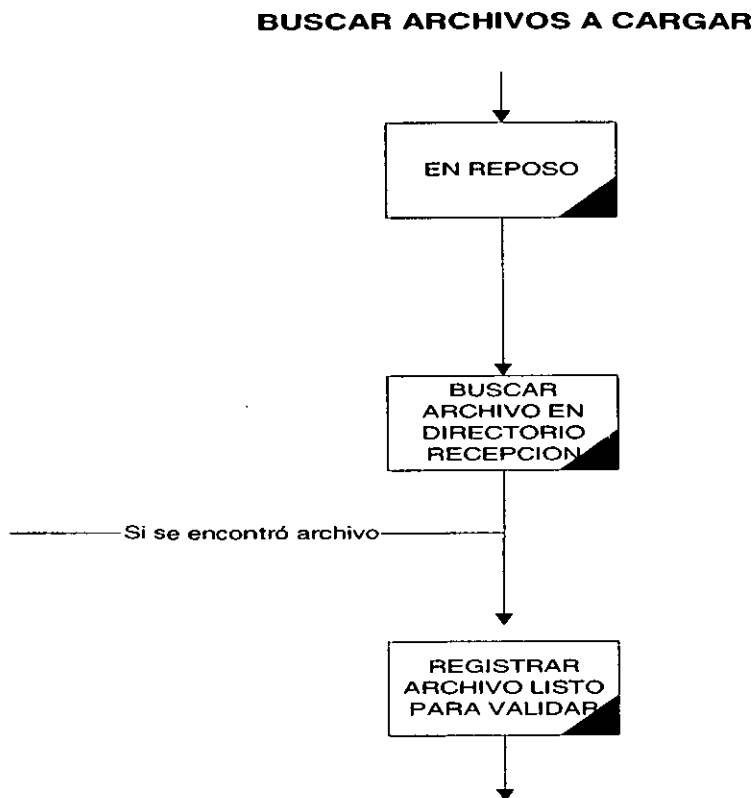


Fig. 3.6<sup>119</sup> Diagrama Entidad- Relación del caso práctico

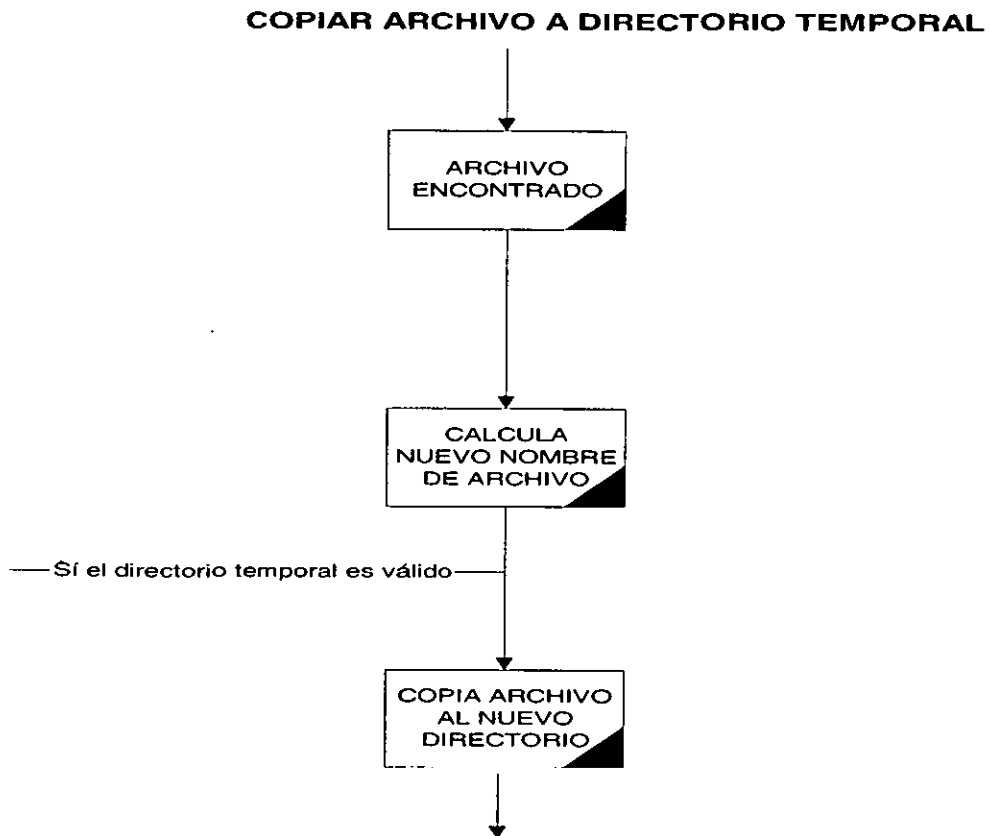
<sup>119</sup> Fuente: Creación propia

**7to. Paso      DIAGRAMA DE TRANSICIÓN DE ESTADOS (DTE)**



**Fig. 3.7<sup>120</sup>** Diagrama de Transición de Estados de Buscar Archivos a Cargar

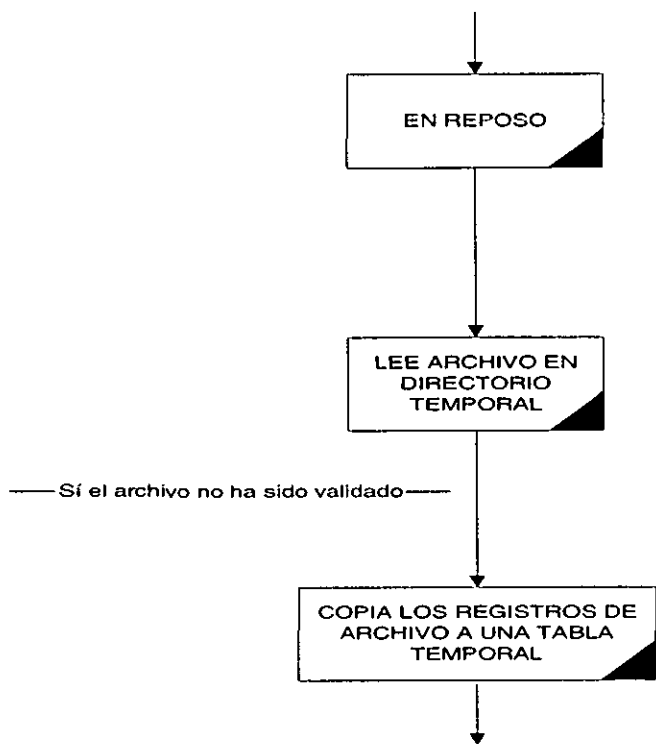
<sup>120</sup> Idem



**Fig. 3.8**<sup>121</sup> Diagrama de Transición de Estados de Copiar Archivo a Directorio Temporal

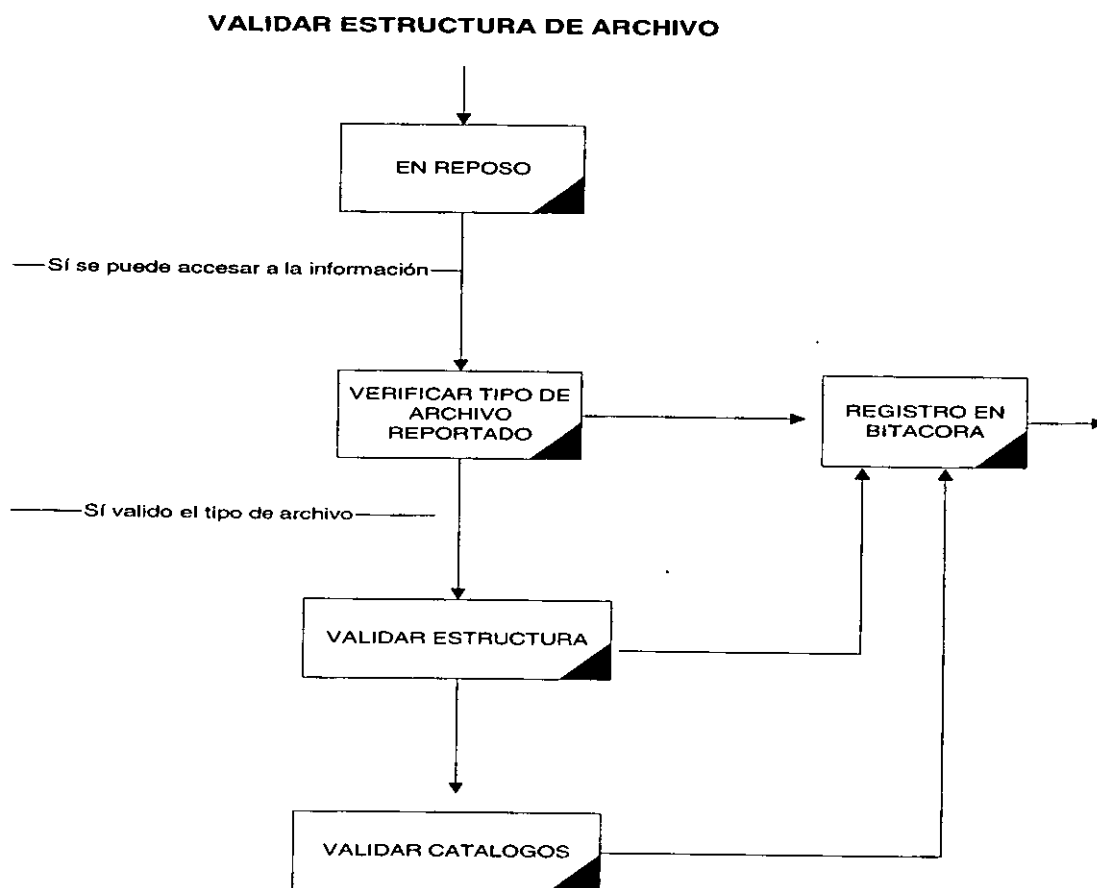
<sup>121</sup> Idem

**ALMACENAR REGISTROS DE ARCHIVO EN TEMPORAL**



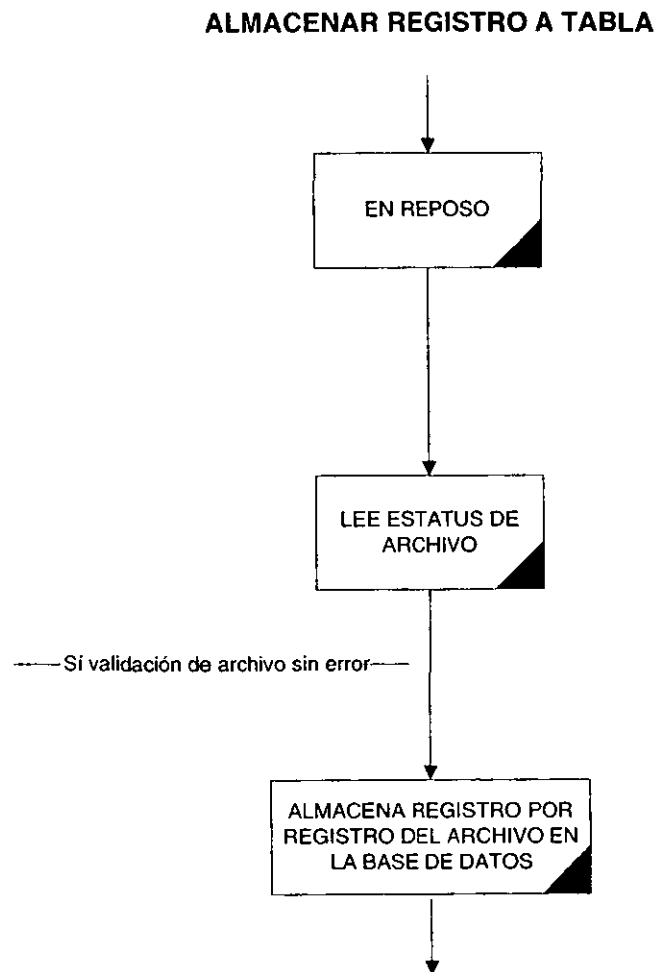
**Fig. 3.9**<sup>122</sup> Diagrama de Transición de Estados de Almacenar Registros en Archivo Temporal

<sup>122</sup> Idem



**Fig. 3.10<sup>123</sup>** Diagrama de Transición de Estados de Validar Estructura de Archivo

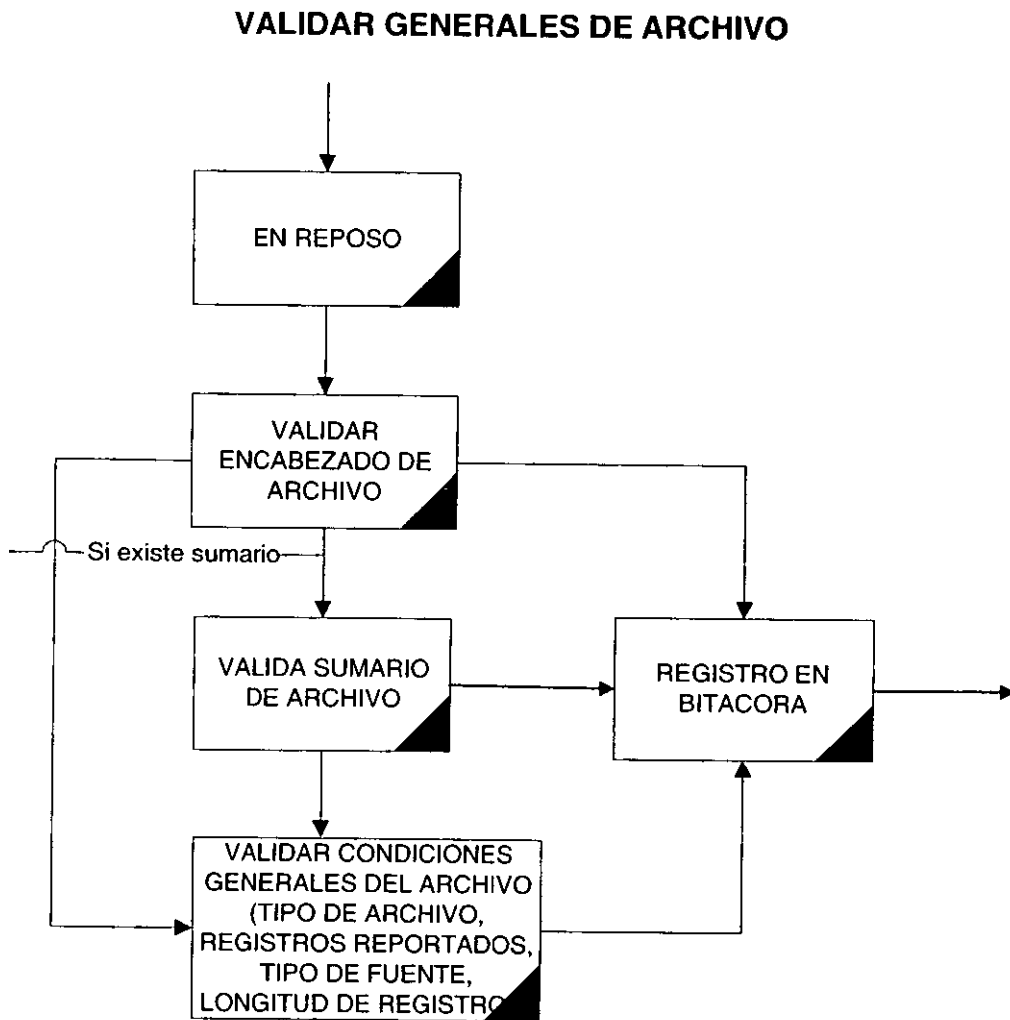
<sup>123</sup> Idem



**Fig. 3.11**<sup>124</sup> Diagrama de Transición de Estados de Almacenar Registro a Tabla

<sup>124</sup> Idem

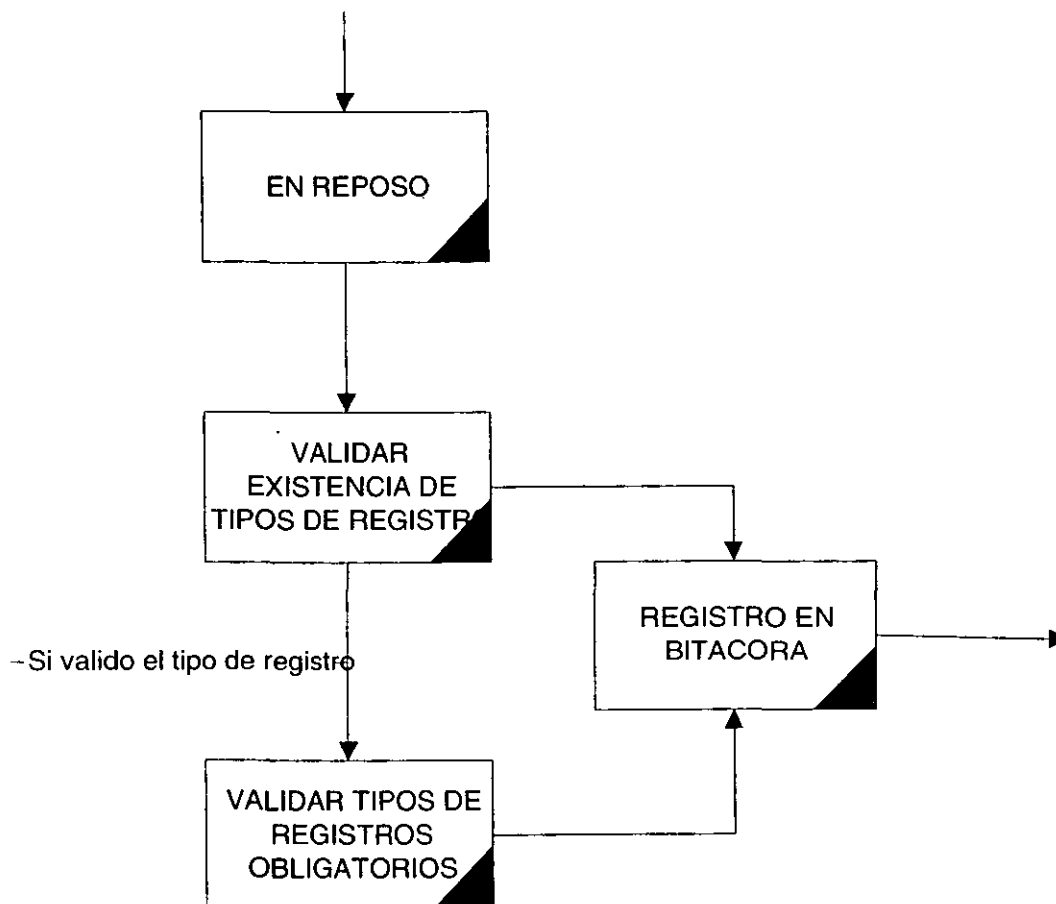




**Fig. 3.12<sup>125</sup>** Diagrama de Transición de Estados de Validar Generales de Archivo

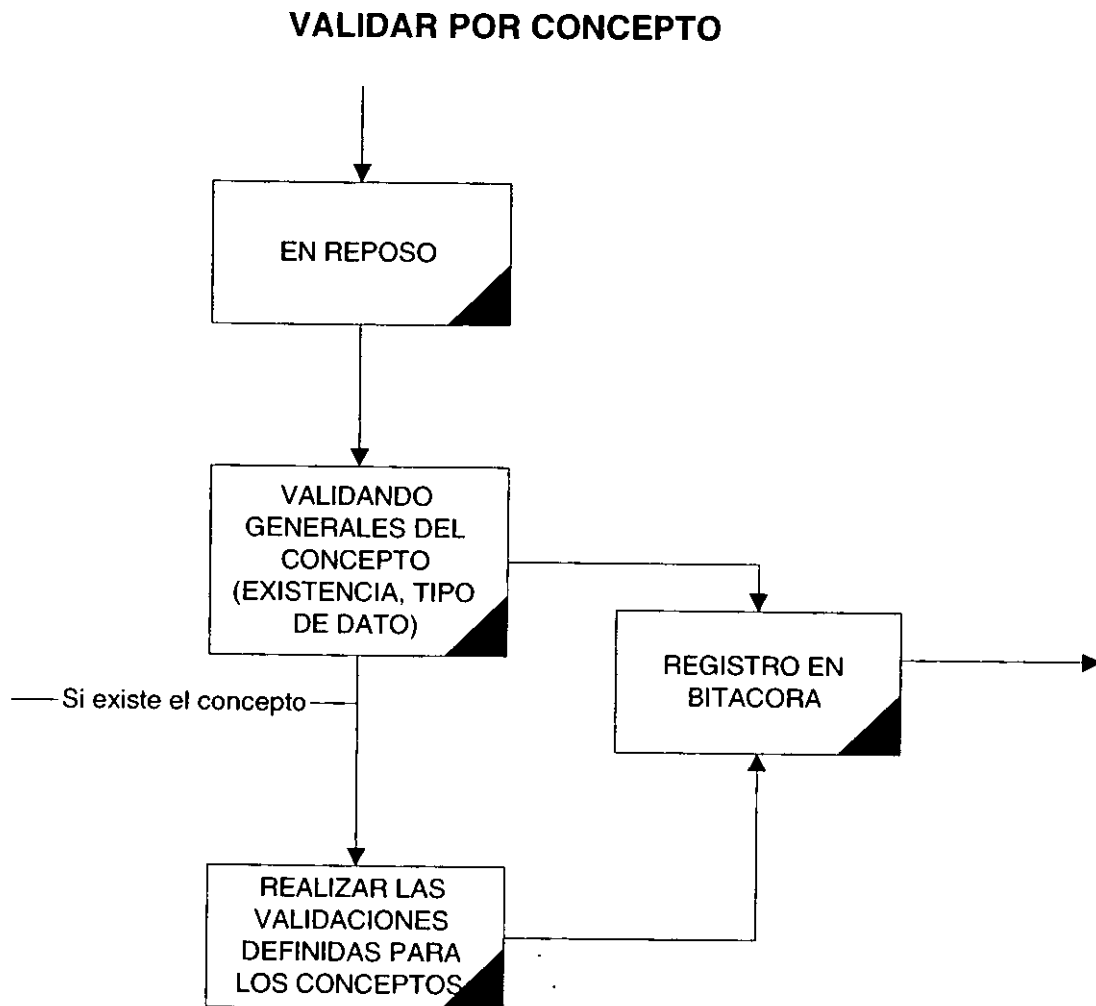
<sup>125</sup> Idem

### VALIDAR POR TIPO DE REGISTRO



**Fig. 3.13**<sup>126</sup> Diagrama de Transición de Estados de Validar por Tipo de Registro

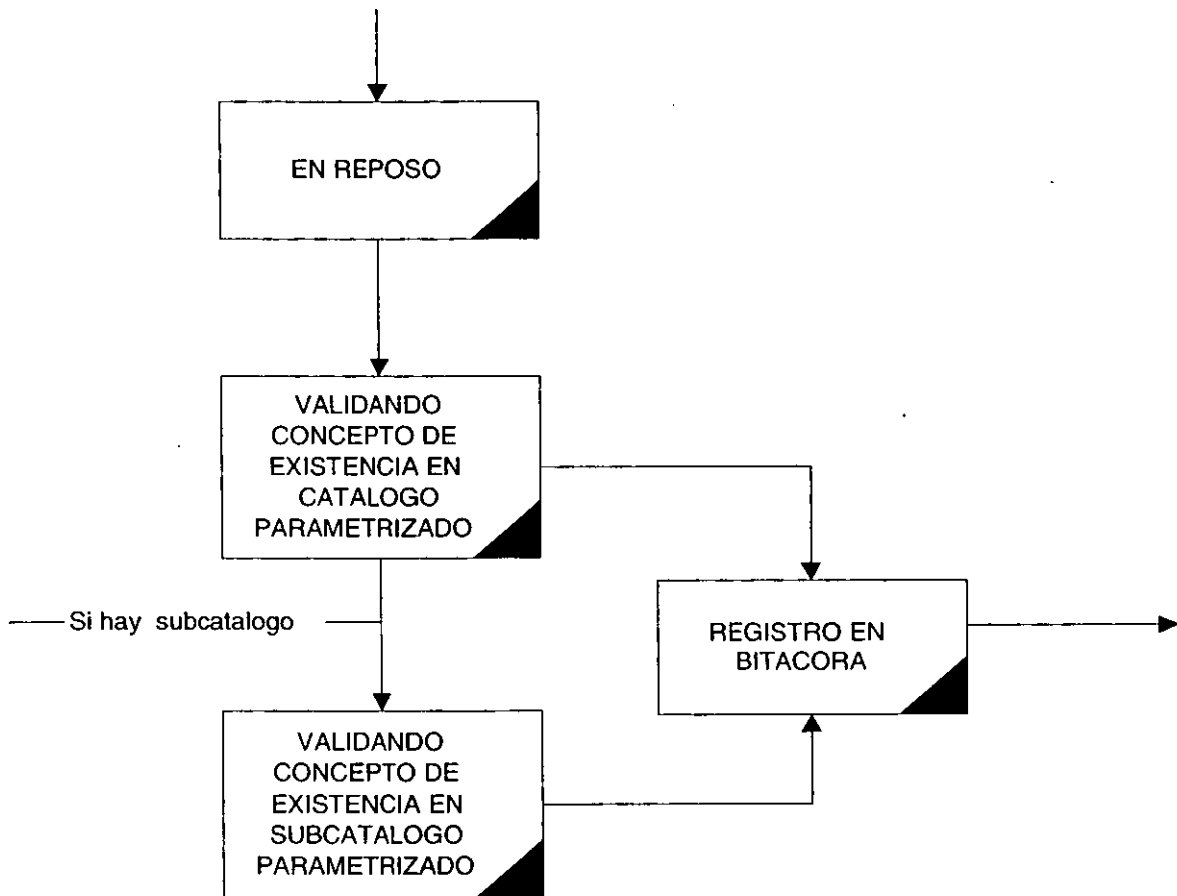
<sup>126</sup> Idem



**Fig. 3.14**<sup>127</sup> Diagrama de Transición de Estados de Validar por Concepto

<sup>127</sup> Idem

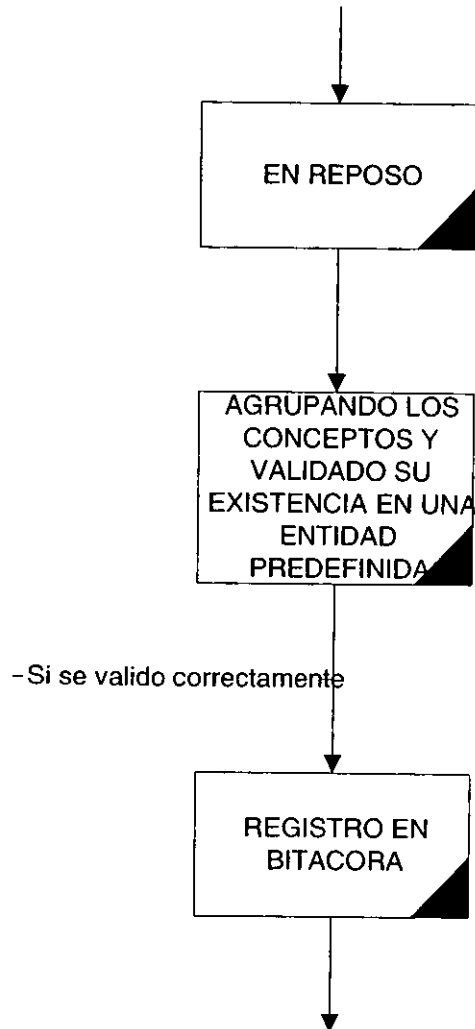
### VALIDAR CATALOGO PARAMETRIZADO



**Fig. 3.15<sup>128</sup>** Diagrama de Transición de Estados de Validar Catálogo Parametrizado

<sup>128</sup> Idem

### VALIDAR CATALOGO FIJO



**Fig. 3.16**<sup>129</sup> Diagrama de Transición de Estados de Validar Catálogo Fijo

<sup>129</sup> Idem

### **3.1.3. Desarrollo de las etapas del análisis orientado a objetos "UML"**

#### **1er. Paso: REQUERIMIENTOS**

El primer paso que se lleva en el desarrollo de una aplicación, se utilice una metodología o no , es el levantamiento de requerimientos resultando en nuestro caso lo siguiente:

*Presentación general.-* Este proyecto tiene como objetivo el de verificar los lineamientos que exige la CONSAR, en cuanto a estructura dinámica de archivos; la existencia de datos en catálogos generales; y el almacenar el resultado e información enviada en algún dispositivo físico, en el caso planteado será en un servidor Windows NT, con manejador SQL-Server 7.

*Los Clientes.-* Cualquier entidad involucrada en el envío de archivos a CONSAR, como lo son: Afores, Siefores, Procesar, etc.

*Las Metas.-* Automatizar el proceso de verificación de los archivos enviados a CONSAR, proporcionando los beneficios de: detectar a tiempo posibles problemas en la recepción de archivos en la comisión evitando posibles penalizaciones monetarias, dando tiempo a su corrección o en su caso al envío de alguna justificación antes de que la CONSAR lo detecte.

*Funciones del Sistema y su categoría.-*

- a) R1.1 El sistema deberá detectar la llegada de un archivo Nuevo, es una función Evidente.
- b) R1.2 El sistema verificará la estructura de un archivo de acuerdo al tipo definido por la CONSAR, es una función evidente.
- c) R1.3 El sistema validará la existencia de información recibida en los catálogos definidos, es una función evidente.
- d) R1.4 El sistema reportará los resultados en un dispositivo de salida, se plantea el uso de bitácora, es una función evidente.
- e) R1.5 El sistema almacenará la información recibida en una base de datos, es una función oculta.
- f) R1.6 El sistema categorizará los tipos de errores que se detecten en la validación, es una función oculta.
- g) R1.7 El sistema determinará el estatus en el que se encuentra el archivo recibido, es una función oculta.

*Atributos del sistema.-*

- a) Estará establecido en un servidor de Windows NT 4, con un manejador de Base de Datos SQL-server 7.0 y programado en Visual Basic 6.0.
- b) El tiempo de validación en un archivo dependerá del número de registro y el número de catálogos que se tengan que validar por cada tipo de registro; así como la velocidad del procesador del servidor y el espacio disponible en este último.
- c) El estado en el que se encuentre un archivo validado se consultará indicando a detalle las fallas posibles.

- d) Se llevará un registro de acciones realizadas durante la validación del archivo.
- e) Permitirá la definición dinámica de la estructura de un tipo de archivo, sus características generales que son: identificador de tipo de archivo, nombre del archivo, descripción, el tipo de entidad que lo reportará, el directorio de donde se tomará el archivo, longitud del registro, definir los tipos de registros que se podrán reportar en el archivo y de cada uno de los tipos de registros se podrán especificar los campos, sus tipos de datos, localizaciones, longitudes, y la indicación de que se validen contra algún catálogo o a un valor determinado por el usuario.

**2do. Paso: CASOS DE USO**

Concepto	Descripción	
<b>Nombre:</b>	Definir validaciones especiales de un campo de un archivo.	
<b>Actores:</b>	EL OPERADOR del Cliente (Afore, Consar, Siefore, Procesar, etc.).	
<b>Propósito:</b>	Definir las validaciones especiales del archivo, los campos obligatorios, valores constantes en los campos, sus mascaras <sup>130</sup> .	
<b>Resumen:</b>	El OPERADOR definirá si el valor recibido en cada campo tendrá un valor constante, si es obligatorio que se reporte, si debe llegar en un formato específico.	
<b>Tipo:</b>	Primario.	
<b>Referencias Cruzadas:</b>	Definir la estructura de un archivo	
<b>Curso normal de los eventos:</b>	<p style="text-align: center;"><b>ACTOR</b></p> <p>1. El OPERADOR agrega cada una de las validaciones especiales, asociadas a cada uno de los campos, capturando el tipo de validación, los formatos y los valores constantes.</p>	<p style="text-align: center;"><b>ACCIONES</b></p> <p>2. El sistema verifica la correcta captura de información.</p> <p>3. El sistema almacena la información en la base de datos.</p>
<b>Curso alternativo de los eventos:</b>	Ninguno.	

**Tabla 3.2<sup>131</sup>** Caso de Uso Definir validaciones especiales de un campo de un archivo.

<sup>130</sup> El significado de mascara es asociado a la presentación física de un campo, por ejemplo si deseamos validar que un campo tenga una mascara de tipo rfc-fisico "XXXX999999-XX" entonces se forzará que el valor del campo sean los cuatro primeros caracteres de tipo alfanumérico (letras o números), los seis siguientes de tipo numérico, un guión y por último tres caracteres alfanuméricos.

Concepto	Descripción	
<b>Nombre:</b>	Definir la estructura de un Archivo.	
<b>Actores:</b>	OPERADOR del Cliente (Afore, Consar, Siefore, Procesar, etc.).	
<b>Propósito:</b>	Definir las características generales del archivo, la estructura y validaciones de su contenido.	
<b>Resumen:</b>	El OPERADOR definirá el identificador, el tipo de origen que lo emitirá, la descripción, la longitud de los registros, los tipos de registros, los campos de los registros, la longitud de los campos, el tipo de dato, y las validaciones especiales y generales <sup>132</sup> de un tipo de archivo. Almacenando esta información en una base de datos.	
<b>Tipo:</b>	Primario.	
<b>Referencias</b>	Definir validaciones especiales de un campo de un archivo.	
<b>Cruzadas:</b>	Definir las validaciones de Existencia de un campo en Catálogos.	
<b>Curso normal de los eventos:</b>	<p style="text-align: center;"><b>ACTOR</b></p> <ol style="list-style-type: none"> <li>1. El OPERADOR define los datos generales del archivo.</li> <li>4. El OPERADOR define los tipos de registros que contendrá el archivo.</li> <li>7. El OPERADOR define los campos y las características de cada uno de ellos.</li> </ol>	<p style="text-align: center;"><b>ACCIONES</b></p> <ol style="list-style-type: none"> <li>2. El sistema verifica la correcta captura de información.</li> <li>3. El sistema almacena la información en la base de datos.</li> <li>5. El sistema verifica la correcta captura de información.</li> <li>6. El sistema almacena la información en la base de datos.</li> <li>8. El sistema verifica la correcta captura de información.</li> <li>9. El sistema almacena la información en la base de datos.</li> </ol>
<b>Curso alternativo de los eventos:</b>	Ninguno.	

**Tabla 3.3<sup>133</sup>** Caso de Uso de Definir la estructura de un Archivo.

<sup>131</sup> Fuente: Creación propia

<sup>132</sup> Entiéndase que las validaciones especiales se refieren a las propias del campo, mientras que en las generales se refieren a las validaciones de tipo "que el valor enviado exista en algún catálogo"

<sup>133</sup> Fuente: Creación propia



<b>Concepto</b>	<b>Descripción</b>	
<b>Nombre:</b>	Definir las validaciones de Existencia de un campo en Catálogos.	
<b>Actores:</b>	EL OPERADOR del Cliente (Afore, Consar, Siefore, Procesar, etc.).	
<b>Propósito:</b>	Asignar al o los campos que se reportaran en un archivo, el catálogo en el que se desea validar la existencia de sus valores.	
<b>Resumen:</b>	El OPERADOR asignará a algún campo o grupo de campos reportados en el archivo que se esta definiendo, su(s) correspondientes campos en un catálogo(s) existente en el sistema.	
<b>Tipo:</b>	Primario.	
<b>Referencias Cruzadas:</b>	Definir la estructura de un archivo.	
<b>Curso normal de los eventos:</b>	<p style="text-align: center;"><b>ACTOR</b></p> <ol style="list-style-type: none"> <li>1. El OPERADOR asigna contra que catálogo y que campo deberá validarse el dato que se reciba.</li> <li>4. El OPERADOR agrupará los campos del archivo contra los campos que validarán su existencia en un catálogo.</li> </ol>	<p style="text-align: center;"><b>ACCIONES</b></p> <ol style="list-style-type: none"> <li>2. El sistema verifica la correcta captura de información.</li> <li>3. El sistema almacena la información en la base de datos.</li> <li>5. El sistema verifica la correcta captura de información.</li> <li>6. El sistema almacena la información en la base de datos.</li> </ol>
<b>Curso alterno de los eventos:</b>	Ninguno	

**Tabla 3.4**<sup>134</sup> Caso de Uso de Definir las validaciones de Existencia de un campo en Catálogos

<sup>134</sup> Idem

<b>Concepto</b>	<b>Descripción</b>	
<b>Nombre:</b>	Detectar llegada de Archivo.	
<b>Actores:</b>	OPERADOR del Cliente (Afore, Consar, Siefore, Procesar, etc.)	
<b>Propósito:</b>	Identificar cuando un Archivo es depositado en algún directorio definido para recepción de archivos, iniciando su validación.	
<b>Resumen:</b>	Para cada uno de los directorios definidos para recepción de archivo buscar que archivos han llegado. Para cada uno de los detectados se agrega un registro en la bitácora de acciones asignándole un estatus de "encontrado".	
<b>Tipo:</b>	Primario.	
<b>Referencias</b>	Definir la estructura de un archivo	
<b>Cruzadas:</b>	Definir las validaciones especiales de un campo de un archivo Definir las validaciones de Existencia de información en Catálogos Validar Archivo. Consultar resultado.	
<b>Curso normal de los eventos:</b>	<p style="text-align: center;"><b>ACTOR</b></p> <p>1. El OPERADOR coloca el archivo en el directorio que se espera encontrar</p>	<p style="text-align: center;"><b>ACCIONES</b></p> <p>2. El sistema busca archivos en los directorios definidos. 3. El sistema encuentra el archivo y hace un registro en la bitácora de acciones indicando el tipo de archivo que reportan, el tipo de fuente que lo reporta<sup>135</sup>, la fecha del archivo, la fecha en que fue encontrado, y el estatus de archivo "encontrado". 4. El sistema valida la estructura del archivo haciendo el registro de cada acción en la bitácora.</p>
<b>Curso alternativo de los eventos:</b>	En el caso de que el OPERADOR no tenga acceso al servidor, se puede recurrir a la carga automática de archivos.	

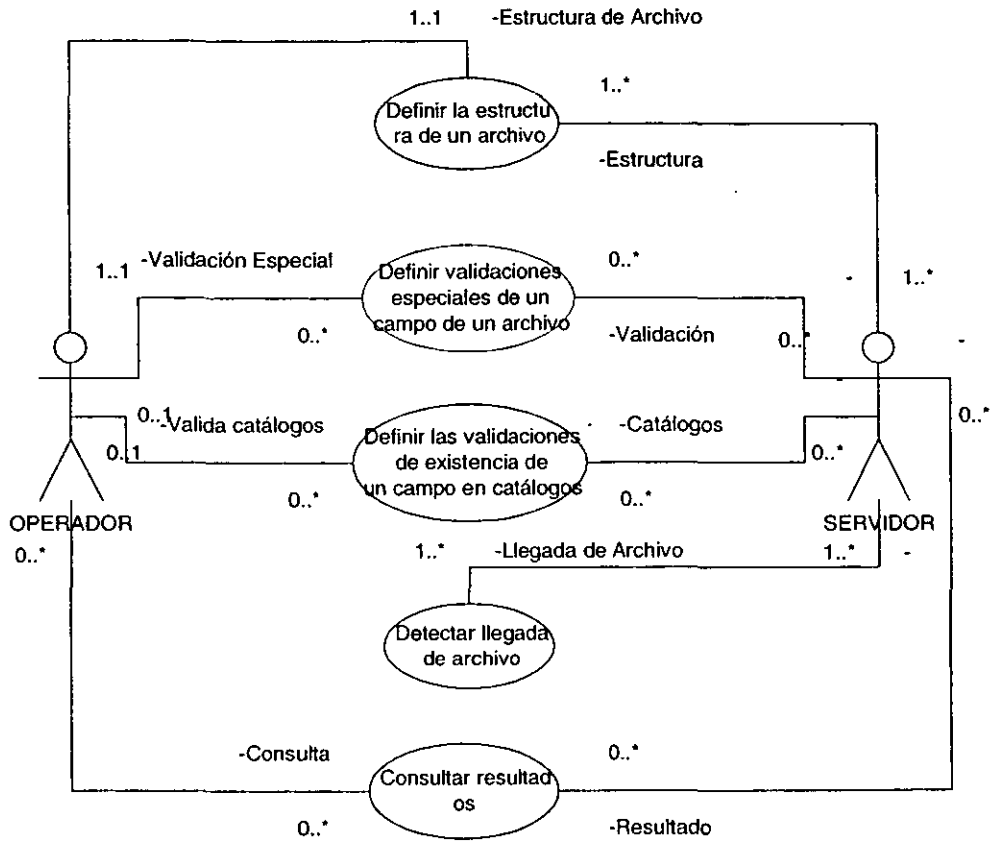
**Tabla 3.5** Caso de Uso de Detectar Llegada de Archivo

<sup>135</sup> Será el origen del archivo, qué entidad lo deposito, por ejemplo la Afore, Consar, Procesar, etc.

<b>Concepto</b>	<b>Descripción</b>	
<b>Nombre:</b>	Consultar resultado.	
<b>Actores:</b>	OPERADOR del Cliente (Afore, Consar, Siefore, Procesar, etc.)	
<b>Propósito:</b>	Consultar el resultado de la validación de un archivo	
<b>Resumen:</b>	El OPERADOR consultará la información que se tuvo como resultado de la validación de un archivo.	
<b>Tipo:</b>	Primario.	
<b>Referencias</b>	Validar Archivo.	
<b>Cruzadas:</b>		
<b>Curso normal de los eventos:</b>	<p style="text-align: center;"><b>ACTOR</b></p> <p>1. El OPERADOR solicitará el estatus y resultado de la validación, proporcionando los filtros por el cual desea hacer una consulta como son tipo, fecha, fecha en que se recibió, origen, y estatus del archivo.</p>	<p style="text-align: center;"><b>ACCIONES</b></p> <p>2. El sistema en base a los parámetros proporcionados por el OPERADOR, buscará la información solicitada en la base de datos.</p> <p>3. El sistema desplegará los resultados obtenidos de la validación del archivo, desplegando el detalle de la bitácora.</p>
<b>Curso alternativo de los eventos:</b>	Ninguno.	

**Tabla 3.6**<sup>136</sup> Caso de uso de Consultar resultado.

**DIAGRAMAS DE CASOS DE USO**



**Fig. 3.17<sup>137</sup>** Diagrama de Casos de uso del caso práctico

**3er. PASO: MODELO CONCEPTUAL**

El modelo conceptual esta compuesto por la Lista de conceptos en la tabla 3.7 y el modelo conceptual al caso práctico Fig. 3.18

<sup>137</sup> Idem

<b>Categoría del Concepto</b>	<b>Asociado al problema en estudio</b>
Objetos físicos o tangibles	Directorios de recepción Archivos
Especificaciones, diseño o descripciones de cosas	Especificación de estructuras por tipo de archivo Validaciones especiales por campos de un archivo Catálogos a validar Bitácora de acciones
Lugares	Empresa cliente (Afore, Procesar, CONSAR, Siefore, etc.)
Transacciones	Busca, Valida, Define, Consulta Registro
Línea o renglón de elemento de transacciones	Búsqueda de archivo Validación de archivo Definición de estructura de archivo Definición de validaciones Búsqueda de Valores en un Catálogo Registro de resultados Registro de información Consulta de Resultado
Papel de las personas	OPERADOR
Contenedores de otras cosas	Estructura de Archivos Información de Archivos Catálogos
Cosas dentro de un contenedor	Campos, Datos o información
Organizaciones	Contraloría normativa Vigilancia Consar
Registros de finanzas, de trabajo, de contratos de asuntos legales	Bitácora de Acciones

**Tabla 3.7**<sup>138</sup> Lista de conceptos del caso práctico

<b>Categoría del Concepto</b>	<b>Asociado al problema en estudio</b>
Eventos	Llegada de archivo Archivo encontrado Archivo validado Archivo con error Petición de validación especial Solicitud de validación contra catálogo
Procesos (usualmente no se representan como conceptos)	Valida un Archivo Registra un Acción Almacena información Consulta resultado Registra estructura de archivo Registra validaciones especiales Registra validaciones contra catálogos
Reglas y políticas	Políticas en la definición de archivos Políticas de penalización de archivos erróneos
Catálogos	Tipos de Archivos Tipos de Fuentes (origen) Fuentes Errores Entidades Federativas Empresas Tipo de Operaciones Salario Mínimo Cuentas de Retiro SubCuentas de Retiro Cuentas de Traspasos SubCuentas de Traspasos Guía Contable Afore Guía Contable Siefore
Manuales, libros	Manuales Operativos, Diario Oficial de la Federación Régimen de inversión

**Tabla 3.7 Continuación** de Lista de conceptos del caso práctico

<sup>138</sup> Fuente: Creación propia

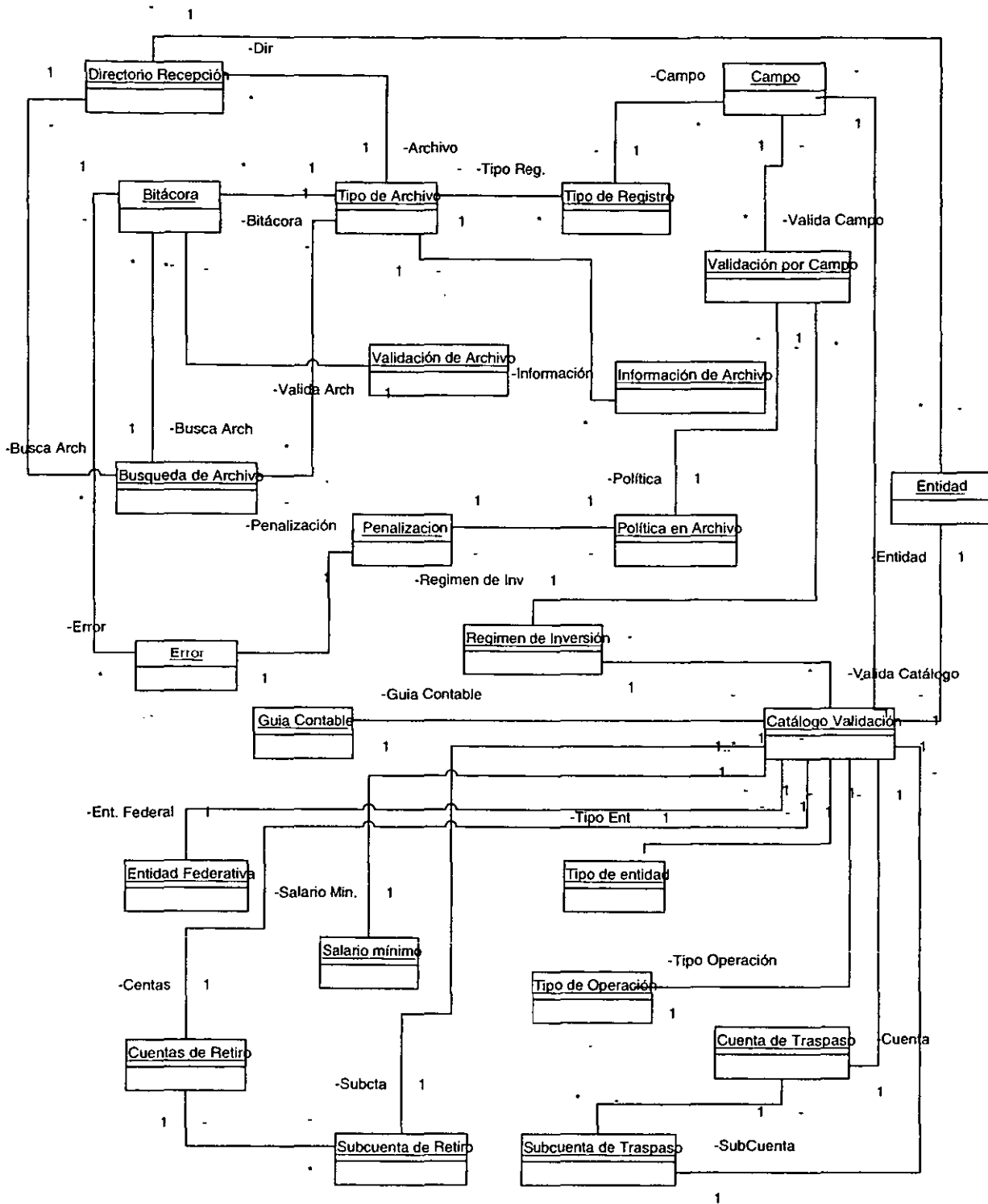


Fig. 3.18<sup>139</sup> Modelo Conceptual al caso práctico

<sup>139</sup> Idem

Directorio Recepción
-Ruta : String
-Entidad Origen : String

Tipo de Archivo
-Tipo de Archivo : String
-Longitud de Registro : int
-Tipo de Operación : int
-Descripción : String

Tipo de Registro
-Identificador de Tipo de registro : String
-Descripción : String
-Cantidad Obligatorio : int

Bitácora
-Fecha Acción : Date
+Tipo de Archivo : String
+Estatus : char
+Error : String

Validación de Archivo
-Identificador de Validación : int
-Descripción : String

Campo
-Identificador de Campo : int
-Descripción : String
-Posición Inicial : int
-Longitud : int
-Número de decimales : int
-Tipo de dato : char

Busqueda de Archivo
+Tipo de Archivo : String
+Directorio : String
-Fecha de Búsqueda : Date

Penalización
-Identificador de Penalización : int
-Descripción : String
-Monto Penalización : long double
-En Salario Mínimo : bool
-Número de Salarios : short

Validación por Campo
-Identificador de Validación : int
-Descripción : String
-Operador : char
-Valor Cosntante : String

Error
-Tipo de Error : int
-Descripción : String
-Estatus de Error : char

Información Archivo
-Tipo de Archivo : String
-Fecha del Archivo : Date
-Fecha de Información : Date
-Dato Numérico : long double
-Dato string : String
+Campo Informado : int

Regimen de Inversión
-Identificador de Clausula : int
-Descripción : String

Guia Contable
-Fecha Vigencia : Date
-Tipo de Entidad : String
-Cuenta Contable : int
-SubCuenta : int
-Cuenta de Cargo : bool
-Descripción : String

Política de Archivo
-Identificador de Política : int
-Descripción : String

Catalogo Validación
-Identificador de Validación Catálogo : int
-Descripción : String
-Encontrar : bool

Entidad Federativa
-Identificador de Entidad Federal : int
-Descripción : String
-Extranjera : bool

Salario Mínimo
-Fecha : Date
-Monto de Salario : double

Tipo de Entidad
-Identificador de tipo de entidad : String
-Descripción : String

Cuentas de Retiro
-Cuenta Retiro : String
-Descripción : String

Cuenta de Traspaso
-Cuenta de Traspaso : String
-Descripción : String

Entidad
-Identificador de tipo de entidad : String
-Identificador de Entidad : String
-Descripción : String

Subcuenta de Retiro
-Cuenta de Retiro : String
-Subcuenta : String
-Descripción : String

Subcuenta de traspaso
-Cuenta de Traspaso : String
-Subcuenta : String
-Descripción : String

Tipo de Operación
-Identificador de Tipo de Operación : int
-Descripción : String



**4to. PASO: EL GLOSARIO**

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Definir la Estructura de un Archivo	Caso de uso	Descripción de como se define la estructura de un archivo
Definir las validaciones especiales	Caso de uso	Descripción de como se asignan las validaciones especiales a los campos definidos en un archivo
Definir validaciones de existencia del valor de un campo en catálogos	Caso de uso	Descripción de como se asignan las validaciones contra un catálogo
Detectar Llegada de Archivo	Caso de uso	Descripción de como encuentra un nuevo archivo a validar
Consultar resultados	Caso de uso	Descripción de como obtener los resultado de la validación de un archivo
Directorio Recepción	Concepto	Es donde se tiene el lugar donde el sistema hará la búsqueda de archivos
Directorio Recepción.Ruta:string	Atributo	Es donde se define la dirección o ruta del servidor donde se depositan los archivos a validar
Directorio Recepción.Entidad Origen:string	Atributo	Es donde se tiene la indicación de que tipo de entidad lo envía, es el origen de la información
Tipo de Archivo	Concepto	Es donde se tiene las características de los archivos
Tipo de Archivo.Tipo de Archivo: string	Atributo	Tiene la identificación del tipo de archivo
Tipo de Archivo.Longitud de registro: int	Atributo	Tiene la definición de cuantos caracteres se deben reportar en cada registro o renglón contenido en el archivo

**Tabla 3.8<sup>140</sup>** Glosario resultante del caso práctico

<sup>140</sup> Fuente: Creación Propia

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Tipo de Archivo.Tipo de Operación:int	Atributo	Tiene la definición del tipo de operación que reportará el archivo
Tipo de Archivo.Descripción: string	Atributo	Tiene la descripción y definición del archivo
Tipo de Registro	Concepto	Es donde se tiene la definición de los tipos de registros que podrá contener un archivo determinado
Tipo de Registro.Identificador de Tipo de reg.:strig	Atributo	Tiene clave o identificador de un tipo de registro
Tipo de Registro.Descripción: string	Atributo	Tiene la descripción del tipo de registro
Tipo de Registro.Cantidad Obligatorio: int	Atributo	Tiene el número de veces que un tipo de registro deberá reportarse en un archivo
Bitácora	Concepto	Es donde se tiene el registro de todas las acciones, secuencia de como se llevo a cabo el proceso de validación
Bitácora.Fecha Acción: Date	Atributo	Tiene la fecha de cuando sucedió una acción
Bitácora.Tipo de Archivo: string	Atributo	Tiene el tipo de archivo que tuvo el suceso
Bitácora.Estatus: char	Atributo	Tiene el estatus en el que se encuentra el archivo validado
Bitácora.Error: string	Atributo	Tiene la descripción del evento
Validación de Archivo	Concepto	Es donde se tiene la validaciones que se harán a un archivo a validar
Validación de Archivo.Identificador de validación:int	Atributo	Tiene la clave que identifica a una validación

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Validación de Archivo.Descripción: string	Atributo	Tiene la descripción de una validación
Campo	Concepto	Es donde se tiene la configuración de los campos que tiene un registro de un archivo
Campo.Identificador de campo: int	Atributo	Tiene el identificador del campo
Campo.Descripción:string	Atributo	Tiene la descripción o nombre del campo
Campo.Posición Inicial: int	Atributo	Tiene el numero de la posición inicial dentro de un registro, en la que se debe reportar el campo
Campo.Longitud: int	Atributo	Tiene la cantidad de caracteres que tiene el campo
Campo.Número de decimales: int	Atributo	Tiene cantidad de decimales que tiene el campo
Campo.Tipo de dato: char	Atributo	Tiene el tipo de dato que es el campo
Busqueda de Archivo	Concepto	Describe como se realiza la búsqueda de un archivo
Búsqueda de Archivo.Tipo de Archivo: string	Atributo	Tiene el tipo de archivo que se busca
Búsqueda de Archivo.Directorio: string	Atributo	Tiene el directorio en donde se busca el archivo
Búsqueda de Archivo.Fecha de Busqueda:date	Atributo	Tiene la fecha en que se hace la búsqueda
Penalización	Concepto	Es donde se define la penalizaciones que se tienen en caso de no tener una validación exitosa

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Penalización. Identificador de penalización: int	Atributo	Tiene la clave de la penalización
Penalización. Descripción: string	Atributo	Tiene la descripción y/o nombre de la penalización
Penalización. Monto Penalización: long double	Atributo	Tiene el monto, en dinero, de la penalización
Penalización. En Salario Mínimo: boolean	Atributo	Indica que la penalización es basada en el salario mínimo
Penalización. Numero de Salarios: short	Atributo	En el caso de que la penalización este basada en el salario mínimo, entonces tiene la cantidad de salarios mínimos
Validación por Campo	Concepto	Es donde se tienen las validaciones de un campo
Validación por Campo. Identificador de validación: int	Atributo	Tiene la clave de la validación
Validación por Campo. Descripción: string	Atributo	Tiene la descripción de la validación
Validación por Campo. Operador: char	Atributo	Tiene el operador que se aplica en la validación
Validación por Campo. Valor Constante: string	Atributo	Tiene el valor que se pueda utilizar en la validación
Error	Concepto	Es donde se tiene los eventos que pueden ocurrir durante el proceso de validación
Error. Tipo de Error: int	Atributo	Tiene la identificación del error
Error. Tipo de Error: string	Atributo	Tiene la descripción del error
Error. Estatus de error: char	Atributo	Tiene el estado que el archivo tendrá si llegase a tener el error

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Régimen de Inversión	Concepto	Es donde se tiene las condiciones de inversión establecidas por la CONSAR
Régimen de Inversión.Identificador de Cláusula: int	Atributo	Tiene la clave de la cláusula
Régimen de Inversión.Descripción: string	Atributo	Tiene la descripción de la cláusula
Información Archivo	Concepto	Es donde se almacena o registra la información reportada en un archivo
Información Archivo.Tipo de archivo: string	Atributo	Tiene el tipo de archivo a quien corresponde la información
Información Archivo.Fecha del archivo: date	Atributo	Tiene la fecha del archivo
Información Archivo.Fecha de información: date	Atributo	Tiene la fecha de cuando es la información reportada
Información Archivo.Dato numérico: long double	Atributo	Tiene la información de tipo numérico que se reportó en el archivo
Información Archivo.Dato string: string	Atributo	Tiene la información de tipo carácter que se reportó en el archivo
Información Archivo.Campo informado: int	Atributo	Tiene el identificador del campo que se esta reportando
Guía Contable	Concepto	Es donde se define la guía contable de la Afore y la Siefore
Guía Contable.Fecha vigencia: date	Atributo	Tiene la fecha a partir de cuándo es vigente la cuenta con subcuenta
Guía Contable.Tipo de Entidad: string	Atributo	Tiene el tipo de entidad al que pertenece la cuenta
Guía Contable.Cuenta contable: int	Atributo	Tiene la cuenta contable
Guía Contable.Subcuenta : int	Atributo	Tiene la subcuenta contable

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Guía Contable.Cuenta de cargo: boolean	Atributo	Indica si la cuenta es de tipo cargo
Guía Contable.Descripción: string	Atributo	Tiene la descripción de la cuenta
Política de Archivo	Concepto	Es donde se tiene las políticas que debe cumplir un archivo
Política de Archivo.Identificador de política: int	Atributo	Tiene la clave con que se identifica a la política
Política de Archivo.Descripción: string	Atributo	Tiene la descripción de la política
Catálogo Validación	Concepto	Se tiene los catálogos que validan a los campos
Catálogo Validación.Identificador de validación catálogo: int	Atributo	Tiene la clave que identifica la validación de un catálogo
Catálogo Validación.Descripción : string	Atributo	Tiene la descripción de la validación del catálogo
Catálogo Validación.Encontrar: boolean	Atributo	Indica si la validación es para que el valor del campo se encuentre en el catálogo o no se encuentre
Entidad Federativa	Concepto	Tiene el catálogo de las entidades federativas
Entidad Federativa.Identificador de entidad federal: int	Atributo	Tiene la clave que identifica la entidad federal
Entidad Federativa.Descripción: string	Atributo	Tiene el nombre de la entidad federativa
Entidad Federativa.Extranjera: boolean	Atributo	Indica si la entidad federal es extranjera o no
Salario Mínimo	Concepto	Tiene el historial del monto (diario) del salario mínimo
Salario Mínimo.Fecha : date	Atributo	Tiene la fecha a partir de cuando es vigente el monto del salario mínimo

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Salario Mínimo.Monto de Salario: double	Atributo	Tiene el monto del salario mínimo
Tipo de Entidad	Concepto	Se tiene el catálogo de los tipos de entidad involucrados con la validación de archivos de envío a CONSAR
Tipo de Entidad.Identificador de tipo de entidad: string	Atributo	Tiene la clave que identifica el tipo de entidad
Entidad.Identificador de tipo de entidad: string	Atributo	Tiene la clave que identifica el tipo de entidad
Tipo de Entidad.Descripción: string	Atributo	Tiene el nombre del tipo de entidad
Entidad	Concepto	Se tiene el catálogo de entidades por tipo de entidad
Entidad.Identificador de entidad: string	Atributo	Tiene la clave que identifica la entidad
Entidad.Descripción: string	Atributo	Tiene el nombre de la entidad
Cuentas de Retiro	Concepto	Se tiene el catálogo de Cuentas de Retiro publicado en el Diario oficial de la federación
Cuentas de Retiro.Cuenta retiro: string	Atributo	Tiene la cuenta de retiro
Cuentas de Retiro.Descripción: string	Atributo	Tiene el nombre de la cuenta de retiro
Subcuenta de Retiro	Concepto	Se tiene el catálogo de subcuentas de retiro agrupadas por cuentas de retiro publicado en el Diario oficial de la federación
Subcuenta de Retiro.Cuenta de Retiro: string	Atributo	Tiene la Cuenta de retiro que incluye a la subcuenta
Subcuenta de Retiro.Subcuenta: string	Atributo	Tiene la subcuenta de retiro
Subcuenta de Retiro.Descripción: string	Atributo	Tiene la descripción de la subcuenta

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<b>Término</b>	<b>Categoría</b>	<b>Comentarios</b>
Cuentas de Traspaso	Concepto	Se tiene el catálogo de Cuentas de Traspaso publicado en el Diario oficial de la federación
Cuentas de Traspaso.Cuenta retiro	Atributo	Tiene la cuenta de traspaso
Cuentas de Traspaso.Descripción	Atributo	Tiene el nombre de la cuenta de traspaso
Subcuenta de Traspaso.Cuenta de Traspaso: string	Atributo	Tiene la Cuenta de traspaso que incluye a la subcuenta
Subcuenta de Traspaso	Concepto	Se tiene el catálogo de subcuentas de traspaso agrupadas por cuentas de traspaso publicado en el Diario oficial de la federación
Subcuenta de Traspaso.Subcuenta: string	Atributo	Tiene la subcuenta de retiro
Subcuenta de Traspaso.Descripción: string	Atributo	Tiene la descripción de la subcuenta
Tipo de Operación	Concepto	Se tiene el catálogo de tipos de operación que existen en el proceso de sistema de ahorro para el retiro <sup>141</sup>
Tipo de Operación.Identificador del Tipo de operación: int	Atributo	Tiene la clave del tipo de operación relacionado con el archivo
Tipo de Operación.Descripción: string	Atributo	Tiene el nombre y/o descripción del tipo de operación

**Tabla 3.8 Continuación** de Glosario resultante del caso práctico

<sup>141</sup> Los tipos de operación involucrados son: afiliación, recaudación, traspasos, retiros, unificación de cuentas, contable, inversiones, estados de cuenta

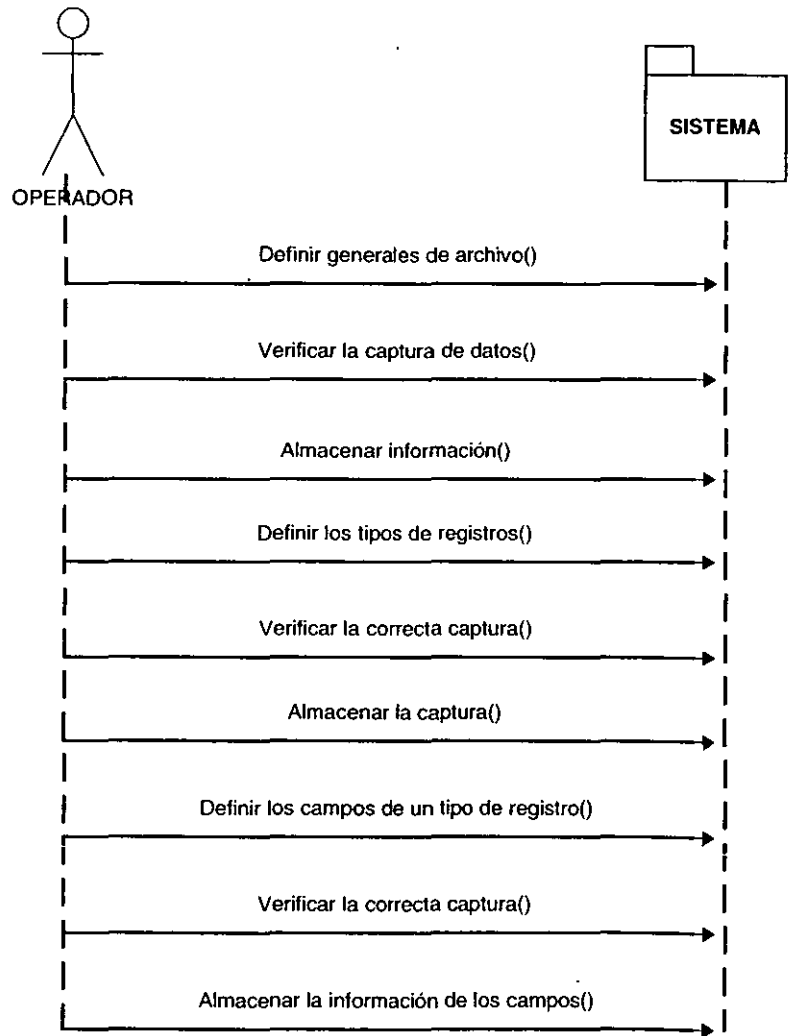


### 5to. PASO    DIAGRAMAS DE SECUENCIA DEL SISTEMA

**CASO DE USO:**  
Definir la estructura de un archivo

Curso normal de los eventos:

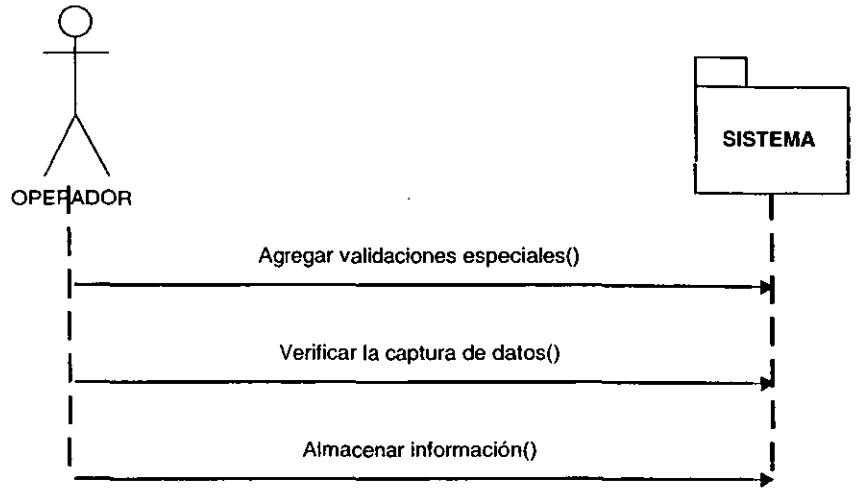
1. El OPERADOR define los datos generales del archivo
2. El Sistema verifica la correcta captura de información
3. El Sistema almacena la información en la base de datos
4. El OPERADOR define los tipos de registros que contendrá el archivo
5. El Sistema verifica la correcta captura de información
6. El Sistema almacena la información en la base de datos
7. El OPERADOR define los campos y las características de cada uno de ellos
8. El sistema verifica la correcta captura de información
9. El sistema almacena la información en la base de datos



**CASO DE USO:**  
Definir validaciones especiales de un campo de un archivo

Curso normal de los eventos:

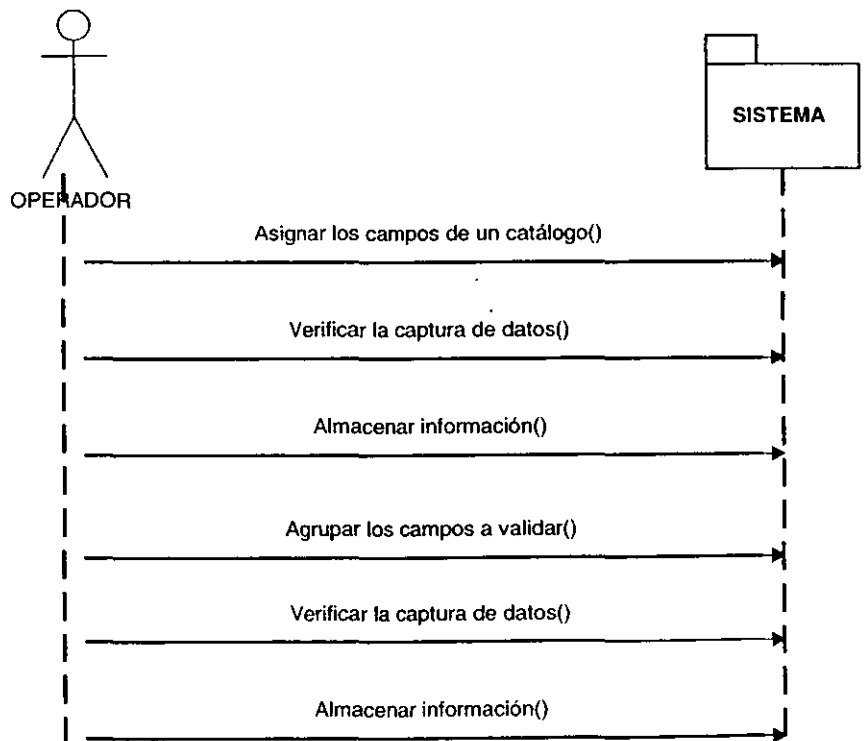
1. El OPERADOR agrega cada una de las validaciones especiales, asociadas a cada uno de los campos, capturando el tipo de validación, los formatos y los valores constantes
2. El Sistema verifica la correcta captura de información
3. El Sistema almacena la información en la base de datos



**CASO DE USO:**  
Definir las validaciones de existencia de un campo en catálogos

Curso normal de los eventos:

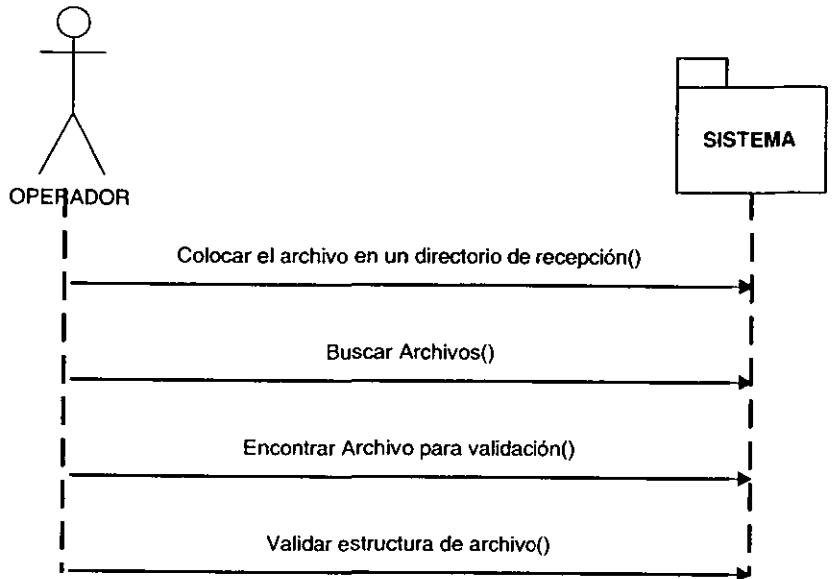
1. El OPERADOR asigna a cada campo el campo de un catálogo contra el que se validará
2. El Sistema verifica la correcta captura de información
3. El Sistema almacena la información en la base de datos
4. El OPERADOR agrupará los campos del archivo contra los campos que validarán su existencia en un catálogo
5. El Sistema verifica la correcta captura de información
6. El Sistema almacena la información en la base de datos



**CASO DE USO:**  
Detectar llegada de archivo

Curso normal de los eventos:

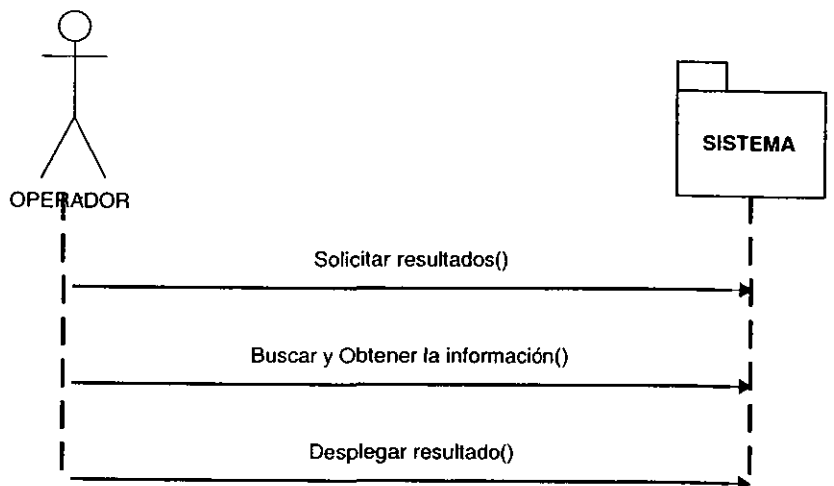
1. El OPERADOR coloca el archivo en el directorio que se espera encontrar
2. El Sistema busca archivos en los directorios definidos
3. El Sistema encuentra el archivo y hace un registro en la bitácora acciones
4. El Sistema valida la estructura del archivo haciendo el registro de cada acción en la bitácora



**CASO DE USO:**  
Consultar resultado

Curso normal de los eventos:

1. El OPERADOR solicitará el estatus y el resultado de la validación
2. El Sistema busca lo solicitado en la bitácora
3. El Sistema desplegará los resultados



**6to. PASO    CONTRATOS**

<b>CONCEPTO</b>	<b>DESCRIPCION</b>
Nombre	Definir generales de archivo
Responsabilidades	Capturar(registrar) las características generales de un archivo
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir la estructura de un archivo
Notas	De fácil entendimiento para el usuario y sólo se requiere que se defina una vez los campos capturados.
Excepciones	Si el archivo que se esta definiendo ya se encontraba almacenado que se despliegue información
Precondiciones	El archivo no estaba anteriormente definido
Poscondiciones	<ul style="list-style-type: none"><li>• Si es un archivo nuevo se creó un tipo de archivo (se creó una instancia)</li><li>• Se asoció a una instancia de directorio recepción (asociación formada)</li><li>• Se asoció a verificar la captura de datos (asociación)</li></ul>

**Tabla 3.9<sup>142</sup>** Contrato de Definir generales de archivo

<sup>142</sup> Fuente: Creación Propia

<b>CONCEPTO</b>	<b>DESCRIPCIÓN</b>
Nombre	Verificar la captura de datos
Responsabilidades	Validar que se haya especificado completa la información obligatoria, que los tipos de valores capturados sean correctos
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Definir la estructura de un archivo
Notas	Valida que se capture la información obligatoria Valida que el tipo de datos corresponda al definido Disminuye la posibilidad de error en la operación de almacenado
Excepciones	Si la captura no fue válida que se informe al usuario
Precondiciones	Lo que se esta definiendo no esta creado
Poscondiciones	<ul style="list-style-type: none"> <li>• Si lo definido no fue nuevo se generó una instancia de error (creación de instancia)</li> <li>• Se creó una asociación con almacenar información (asociación formada)</li> </ul>

**Tabla 3.10<sup>143</sup>** Contrato de Verificar la captura de datos

<sup>143</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCIÓN</b>
Nombre	Almacenar Información
Responsabilidades	Hacer el registro de la información capturada
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Definir la estructura de un archivo
Notas	Asegura el correcto almacenamiento de la información
Excepciones	En caso de que ocurra algún error al momento de guardar la información se captará este y se le informará al usuario.
Precondiciones	El archivo no estaba anteriormente definido y la información capturada es correcta
Poscondiciones	<ul style="list-style-type: none"><li>• Se acepto los valores para la instancia generada</li><li>• Si el almacenamiento fue exitoso (se eliminó la instancia)</li><li>• La información capturada esta lista para su uso</li></ul>

**Tabla 3.11**<sup>144</sup> Contrato de Almacenar Información

---

<sup>144</sup> Idem

CONCEPTO	DESCRIPCIÓN
Nombre	Definir los tipos de registros
Responsabilidades	Capturar las características que definen un tipo de registro de un tipo de archivo
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir la estructura de un archivo
Notas	Captura de fácil entendimiento para el usuario Solo se requiere que se defina una vez.
Excepciones	Si el tipo de registro que se esta definiendo ya se encuentra almacenado que le informe al usuario
Precondiciones	El tipo de archivo asociado ya esta definido El tipo registro no estaba anteriormente definido
Poscondiciones	<ul style="list-style-type: none"> <li>• Si es un tipo de registro nuevo se creó un tipo de registro (se creó una instancia)</li> <li>• Se asoció al tipo de archivo (asociación formada)</li> <li>• Se asoció a verificar captura de datos(asociación formada)</li> </ul>

**Tabla 3.12<sup>145</sup>** Contrato de Definir los tipos de registros

CONCEPTO	DESCRIPCIÓN
Nombre	Agregar validaciones especiales
Responsabilidades	Asociar a un campo las validaciones necesarias
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir validaciones especiales de un campo de un archivo
Notas	Asignar a un campo cada una de las validaciones que se requiera hacer
Precondiciones	El campo, comparadores y funciones de validación están definidos
Poscondiciones	<ul style="list-style-type: none"> <li>• Si es una validación nueva se creó una validación especial(se creó una instancia)</li> <li>• Se asoció a un campo de un tipo de archivo (asociación formada)</li> </ul>

**Tabla 3.13<sup>146</sup>** Contrato de Agregar validaciones especiales

<sup>145</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCIÓN</b>
Nombre	Definir los campos de un tipo de registro
Responsabilidades	Capturar(registrar) las características generales de un campo de un tipo de registro en un tipo de archivo
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir la estructura de un archivo
Notas	De fácil entendimiento para el usuario Solo se requiere que se defina una vez Facilita la validación de información
Excepciones	Si el campo que se está definiendo ya se encontraba almacenado que se informe Si no existe el tipo de registro asociado que lo informe Si no existe el tipo de archivo asociado que lo informe
Precondiciones	El tipo de archivo está definido El tipo de registro está definido El campo no se encuentra definido
Poscondiciones	<ul style="list-style-type: none"> <li>• Si es un campo nuevo se creó un campo (se creó una instancia)</li> <li>• Se asoció a Tipo de Archivo (asociación formada)</li> <li>• Se asoció a Tipo de registro (asociación formada)</li> <li>• Se asoció a verificar captura de datos(asociación formada)</li> </ul>

**Tabla 3.14<sup>147</sup>** Contrato de Definir los campos de un tipo de registro

<sup>146</sup> Idem

<sup>147</sup> Idem



CONCEPTO	DESCRIPCIÓN
Nombre	Asignar los campos de un catálogo
Responsabilidades	Registrar la asociación de un campo de un tipo de registro al campo de un catálogo con el que se desea validar
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir las validaciones de existencia de un campo en catálogos
Notas	Asigna el campo que se desean validar al campo correspondientes del catálogo De fácil comprensión para el usuario Flexibilidad de cambio
Excepciones	Si no corresponde el tipo de dato desplegar mensaje
Precondiciones	El catálogo y el campo están definidos
Poscondiciones	<ul style="list-style-type: none"> <li>• Si es una asociación nueva se creó una asignación de campos de un catalogo(se creó una instancia)</li> <li>• Se asoció a un campo de un tipo de archivo (asociación formada)</li> </ul>

**Tabla 3.15<sup>148</sup>** Contrato de Asignar los campos de un catálogo

<sup>148</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCIÓN</b>
Nombre	Agrupar los campos a validar
Responsabilidades	Registrar la asociación de los campos de un tipo de registro a los campos de un catálogo con el que se desea validar indicando la asociación
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Definir las validaciones de existencia de un campo en catálogos
Notas	Asigna el campo que se desean validar al sus campos correspondientes del catálogo Indicar la agrupación de los campos de un tipo de registro De fácil comprensión para el usuario Flexibilidad de cambio
Excepciones	Si no corresponden los tipos de dato desplegar información
Precondiciones	El catálogo y los campos están definidos
Poscondiciones	<ul style="list-style-type: none"> <li>• Si es una asociación nueva se creó una agrupación de campos a validar(se creó una instancia)</li> <li>• Se asoció a un campo de un tipo de registro (asociación formada)</li> <li>• Se asoció a un catálogo (asociación formada)</li> </ul>

**Tabla 3.16<sup>149</sup>** Contrato de Agrupar los campos a validar

<sup>149</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCIÓN</b>
Nombre	Colocar el archivo en un directorio de recepción
Responsabilidades	Se pone el archivo en alguno de los directorios que están definidos como recepción para el tipo de archivo que se reporte
Tipo	Interfaz con el usuario
Referencias Cruzadas	Caso de Uso: Detectar llegada de archivo
Notas	El OPERADOR coloca, copia el archivo a validar en algún directorio que espere
Excepciones	Si el archivo es puesto en un directorio no definido para el tipo de archivo, no se hará las validaciones que le corresponden.
Precondiciones	El directorio deberá estar asignado al tipo de archivo que se va a validar
Poscondiciones	Quedó el archivo en espera de ser encontrado por el sistema

**Tabla 3.17**<sup>150</sup> Contrato de Colocar el archivo en un directorio de recepción

<sup>150</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCION</b>
Nombre	Buscar Archivo
Responsabilidades	Hacer un recorrido por todos los directorios definidos en busca de nuevos archivos
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Detectar llegada de archivo Funciones: R1.1
Notas	Directorio por directorio se busca archivos nuevos
Excepciones	Si las rutas definidas para recepción de archivos no son validas para el servidor entonces desplegar mensaje
Precondiciones	Las rutas de los directorios de recepción deben estar registrados
Poscondiciones	<ul style="list-style-type: none"> <li>• Por cada archivo encontrado creó una bitácora(se creó una instancia)</li> <li>• Se asoció a una instancia de directorio recepción (asociación formada)</li> </ul>

**Tabla 3.18**<sup>151</sup> Contrato de Buscar Archivo

<sup>151</sup> Idem

CONCEPTO	DESCRIPCION
Nombre	Encontrar archivo para validación
Responsabilidades	Encontrar un archivo para validar y registrarlo como listo para la verificación.
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Detectar llegada de archivo Funciones: R1.1
Notas	Encuentra el archivo de manera oportuna para su validación
Excepciones	Si el archivo esta dañado o tiene un formato inválido despliega un mensaje
Precondiciones	El directorio esta definido
Poscondiciones	<ul style="list-style-type: none"> <li>• Se creó una instancia en bitácora (se creó una instancia)</li> <li>• Se asoció a una instancia de directorio recepción (asociación formada)</li> <li>• Se asoció a un tipo de archivo (asociación formada)</li> <li>• Se asoció a un tipo de entidad (asociación formada)</li> </ul>

**Tabla 3.19<sup>152</sup>** Contrato de Encontrar archivo para validación

<sup>152</sup> Idem

<b>CONCEPTO</b>	<b>DESCRIPCION</b>
Nombre	Validar estructura de archivo
Responsabilidades	Hacer lo siguiente en el archivo validado: Verificar que el archivo cumpla con el formato establecido por el tipo de archivo que lo define Verificar los tipos de registro obligatorios se estén reportando Verificar que los campos obligatorios se estén reportando Verificar que se cumplan las validaciones especiales Verificar que se cumplan las validaciones de asignación de campos a catálogos
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Detectar llegada de archivo Funciones: R1.2
Notas	Una vez encontrado el archivo se empieza con su validación Validar la estructura del archivo Validar los tipos de registro y campos obligatorios Verificar todas las validaciones especiales de los campos Verificar todas las validaciones de asignación de catálogos
Excepciones	Si el directorio donde se encontró el archivo no le corresponde entonces desplegar un mensaje de error
Precondiciones	El archivo fue encontrado en algún directorio que le corresponde

**Tabla 3.20<sup>153</sup>** Contrato de Validar estructura de archivo

<sup>153</sup> Idem

Poscondiciones	<ul style="list-style-type: none"><li>• Se creó un registro en bitácora por cada acción ejecutada durante la validación del archivo (se creó una instancia)</li><li>• Se asoció a una instancia de directorio recepción (asociación formada)</li><li>• Se asoció a Tipo de archivo (asociación formada)</li><li>• Se asoció a Tipo de registro (asociación formada)</li><li>• Se asoció a Campo (asociación formada)</li><li>• Se asoció a Validación especial (asociación formada)</li><li>• Se asoció a Tipo de archivo (asociación formada)</li><li>• Se asoció a Error (asociación formada)</li><li>• Se asoció a Tipo de entidad (asociación formada)</li><li>• Se asoció a Entidad (asociación formada)</li><li>• Se asoció a Guía Contable (asociación formada)</li><li>• Se asoció a Cuentas de Retiro (asociación formada)</li><li>• Se asoció a Subcuentas de Retiro (asociación formada)</li><li>• Se asoció a Cuentas de Traspasos (asociación formada)</li><li>• Se asoció a Subcuentas de Traspasos (asociación formada)</li><li>• Se asoció a Entidad federativa (asociación formada)</li><li>• Se asoció a Penalización (asociación formada)</li></ul>
----------------	--

**Tabla 3.20 Continuación** de Contrato de Validar estructura de archivo

CONCEPTO	DESCRIPCION
Nombre	Solicitar resultados
Responsabilidades	Hacer la petición parametrizada de consulta de resultados.
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Consultar resultados
Notas	La captura de parámetros es sencilla y fácil entendimiento para el usuario Presenta los resultados en un tiempo breve Muestra los resultados en forma detallada
Precondiciones	El archivo a consultar ha sido verificado
Poscondiciones	<ul style="list-style-type: none"> <li>• Se crearon las condiciones de búsqueda de resultados creando Solicitar resultados(se creó una instancia)</li> <li>• Se asoció a una instancia de bitácora(asociación formada)</li> <li>• Se asoció a Tipo de Archivo (asociación formada)</li> <li>• Se asoció a Tipo de Operación (asociación formada)</li> <li>• Se asoció a Directorio recepción (asociación formada)</li> <li>• Se asoció a Tipo de entidad (asociación formada)</li> <li>• Se asoció a Entidad (asociación formada)</li> </ul>

**Tabla 3.21<sup>154</sup>** Contrato de Solicitar resultados

<sup>154</sup> Idem



CONCEPTO	DESCRIPCIÓN
Nombre	Buscar y obtener la información
Responsabilidades	Buscar los resultados obtenidos de la validación de un archivo
Tipo	Sistema
Referencias Cruzadas	Caso de Uso: Consultar resultados Funciones: R1.4
Notas	En base a los parámetros de solicitud de resultados se obtiene o consulta la información de la validación registrada en la bitácora
Precondiciones	El archivo a consultar esta validado
Poscondiciones	<ul style="list-style-type: none"> <li>Se consultó la información requerida de la bitácora (se creó una instancia)</li> <li>Se asoció a solicitar resultados (asociación formada)</li> </ul>

**Tabla 3.22**<sup>155</sup> Contrato de Buscar y obtener la información

CONCEPTO	DESCRIPCION
Nombre	Desplegar resultado
Responsabilidades	Desplegar de manera entendible al usuario el resultado solicitado de la validación de un archivo
Tipo	Clase
Referencias Cruzadas	Caso de Uso: Consultar resultados Funciones: R1.4
Notas	Desplegar en forma rápida los resultados solicitados
Excepciones	Si no existe información desplegar mensaje que lo indique
Precondiciones	La consulta de resultados ya se realizó
Poscondiciones	<ul style="list-style-type: none"> <li>Se desplegó los resultados deseados en una instancia desplegar resultado(se creó una instancia)</li> <li>Se asoció a la bitácora (asociación formada)</li> <li>Se asoció a Solicitar resultados (asociación formada)</li> <li>Se asoció a Penalización (asociación formada)</li> <li>Se asoció a Política en Archivo (asociación formada)</li> </ul>

**Tabla 3.23**<sup>156</sup> Contrato de Desplegar resultado

<sup>155</sup> Idem

## **3.2. Evaluación de Resultados**

### **3.2.1. Pros y Contras de la metodología estructurada "Yourdon"**

Considerando la evaluación subjetiva como parte de la *planeación*, tenemos que:

**Pros:**

- Esta metodología se aprende rápidamente,
- Es de fácil entendimiento para cualquier persona
- Sus etapas están interconectadas
- Conducen al planteamiento de funciones o procesos
- Desde la etapa de análisis va señalando al analista como se puede realizar los procesos
- Existe mucho más experiencia en este tema

**Contras:**

- No enfatiza en las entidades involucradas por lo que se pueden omitir situaciones importantes
- No es posible utilizarla para programación orientada a objetos
- Sólo puede representar un número limitado de relaciones entre datos
- Es compleja la representación de subtipos ya que requiere un producto cartesiano
- No existe el concepto de clasificación múltiple (clases compuestas)
- Presenta problemas al expresar las entidades relacionadas con el tiempo

### **3.2.2. Pros y Contras de la metodología orientada a objetos "UML"**

**Pros:**

- Orienta al analista a considerar a todas las personas y entidades involucradas
- Es de fácil interpretación
- Contempla todos los detalles involucrados en el sistema
- Obliga a encontrar los huecos que pudieran existir en el análisis, obligando a considerar más de las situaciones aunque algunas sean obvias, el pensamiento es "más vale que sobre"

**Contras:**

- Sus etapas toman tiempo considerable en su desarrollo
- Se debe contar con experiencia sólida en la programación orientada a objetos
- En la etapa de análisis no se involucra tanto en el como se conseguirán los resultados
- No existen en el mercado muchas bases de datos orientadas a objetos
- Presenta problemas al expresar las entidades relacionadas con el tiempo

---

<sup>156</sup> Idem

### 3.3. Presentación de los cambios al procedimiento de Desarrollo de Aplicaciones a la Medida

Como se mostró en el capítulo I apartado 1.8, actualmente el procedimiento cuenta con 8 actividades en la etapa de análisis, ahora bien, el cambio propuesto se presenta en la tabla 3.24

Actividad	ACTUALMENTE	Actividad	PROPUESTO
1	Elaboración del plan de trabajo de Análisis	1	Se queda igual
2	Obtener y documentar Entidades involucradas, Entradas y salidas y Relaciones entre entidades	2	Entrevistas/Requerimientos
3	Elaborar Diagrama de Contexto General/Diagrama de Negocio	3	El modelo Ambiental que esta compuesto por: Propósito, Diagrama de contexto y Lista de Acontecimientos/Casos de Uso
4	Revisión y Aprobación del Diagrama de contexto/Diagrama de Negocio	4	Revisión y Aprobación del Modelo Ambiental/ Casos de Uso (Definición y Diagramas)
5	Lista de eventos		Esta actividad esta contemplada en el Modelo ambiental
6	Elaborar Diagrama de Flujo de Datos (DFD)/Casos de Uso	5	Elaborar Diagramas de Flujo de Datos (DFD)/ Modelo Conceptual
7	Revisión y Aprobación del DFD/Casos de Uso	6	Revisión y Aprobación del DFD/Modelo Conceptual
8	Especificaciones del Proceso/Documentación de Casos de Uso	7	Especificaciones del Proceso (para metodología de Yourdon)

**Tabla 3.24<sup>157</sup>** Etapa de Análisis el existente y el propuesto

<sup>157</sup> Fuente: Creación Propia

Actividad	ACTUALMENTE	Actividad	PROPUESTO
		8	Revisión y Aprobación de las Especificaciones del Proceso
		9	Diccionario de Datos/Glosario
		10	Diagrama Entidad Relación (DER)/Diagramas de Secuencia
		11	Revisión y Aprobación del DER/Diagramas de Secuencia
		12	Diagrama de Transición de Estados (DTE)/Contratos
		13	Revisión y Aprobación de DTE/Contratos

**Tabla 3.24 Continuación de Etapa de Análisis el existente y el propuesto**

Las *Guías y Formatos*. Estos son lo desarrollado en el capítulo II, por ejemplo tenemos la actividad propuesta número 5 ( en la tabla 3.24), referente a la elaboración de los diagramas de flujo de datos, en el caso de la metodología estructurada, entonces la guía es lo descrito en el punto 2.4.1 Paso 3. Y para el Modelo Conceptual, en el caso de la metodología UML, se debe revisar el punto 2.4.2 Paso 2.

Los *Ejemplos*. De manera similar a lo indicado en el párrafo anterior, para tener los ejemplos se debe consultar lo desarrollado en el presente capítulo. Ahora tenemos que si se desea ver el ejemplo de la actividad 5; para la metodología estructurada, se debe consultar el punto 3.1.2 el paso 3 y para la metodología UML será el punto 3.1.3 paso 2.

Cumpliendo con esto con las Guías, Formatos y ejemplos para cubrir los puntos de : 1) El desarrollo completo de una metodología y 2) La normativa de ISO 9001.

Los responsables de las actividades propuestas, requisito de la normativa, están distribuidas como se muestra en la tabla 3.25

Actividad	Responsable
1	Responsable del Proyecto (Líder de proyecto y/o Gerente de proyecto <sup>158</sup> )
2	Analista
3	Analista
4	Responsable del Proyecto (Líder de Proyecto y/o Gerente de proyecto)
5	Analista
6	Responsable del Proyecto (Líder de Proyecto y/o Gerente de proyecto)
7	Analista
8	Responsable del Proyecto (Líder de Proyecto y/o Gerente de proyecto)
9	Analista
10	Analista
11	Responsable del Proyecto (Líder de Proyecto y/o Gerente de proyecto)
12	Analista
13	Responsable del Proyecto (Líder de Proyecto y/o Gerente de proyecto)

**Tabla 3.25<sup>159</sup>** Asignación de Responsables de las actividades propuestas

El actual procedimiento en su etapa de análisis cubre con los requisitos 4.4, 4.20 del NMX-CC-003:1995 IMNC<sup>160</sup>; dichos requisitos, únicamente se relacionan con la etapa de análisis y se refieren a:

<sup>158</sup> Título o nombramiento a la persona que administra un proyecto/ sistema de información, esta asignación depende del comité ejecutivo de la empresa consultora

<sup>159</sup> Fuente: Creación Propia

<sup>160</sup> NMX-CC-003:1995 IMNC, "Sistema de Calidad- Modelo para el aseguramiento de la calidad en diseño, desarrollo, producción, instalación y servicio", ISO 9001:1994

## **4.4 Control del diseño.**

### **4.4.1 Generalidades**

El proveedor debe establecer y mantener procedimientos documentados para controlar y verificar el diseño del producto, con el fin de asegurar que se cumplan los requisitos especificados.

### **4.4.2 Planeación del diseño y desarrollo**

El proveedor debe elaborar planes para cada actividad de diseño y desarrollo. Los planes deben describir o hacer referencia a estas actividades, y definir la responsabilidad para su implantación.

### **4.4.3 Interrelaciones organizacionales y técnicas**

Deben estar definidas las interrelaciones organizacionales y técnica entre los deferentes grupos que proporcionan datos de entrada para el proceso del diseño, y la información necesaria debe estar documentada, y ser transmitida y revisada regularmente.

### **4.4.4 Datos de entrada del diseño**

Se deben identificar y documentar los requisitos para los datos de entrada del diseño relacionados con el producto. Los datos de entrada del diseño deben tomar en consideración los resultados de cualquiera de las actividades de revisión del contrato.

Aunque en estos puntos se nombra al diseño y no al análisis, se debe recordar que el diseño es una consecuencia de la fase de análisis. Retomando lo definido en el capítulo I apartado 1.4.1, en lo referente a la normativa, en este momento es preciso recordar que la Normativa *no* obliga a que se desarrollen todos los requisitos que contiene, dejando esta decisión al libre albedrío. Como se ve la etapa de análisis propuesta sigue con el cumplimiento del requisito 4.4 en sus incisos 4.4.1, 4.4.2, 4.4.3, 4.4

## **3.4. Una Percepción del DAM**

Por otra parte como una herramienta de percepción del presente estudio, se utiliza la generación de escenarios, como lo define Van Der Heijden: *Los escenarios son un conjunto de futuros razonablemente pausables, pero diferentes desde el punto de vista estructural.* En la creación de los escenarios expuestos a continuación se utilizó el proceso mental de causa-efecto. Esto se hizo con el fin de proporcionar una visión sobre como se comportarán las variables relacionadas con la aplicación de una metodología de análisis y diseño de sistemas en un proceso de certificación de calidad de una empresa consultora.

Los escenarios generados son seis; dos optimistas definidos como los positivos que tendrían los mejores resultados esperados; dos pesimistas definidos como los negativos y que presentarían las peores condiciones; y por último dos esperados definidos como los promedio estos se consideran como los de mayor probabilidad de ocurrencia. Estos escenarios fueron creados con una visión en dos tiempos para el año 2006 y para el año 2021, el primer año horizonte es por la importancia del cambio de sexenio en nuestro país y la segunda para visualizar el impacto a largo plazo de lo que hoy puede ser una buena decisión.

A continuación se expone para cada uno de los años horizontes propuestos; un escenario optimista, un pesimista y un esperado.

**AÑO HORIZONTE: 2006**

**ESCENARIO OPTIMISTA** (Con un porcentaje de ocurrencia del 35%)

La situación para que se cumpla el escenario será: Existirá una posibilidad de un 30% de que habrá una convergencia política, mejorando la situación económica del país en un 7%. Habrá un impulso de cambio en el pensamiento del país orientando en un 80% la educación a la tecnología y a la información. El país se abrirá a la competencia mundial con respecto a la capacidad de generación de software, orientado en un 80% a la calidad en sus productos y servicios.

<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<b><i>Mercado de las Aplicaciones a la Medida</i></b> Solicitud de desarrollo de Aplicaciones computacionales a la medida	La solicitud de desarrollo de aplicaciones a la medida se incrementará con base a la generación de nuevas empresas con nuevos giros en un 15%, tendiendo a ser constante, debido a la mejora de la situación económica del país y a la globalización de los mercados; esto llevará a que el 90% de las empresas tendrán la necesidad de aumentar su participación en el mercado tanto nacional como internacional, requiriendo así que sus procesos especializados sean computarizados. Esto con una probabilidad de ocurrencia del 40%.
<b><i>Nuevas técnicas de programación</i></b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis	La creación de alguna nueva técnica de programación puede salir en cualquier momento, pero es poco probable, se espera un 3% de probabilidad de que sea una técnica que ocupe el lugar de alguna o de las técnicas más utilizadas. Por lo que no se requerirá de el estudio de nuevas metodologías de análisis y diseño de sistemas. Con una probabilidad de ocurrencia del 50%.

<p><b>Situación Empresarial</b> Afectación en la empresa consultora</p>	<p>La competencia de pequeñas empresas crecerá en un 80%, sin embargo el ofrecer una certificación de calidad a nivel internacional no lo podrán hacer. Permitiendo con esto mantener una ventaja competitiva a nivel nacional e internacional, aumentando la participación en el mercado mundial en un 30%, generando seguramente el crecimiento de la empresa. Con una probabilidad de ocurrencia del 75%.</p>
<p><b>Cultura de Análisis y Diseño de Sistemas</b> Involucración de la gente en el conocimiento de alguna metodología de análisis</p>	<p>Existirá una mejor comunicación entre los usuarios y los analistas de sistemas ya que el conocimiento de las metodologías de análisis y diseño será parte de la formación educativa desde la escuela básica; y también se considerará formalmente en la capacitación proporcionada en las empresas. Con una probabilidad de ocurrencia del 30%.</p>
<p><b>Certificación de Calidad</b> Panorama de la búsqueda de una certificación internacional</p>	<p>La búsqueda de calidad es continua y las actualizaciones a metodologías de análisis y diseño de sistemas también lo es; sin embargo estos cambios no se presentan de manera inmediata, permitiendo que se realicen las modificaciones necesarias en los procedimientos de calidad generados y así continuar con la búsqueda de la certificación internacional de calidad. Con una probabilidad de ocurrencia del 80%.</p>
<p><b>Herramientas Metodológicas de análisis y diseño de sistemas</b> Generación y modificaciones de metodologías</p>	<p>El surgimiento de nuevas metodologías de análisis y diseño de sistemas, así como de actualizaciones en las ya existentes, requerirá de su evaluación haciendo necesarias alteraciones en los procedimientos de calidad ya establecidos, para su mejora asegurando de mejor manera la calidad en los programas desarrollados. Con una probabilidad de ocurrencia del 50%.</p>



**ESCENARIO PESIMISTA** (Con un porcentaje de ocurrencia del 15%)

La situación para que se cumpla el escenario será: Con una probabilidad del 70% habrá resistencia a la igualdad de fines entre los partidos políticos por lo que no habrá cambios estructurales ni crecimiento económico. El nivel de educación tecnológica sólo sufrirá pocos cambios en un 20%; permaneciendo cerrados a la competencia mundial. La creación de grupos de protesta se incrementará en un 70% lo que no permitirá inversión extranjera. Tecnológicamente existe una probabilidad del 5% en que aparezca un nuevo pensamiento de programación que sustituya los existentes, este será establecido como el lenguaje mundial de programación.

<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<p><b><i>Mercado de las Aplicaciones a la Medida</i></b> Solicitud de desarrollo de Aplicaciones computacionales a la medida</p>	<p>El mercado de aplicaciones a la medida disminuye en un 30%, sustituyéndose por la utilización de las aplicaciones especializadas conocidas como ERP (Enterprise Resource Planning) ejemplo de ellas son SAP, PeopleSoft, J.D. Edwards, Oracle ERP, Baan, etc., las cuales sólo requieren un porcentaje mínimo de programación haciendo adecuaciones del software modelo, según las necesidades de la empresa cliente. Disminuyendo el costo y tiempo requerido para su puesta en marcha, contra los costos y tiempos que pudiera ocupar un desarrollo a la medida. Con una probabilidad de ocurrencia del 70%.</p>
<p><b><i>Nuevas técnicas de programación</i></b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis</p>	<p>La generación de una nueva técnica de programación requerirá la creación de nuevas formas de pensamiento, lo que llevará a la necesidad de nuevas metodologías de análisis y diseño, por lo que los procedimientos para obtener calidad en las aplicaciones a la medida tendrán que estar en una constante actualización dejando obsoletas las versiones generadas. Obligando al departamento de calidad a permanecer a la punta en estos procedimientos. Con una probabilidad de ocurrencia del 15%.</p>

<p><b>Situación Empresarial</b></p> <p>Afectación en la empresa consultora</p>	<p>La creación de varias empresas especializadas por cada técnica de programación nueva (generalmente se trata de empresas pequeñas, con personal menor a 50 gentes), generará una competencia que afectará definitivamente a la captación de clientes en un 80%, disminuyendo por lo tanto la participación en el mercado, deteniendo el crecimiento de la empresa consultora; esta última carecerá del personal suficientemente capacitado en la nueva técnica de programación y mucho menos en las nuevas metodologías de análisis y diseño.</p> <p>La participación en el mercado internacional será casi nula. Ya que no se contará con el prestigio suficiente sobre las nuevas tecnologías, también porque será muy difícil ofrecer la certificación de calidad necesaria en el software ofrecido.</p> <p>Con una probabilidad de ocurrencia del 30%.</p>
<p><b>Cultura de Análisis y Diseño de Sistemas</b></p> <p>Involucración de la gente en el conocimiento de alguna metodología de análisis</p>	<p>Aunque existe una tendencia de que se proporcione el conocimiento de alguna metodología de análisis y diseño de sistemas dentro de las escuelas y empresas con una probabilidad del 20%. No se podrá tener a la población actualizada de las nuevas formas de pensamiento de programación lo que seguirá afectando a la comunicación entre los analistas de sistemas y los usuarios. Con una probabilidad de ocurrencia del 30%.</p>

<p><b>Certificación de Calidad</b> Panorama de la búsqueda de una certificación internacional</p>	<p>A pesar de los esfuerzos por cumplir con la calidad en los programas generados, cada vez será más difícil cubrir los requerimientos para una certificación debido a la constante actualización de los procedimientos necesarios por las nueva técnicas de programación. Con una probabilidad de ocurrencia del 10%.</p>
<p><b>Herramientas Metodológicas de análisis y diseño de sistemas</b> Generación y modificaciones de metodologías</p>	<p>La generación de nuevas técnicas de programación va de la mano con la creación de las nuevas metodologías de análisis y diseño de sistemas, así como, adecuaciones en las ya existentes. Por lo que se requerirá constantes cambios en los procedimiento de calidad. Con una probabilidad de ocurrencia del 15%.</p>

**ESCENARIO ESPERADO** (Con un porcentaje de ocurrencia del 50%)

La situación para que se cumpla el escenario será: La participación de la sociedad en las cuestiones políticas obligarán al gobierno a ser pluralista orientado al beneficio social, económico y cultural. Incrementando la economía del país en un 5%. La apertura comercial con el extranjero se incrementa en un 60% generando pactos comerciales con estos sin embargo será un país controlado por los EUA. Este último será el mejor cliente del país del cual seremos en un 60% su maquilador de software. Tecnológicamente las metodologías de análisis y diseño de sistemas, así como las técnicas de programación sólo sufrirán mejoras. La educación estará enfocada a la tecnología y a la información. El tema de calidad en todos los productos ofrecidos será una manera de pensar esto con el fin de permanecer en la competencia mundial.

<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<b><i>Mercado de las Aplicaciones a la Medida</i></b> Solicitud de desarrollo de Aplicaciones computacionales a la medida	El mercado de las aplicaciones a la medida tiende a disminuir en un 50%, esto es por la aparición de aplicaciones ERP (Enterprise Resource Planning), sin embargo seguirá existiendo la necesidad de desarrollos especiales. Donde sólo las empresas con mejor prestigio tendrán la oportunidad de participar. Con una probabilidad de ocurrencia del 85%.
<b><i>Nuevas técnicas de programación</i></b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis	La generación de nuevas técnicas de programación que sustituyan las existentes es casi nula, por lo que no afectará a corto plazo la generación de alguna nueva en el uso de las actuales. Con una probabilidad de ocurrencia del 90%.

<p><b>Situación Empresarial</b> Afectación en la empresa consultora</p>	<p>La competencia crece de manera acelerada a un 100%, sin embargo estas empresas nuevas no suelen aplicar metodologías formales de análisis y diseño de sistemas ya que son empresas pequeñas que no invierten en investigación y capacitación para metodologías que les ayuden a mejorar su forma de desarrollo de sistemas. Por esto carecen de prestigio y confiabilidad en el mercado haciendo que el crecimiento de la empresa siga constante, de esta manera se incrementará la participación en el mercado global en un 30%. Con una probabilidad de ocurrencia del 70%.</p>
<p><b>Cultura de Análisis y Diseño de Sistemas</b> Involucración de la gente en el conocimiento de alguna metodología de análisis</p>	<p>La enseñanza de alguna metodología de análisis y diseño de sistemas será parte de la cultura empresarial y de la educación a nivel intermedio (secundaria y/o preparatoria); Agilizando el entendimiento de las metodologías de análisis, así como también se mejorará la comunicación entre el analista de sistemas y el usuario. Con una probabilidad de ocurrencia del 50%.</p>
<p><b>Certificación de Calidad</b> Panorama de la búsqueda de una certificación internacional</p>	<p>La búsqueda de calidad en software se ha convertido como un requisito en todas las empresas que se dedican al desarrollo de sistemas, sin embargo un porcentaje del 10% de ellas buscan la certificación internacional de calidad ya que se requiere inversión de tiempo y dinero que la mayoría no tiene. Siendo esto una ventaja para los que si pueden mantener una certificación internacional. Con una probabilidad de ocurrencia del 80%.</p>

<b>Herramientas Metodológicas de análisis y diseño de sistemas</b>	Sólo existirán adecuaciones en las metodologías de análisis existentes, generando cambios en los procedimientos de calidad en uso en un 15%. Con una probabilidad de ocurrencia del 60%.
Generación y modificaciones de metodologías	

### **AÑO HORIZONTE: 2021**

#### **ESCENARIO OPTIMISTA** (Con un porcentaje de ocurrencia del 30%)

La situación para que se cumpla el escenario será: La probabilidad de que exista democracia política es de un 50%, disminuyendo en un 50% los grupos de inconformidad. Con un crecimiento económico del 20%. La educación estará a nivel mundial, generando profesionistas capaces, que incrementarán la participación tecnológica del país en el mercado global en un 30%.

<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<b>Mercado de las Aplicaciones a la Medida</b> Solicitud de desarrollo de Aplicaciones computacionales a la medida	Se Tendrá un incremento del 10% debido a su participación en Latinoamérica. Las empresas que proporcionen este tipo de servicio serán pocas mundialmente. Con una probabilidad de ocurrencia del 25%.
<b>Nuevas técnicas de programación</b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis	Existirán nuevas técnicas de programación con sus propias metodologías de análisis y diseño, pero estas no tendrán el auge suficiente como para dedicarles tiempo de estudio. Con una probabilidad de ocurrencia del 40%.
<b>Situación Empresarial</b> Afectación en la empresa consultora	La empresa tendrá un crecimiento en su participación global del 20%. La competencia nacional para el desarrollo de aplicaciones a la medida se reduce a lo más a 10 empresas debido a que habrá poca demanda de este tipo de desarrollo. Con una probabilidad de ocurrencia del 20%

<b>Cultura de Análisis y Diseño de Sistemas</b>	Este tema será aplicado en cualquier tipo de empleo por lo que no existirá problemas de comunicación entre usuarios y analistas de sistemas. Con una probabilidad de ocurrencia del 80%.
Involucración de la gente en el conocimiento de alguna metodología de análisis	
<b>Certificación de Calidad</b>	El ofrecer un certificado de calidad será una disposición oficial en cualquier tipo de negocio que ofrezca productos o servicios. Con una probabilidad de ocurrencia del 40%.
Panorama de la búsqueda de una certificación internacional	
<b>Herramientas Metodológicas de análisis y diseño de sistemas</b>	Las ya existentes tendrán mejoras (reduciendo las actividades a realizar por lo tanto el tiempo de ejecución). Se encontrará en estudio las metodologías nuevas para su evaluación. Con una probabilidad de ocurrencia del 40%.
Generación y modificaciones de metodologías	

**ESCENARIO PESIMISTA** (Con un porcentaje de ocurrencia del 30%)

La situación para que se cumpla el escenario será: Políticamente hay una probabilidad del 80% de que sólo existan partidos de centro. Aumentando la desigualdad social en un 70% delimitando las posiciones económicas de la población. La creación de grupos de contra en todo el país aumentará en un 200%. La educación pública será proporcionada con bajo nivel por lo que la población pobre no tendrá oportunidad de desarrollo. El país solo será maquilador no incluyendo las fabricas de software, de las cuales no se tendrá la capacidad para ofrecer este tipo de servicios, esto con una probabilidad del 50%. Resultando una contracción en el crecimiento económico.

<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<b>Mercado de las Aplicaciones a la Medida</b> Solicitud de desarrollo de Aplicaciones computacionales a la medida	Existirá poca demanda de este tipo de desarrollos, reduciéndose en un 80%; estos serán a nivel nacional; y con una probabilidad del 60% desarrollados por empresas extranjeras. Con una probabilidad de ocurrencia del 40%.

<p><b><i>Nuevas técnicas de programación</i></b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis</p>	<p>Existirá a lo más 2 tipos nuevos de técnicas de programación, con sus respectivas metodologías de análisis y diseño. Con una probabilidad de ocurrencia del 40%.</p>
<p><b><i>Situación Empresarial</i></b> Afectación en la empresa consultora</p>	<p>Las consultoría será propiedad de alguna empresa extranjera y el 80% de sus empleados no serán nacionales. Con una probabilidad de ocurrencia del 25%.</p>
<p><b><i>Cultura de Análisis y Diseño de Sistemas</i></b> Involucración de la gente en el conocimiento de alguna metodología de análisis</p>	<p>La población tendrá el conocimiento de alguna metodología de análisis y diseño de sistemas; sin embargo no estará actualizado y no tendrá la oportunidad de ponerlo en práctica; por lo que seguirá el problema de comunicación entre el usuario y el analista. Con una probabilidad de ocurrencia del 30%.</p>
<p><b><i>Certificación de Calidad</i></b> Panorama de la búsqueda de una certificación internacional</p>	<p>Las empresas consultoras de sistemas deberán contar con una certificación de calidad que respalde su trabajo. Con una probabilidad de ocurrencia del 85%.</p>
<p><b><i>Herramientas Metodológicas de análisis y diseño de sistemas</i></b> Generación y modificaciones de metodologías</p>	<p>Las metodologías en práctica estarán en mejora continua por lo que habrá de modificarse los procedimientos de calidad existentes. Con respecto a las nuevas estarán en evaluación y constantes cambios. Con una probabilidad de ocurrencia del 50%.</p>

***ESCENARIO ESPERADO*** (Con un porcentaje de ocurrencia del 40%)

La situación para que se cumpla el escenario será: Existirá democracia política con una probabilidad de un 70%, con un crecimiento económico del 25%. Habrá una apertura comercial del 100% con el resto del mundo. La educación tendrá dos líneas; una es la pública que no ofrecerá educación de nivel mundial y la privada que proporcionará la mejor opción; dejando con esto a la población de bajos recursos (y con pensamiento no progresista), fuera del desarrollo del país. Tecnológicamente se dará servicio de buen nivel y con calidad teniéndose una participación global del 10%.



<b>Variable Involucrada</b>	<b>Situación Esperada</b>
<p><b><i>Mercado de las Aplicaciones a la Medida</i></b> Solicitud de desarrollo de Aplicaciones computacionales a la medida</p>	<p>Serán muy pocos los requerimientos de este tipo de aplicaciones, por lo que existirán pocas empresas dedicadas a este tipo de desarrollos, reduciéndose a menos de 10 consultorías. Con una probabilidad de ocurrencia del 50%.</p>
<p><b><i>Nuevas técnicas de programación</i></b> Aparición de técnicas de programación que requieran nuevas metodologías de análisis</p>	<p>Existirá una nueva técnica de programación, por lo que tendrá sus nuevas metodologías de análisis y diseño, así que requerirá dedicación a su estudio. Con una probabilidad de ocurrencia del 40%.</p>
<p><b><i>Situación Empresarial</i></b> Afectación en la empresa consultora</p>	<p>La competencia se reducirá en un 70%, teniendo clientes constantes. Sin embargo para otro tipo de desarrollos si habrá competencia considerable. La participación en el mercado internacional será estable. Con una probabilidad de ocurrencia del 80%.</p>
<p><b><i>Cultura de Análisis y Diseño de Sistemas</i></b> Involucración de la gente en el conocimiento de alguna metodología de análisis</p>	<p>El conocimiento de metodologías de análisis y diseño serán tema básico en la población por lo que no existirán problemas de entendimiento y comunicación entre los usuarios y los analistas. Con una probabilidad de ocurrencia del 80%.</p>
<p><b><i>Certificación de Calidad</i></b> Panorama de la búsqueda de una certificación internacional</p>	<p>Las empresas dedicadas al desarrollo de software cumplirán con los estándares internacionales de calidad; por lo que estas en un 100% aplicarán alguna metodología de análisis y diseño de sistemas. Con una probabilidad de ocurrencia del 90%.</p>
<p><b><i>Herramientas Metodológicas de análisis y diseño de sistemas</i></b> Generación y modificaciones de metodologías</p>	<p>Las metodologías de análisis y diseño estarán estandarizadas mundialmente por lo que únicamente existirán adecuaciones y actualizaciones en las ya existentes. Sólo se requerirá mantenimiento de mejora en los procedimientos de calidad. Con una probabilidad de ocurrencia del 60%.</p>

Los porcentajes asignados a la posibilidad de ocurrencia de cada escenario así como a sus condiciones y variables fueron definidos de manera apriori, en base a la experiencia de quien escribe y la colaboración de diferentes participantes de la empresa consultora en estudio.

Estos escenarios proporcionan los elementos para juzgar el futuro posible del uso de las metodologías de análisis y diseño, y desde ahora buscar las posibles soluciones a las situaciones que se pudieran presentar. Y así modificar la Planeación estratégica de la empresa consultora.

## **Conclusiones del Capítulo III:**

En este capítulo se tuvo la oportunidad de utilizar la etapa de análisis de las metodologías *de Análisis y Diseño Estructurado "Yourdon"* y *UML* a una aplicación real, esto expuesto en el inciso 3.1 con sus apartados, calificando el uso de estas como necesarias y obligatorias en el desarrollo de cualquier aplicación software.

El uso de una metodología formal en las actividades cotidianas ordena el pensamiento; dando una ayuda, al facilitar la aplicación de una normatividad ISO 9001 a dichos procesos. Se obtuvo entonces en el apartado 3.3 las modificaciones esperadas al procedimiento de calidad "Desarrollo de Aplicaciones a la Medida".

El seguir los pasos planteados en el capítulo dos por cada una de las metodologías en estudio, llevó a caminos diferentes aunque resultados iguales, esto es, ambas cumplieron con el propósito general del caso práctico. Cada una de las metodologías guía al analista a cubrir el objetivo principal del sistema de información, sin embargo lo obliga a pensar de forma diferente, como se planteó en el inciso 3.2; cada una de ellas induce a la creación de diferentes estructuras de datos, diferentes componentes, y a considerar diferentes procesos; obteniendo al final el mismo resultado. Por lo tanto el llevar a cabo cualquiera de las dos metodologías, de acuerdo al procedimiento DAM, que el responsable del desarrollo de una aplicación a la medida decida emplear es funcional; no obstante se debe tomar en cuenta las consideraciones expuestas en los apartados 2.5 y 3.2 antes de decidir cual metodología utilizar.

Por último se concluye que vale la pena dedicar tiempo en la aplicación de alguna metodología, no únicamente en el tema de ingeniería de software, sino también en nuestra vida cotidiana como puede ser: el trabajo, situaciones personales que contengan, o no, algún riesgo, momentos decisivos, etc. Esto es, utilizar una metodología que nos ayude a la PLANEACIÓN de nuestras vidas y si a esto le añadimos la realización de escenarios como la herramienta de apoyo para la toma de decisiones es seguro el éxito de nuestras actividades.

## **FUENTES DE CONSULTA**

Análisis Estructurado Moderno

Yourdon Edward  
Prentice- Hall Hispanoamericana, S.A.  
México 1993

Análisis y Diseño Orientado a Objetos  
Con Aplicaciones

Booch Grady  
Segunda Edición  
Addison-Wesley / Diaz de Santos  
E.U.A. 1996

Análisis y Diseño de Sistemas de Información

Senn James A.  
Segunda Edición  
Mc Graw Hill  
México, D.F. 1992

El Lenguaje Unificado de Modelado

Booch Grady, Rumbaugh James, Jacobson Ivar  
Addison Wesley  
Madrid España 1999

Object-Oriented Modeling and Design

Rumbaugh, James, Michael Blaha  
Prentice Hall  
E.U.A 1991

UML y Patrones "Introducción al análisis y diseño orientado a objetos"

Larman G.  
Prentice-Hall Pearson  
México 1999

Procedimiento: Desarrollo de Aplicaciones a la Medida (DAM)

ISO 842: 1994/NMX-CC-001  
México 1995

Sistema de Calidad- Modelo para el aseguramiento de la calidad en diseño, desarrollo, producción,  
instalación y servicio

NMX-CC-003: 1995 IMNC  
ISO 9001:1994

## CONCLUSIONES GENERALES

El desarrollo de un Sistema de Información especializado, aquí llamado Aplicación a la Medida, se construye con el manejo de determinadas etapas, que a su vez contiene varios pasos a seguir; desde que aparece la necesidad de la automatización de un proceso hasta su implantación, es decir hasta su puesta en producción.

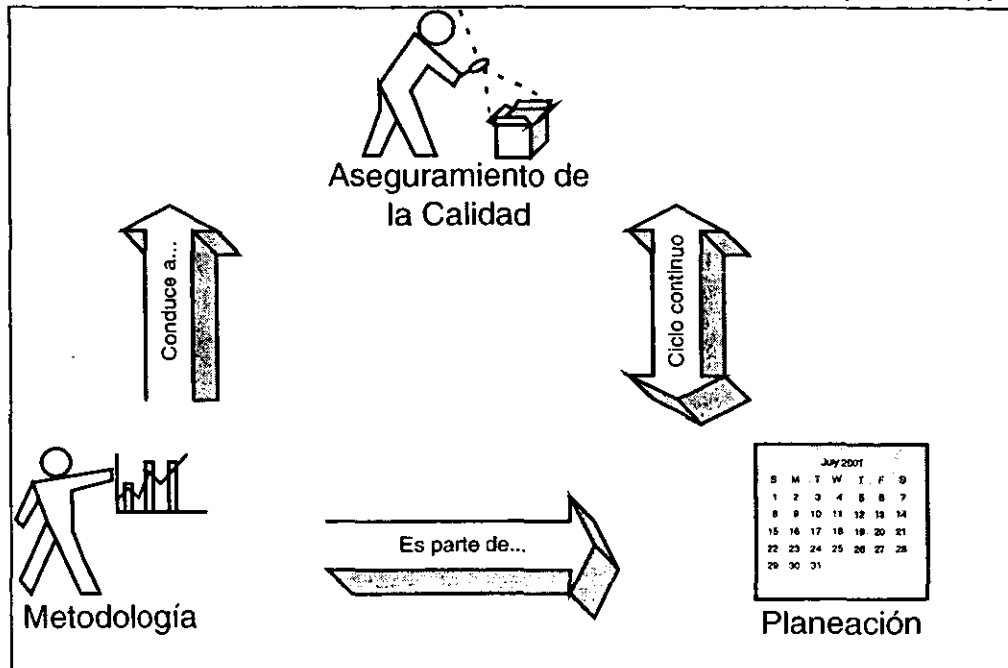
A pesar de la constante recomendación para hacer uso de alguna metodología de análisis y diseño de sistemas de información, la industria de la computación no se ha aplicado adecuadamente a la utilización de cualquiera de ellas, generando en la mayoría de los casos productos de baja calidad lo que da como consecuencias pérdidas económicas importantes, tanto para el cliente como para el proveedor responsables del sistema informático a realizar, así como también desgastes físicos de las personas involucradas en el desarrollo de programas; de lo cual la empresa consultora a la que pertenece el DAM, no se ha librado; la finalidad del presente estudio es el disolver esta problemática.

Se determinó que uno de los pasos esenciales que tiene la planeación es el *Diagnóstico* del problema en estudio, de ahí la importancia de seguir una metodología de análisis para el desarrollo de un sistema informático, como se observa en el apartado 1.5.3. También se confirma que el uso adecuado de alguna metodología de análisis es aplicable a varios tipos de problemas, no exclusivamente computacionales, ayudándonos a entender cualquier situación a la que nos enfrentemos. Un claro ejemplo es el obtenido en el capítulo I apartado 1.11 de este trabajo, en el que se describe el análisis de cómo se desarrollaría este estudio y su relación con las diferentes entidades involucradas; el cual se llevó a cabo de manera exitosa en los capítulos II y III.

En el capítulo I se expuso la clara relación que tienen los conceptos de *Metodología, Planeación y Calidad*. Verificando que; la aplicación de una *Metodología* para el desarrollo de sistemas es una herramienta útil para la *Planeación*; la correcta aplicación de una *Metodología de Análisis* reduce el porcentaje de falla de un sistema, lo que genera *Calidad* de al menos un 80% en el producto obtenido. Por lo tanto si en la *Planeación* de un empresa o negocio se encuentra la búsqueda de *Calidad* en sus sistemas de información entonces se debe concientizar el uso de una *Metodología* formal.

El obtener información de varios autores y bibliografía en el tema de metodologías de análisis y diseño, permitió obtener un trabajo especializado, determinando el uso adecuado de los pasos requeridos en el desarrollo de análisis de sistemas, definiéndolo en los apartados 2.4.1 y 2.4.2. Se obtuvo las definiciones específicas, en un lenguaje y gráficas entendibles para el analista; y las indicaciones de cómo son construidos cada uno de estos pasos. Se confirmó la facilidad y rapidez de aplicación con lo obtenido en el desarrollo del caso práctico expuesto en el capítulo III apartados 3.1.2, 3.1.3 de éste trabajo; esto último también representa los ejemplos que se utilizarán para una adecuada implementación de la etapa de análisis. Además se obtuvo las diferencias y características que implica el utilizar una u otra de las metodologías estudiadas, lo que ayudará a decidir oportunamente cual de ellas conviene utilizar, resultado expuesto en los apartados 2.5, 3.2, 3.2.1 y 3.2.2. Concluyendo con esto último el desarrollo de la solución planteada, determinando los cambios adecuados al DAM, en el apartado 3.3. Así también se termina un ciclo de lo expuesto en el apartado 1.9 de Mejora continua en la implantación de ISO 9001.

**Relación observada entre Metodología, Aseguramiento de la Calidad y Planeación**



Finalmente se concluye que los resultados obtenidos durante el desarrollo de éste estudio son: se eliminó el tiempo invertido en entender el procedimiento en un 85% al no tener que conseguir, consultar y repasar bibliografía sobre como realizar las actividades contenidas en las metodologías estructurada y UML; no existe el error en la elección adecuada de una metodología de análisis; la ejecución de la etapa de análisis se agilizó en un 90%, ya que hay un pronto entendimiento de las actividades a realizar además de poder contar con los ejemplos desarrollados; disminuyó la generación de errores en la aplicación de las actividades de análisis hasta en un 90%, pues la explicación de como se realizan las actividades de análisis hacen que su entendimiento sea confiable por lo tanto tiende a la no generación de errores; entonces se tiene que la disminución en: el tiempo invertido para entender el procedimiento, la ejecución de la etapa de análisis y la generación de errores hacen que las fechas estimadas para cubrir el análisis de un sistema de información se cumplan, lo que da una reducción de los atrasos de fechas comprometidas aquejadas a la etapa de análisis hasta en un 85%; y por último se obtuvo un análisis completo y actualizado de acuerdo a las metodologías estudiadas. Cumpliendo la finalidad del presente trabajo el: Especificar y fortalecer la etapa de análisis contenida actualmente en el procedimiento DAM.

## APENDICE

**Amigable.** Una aplicación es amigable si el usuario humano lo encuentra fácil de usar.

**Albedrío.** Obrar por reflexión y elección. Voluntad de tomar una decisión en vez de otra.

**Análisis.** Analizar un módulo significa inspeccionarlo para entender sus propiedades y capacidades.

**Aplicaciones a la Medida.** Sistemas de información creados para resolver problemas específicos.

**A tiempo.** Es una cualidad del proceso, que se refiere a poder liberar el producto en el tiempo establecido.

**Atributo.** Característica o propiedad de tipo con un nombre asignado.

**Booleano.** Enumeración cuyos valores son verdadero y falso.

**Cadena.** Secuencia de caracteres de texto.

**Confiabilidad de una aplicación.** Es exacto si el usuario puede depender de él.

**Contractual.** Precedente o derivado del contrato.

**Convergencia.** Cuando se dirigen a un mismo punto.

**Diagrama.** Representación gráfica de un conjunto de elemento, representado la mayoría de las veces como un grafo conexo de nodos (elementos) y arcos (relaciones)

**Eficiencia.** Un programa es eficiente si usa sus recursos adecuadamente.

**Entendible.** Si es fácil de entender por otras personas.

**Estructura de datos.** Definición abstracta que representa un tipo de dato, que contendrá un valor o varios valores, que son requeridos por un sistema de información.

**Evolucionable.** Si se puede modificar obteniendo una nueva versión.

**Herramientas.** Sirven de apoyo para la aplicación de técnicas, métodos y metodologías.

**Implantación.** Establecimiento, instauración o puesta en ejecución.

**Implementación.** Instalación y puesta en marcha, en un ordenador, de un sistema.

**Mantenible.** Una aplicación es mantenible si es fácil hacerle modificaciones.

**Método.** Guía general que gobierna la ejecución de una actividad. Son rigurosos, sistemáticos y disciplinados. Es la forma de hacer algo.

**Predecir.** Anunciar por suposición, adivinar el futuro.

**Producto.** Es el resultado de actividades o procesos. Un producto puede ser tangible (ejemplo: ensambles o materiales procesados) o intangible (ejemplo: información o conceptos) o una combinación de los mismos.

**Reparable.** Si permite la corrección de sus defectos con poco trabajo.

**Reusable.** Si se pueden usar partes o todo con cambios menores para la construcción de un nuevo programa.

**Robustez.** Un Software es robusto si se comporta "razonablemente", aún en circunstancias no anticipadas en la especificación de requerimientos.

**Sintaxis.** Modo en que deben ordenarse o enlazarse las palabras (instrucciones, controles, variables, constantes, etc.) para completar una o varias instrucciones en un programa de computo.

**Sistema.** Conjunto de partes coordinadas para lograr un conjunto de metas.

**Software.** Es la suma total de programas de computadora, procedimientos, reglas, documentación asociada y datos necesarios para la operación de un sistema computarizado.

**Técnica.** Son más mecánicas que los métodos y tienen una aplicación más restringida. Ambos se empaquetan en una Metodología.

**Tipo de Dato.** Tipo cuyos valores no tienen identidad. Los tipos de datos incluyen tipos primitivos predefinidos (tales como números y cadenas de caracteres), sí como tipos enumerados (tales como los booleanos).



## FUENTES DE CONSULTA

Administración Teoría y Práctica

Robbins Stephen P.  
Prentice- Hall Hispanoamericana, S.A.  
México

Análisis Estructurado Moderno

Yourdon Edward  
Prentice- Hall Hispanoamericana, S.A.  
México 1993

Análisis y Diseño Orientado a Objetos  
Con Aplicaciones

Booch Grady  
Segunda Edición  
Addison-Wesley / Diaz de Santos  
E.U.A. 1996

Análisis y Diseño de Sistemas de Información

Senn James A.  
Segunda Edición  
Mc Graw Hill  
México, D.F. 1992

Análisis y Diseño de Sistemas de Información

Senn, James A.  
1a. Edición  
Mc Graw-Hill  
México 1990

C++ and the OOP Paradigm

Rao Bindu R.  
Cap Gemini America Series  
Mc. Graw Hill  
E.U.A. 1993

Diseño de Bases de Datos con UML

OsBorne  
Mc Graw Hill  
Edición Oracle Press

El Enfoque de Sistemas

Churchman, West C.  
Diana  
México 1993

El Lenguaje Unificado de Modelado

Booch Grady, Rumbaugh James, Jacobson Ivar  
Addison Wesley  
Madrid España 1999

El proceso de la Investigación Científica  
Fundamentos de Investigación

Tamayo y Tamayo Mario  
Segunda Edición  
Limusa  
1989

Escenarios, El arte de prevenir el futuro

Van Der Heijden Kess  
2da. Reimpresión  
Panorama  
México 1998

Introducción a la Computación y a la Programación Estructurada

Lenne G. Guillermo  
Segunda Edición  
MC. Graw-Hill  
México, D.F. 1990

Ingeniería de Software

Segunda Edición  
Sommerville Ian  
Addison Wesley Iberoamericana  
México D.F. 1988

Metodología de la Programación

Joyanes, Aguilar Luis  
1a. Edición  
Mc. Graw-Hill  
México 1990

Notas de clase: Ingeniería de Software

Ibargüengoitia, González Ma. Guadalupe  
Maestría en Ciencias de la Computación  
Facultad de Ciencias, UNAM

Object-Oriented Modeling and Design

Rumbaugh, James, Michael Blaha  
Prentice Hall  
E.U.A 1991

Técnicas de Programación

Alonso, Amo F.  
Morales, Lozano A.  
1a. Edición  
Paraninfo  
Madrid 1988