



UNIVERSIDAD VERACRUZANA

INSTITUTO DE INGENIERIA

**PARADIGMA DE INGENIERIA DE
SOFTWARE BASADO EN LA
TEORIA GENERAL DE SISTEMAS**

T E S I S

PARA OBTENER EL GRADO DE

MAESTRIA EN CIENCIAS DE LA COMPUTACION

P R E S E N T A

CARLOS LEY BORRAZ

H. VERACRUZ, VER.

1994



UNIVERSIDAD VERACRUZANA INSTITUTO DE INGENIERIA

SEP-FOMES
933109

H. Veracruz, Ver., a 25 de Noviembre de 1994
SI570/94

Al Candidato al Grado señor: ING. CARLOS LEY BORRAZ
P R E S E N T E .

En atención a su solicitud relativa, me es grato transcribir a Usted a continuación el tema que aprobado por esta Dirección propuso el M.C. Miguel A. Fernández Marin, para que lo desarrolle como tesis, para obtener el Grado de Maestro en Ciencias de la Computación.

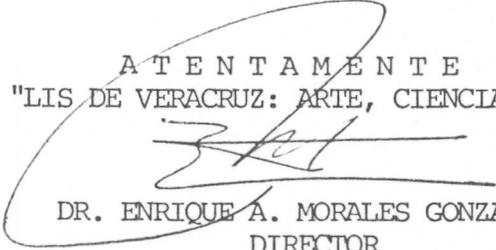
TEMA :

" PARADIGMA DE INGENIERIA DE SOFTWARE BASADO EN LA TEORIA GENERAL DE SISTEMAS "

- 1.- Introducción
 - 2.- Objetivo
 - 3.- Justificación
 - 4.- Estado del Arte de la Ingeniería de Software
 - 5.- Teoria General de Sistemas
 - 6.- Paradigma de Ingeniería de Software basado en la Teoria General de Sistemas
 - 7.- Conclusiones
- Bibliografía
Glosario

Sin otro particular, me es grato reiterarle la seguridad de mi más atenta y distinguida consideración.

ATENTAMENTE
"LIS DE VERACRUZ: ARTE, CIENCIA, LUZ"


DR. ENRIQUE A. MORALES GONZALEZ
DIRECTOR

ccp. Interesado
ccp. Archivo
ccp. Minutario
EMG/apm*.

Proyecto de Digitalización de Tesis
Responsable M.B. Alberto Pedro Lorandi Medina
Colaboradores: Estanislao Ferman García
M.B. Enrique Rodríguez Magaña

UNIVERSIDAD VERACRUZANA

INSTITUTO DE INGENIERIA

PARADIGMA DE INGENIERIA DE
SOFTWARE BASADO EN LA
TEORIA GENERAL DE SISTEMAS

T E S I S

PARA OBTENER EL GRADO DE
MAESTRIA EN CIENCIAS DE LA COMPUTACION

P R E S E N T A

CARLOS LEY BORRAZ

DIRECTOR DE TESIS: M.C. MIGUEL A. FERNANDEZ MARIN

H. VERACRUZ, VER.

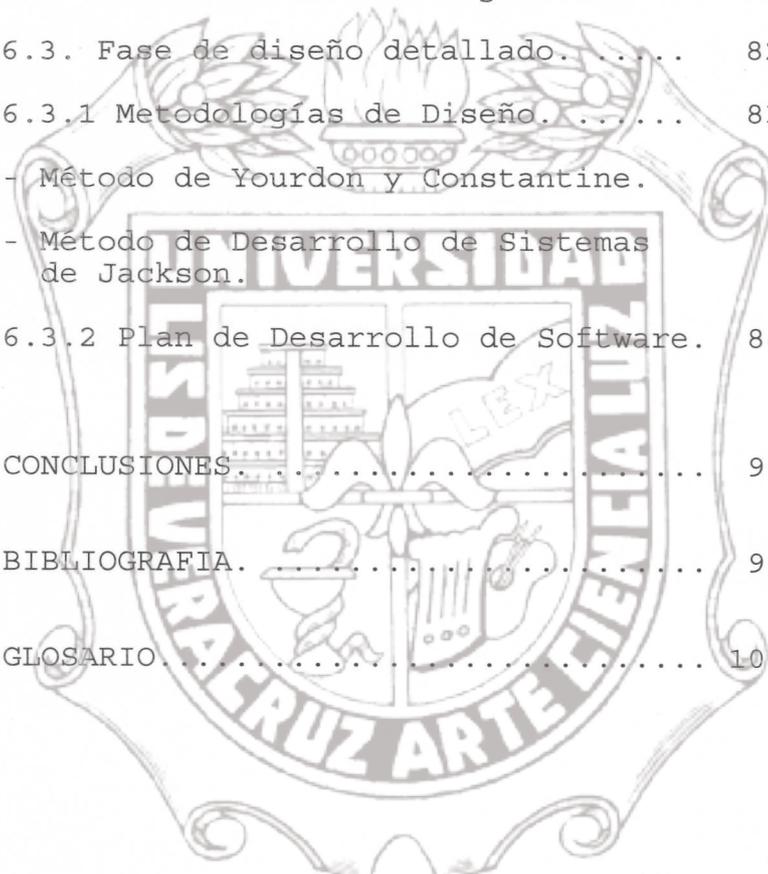
1994

I N D I C E

I.	INTRODUCCION.	1
II.	OBJETIVO.	3
III.	JUSTIFICACION.	4
IV.	ESTADO DEL ARTE DE LA INGENIERIA DE SOFTWARE	7
	4.1 Ingeniería de Software.	7
	4.2 Paradigma de cascada.	8
	4.3 Paradigma de prototipo.	12
	4.4 Paradigma de espiral.	18
V.	TEORIA GENERAL DE SISTEMAS.	22
	5.1 Conceptos básicos de la Teoría General de sistemas.	23
VI.	PARADIGMA DE INGENIERIA DE SOFTWARE BASADO EN LA TEORIA GENERAL DE SISTEMAS.	29
	6.1. Fase de diagnóstico.	33
	6.1.1 La situación problemática no estructurada.	35
	6.1.1.1 Confrontación con la situación problemática.	36
	6.1.1.2 Ubicación de la situación problemática.	39

6.1.2 La situación problemática analizada.	42
6.1.3 Definición esencial de los sistemas relevantes.....	45
6.1.3.1 Definir los sistemas relevantes a la situación problemática	45
6.1.3.2 Formulación de las definiciones esenciales de los sistemas relevantes.....	46
6.1.4 Conformación y prueba de los modelos conceptuales.....	51
6.1.4.1 Análisis funcional.....	51
6.1.4.2 Evaluación por propiedades de los sistemas.....	56
6.1.5 Comparación de la situación problemática con los modelos conceptuales.....	60
6.1.6 Definición de cambios deseables y posibles.....	62
6.2. Fase de diseño conceptual.....	63
6.2.1 Formulación de la definición esencial de la alternativa de solución.....	69
6.2.2 Derivación del modelo conceptual.....	70
6.2.3 Derivación de las categorías de información.....	70
6.2.3.1 Identificación de la información.....	71
- Información requerida	
- Información generada	
- Información de control	
6.2.3.2 Definición de las categorías de información.....	74

	6.2.4 Análisis de flujos de información actividad-actividad.	77
	6.2.5 Definición de procedimientos de procesamiento de información. ..	79
	6.2.6 Análisis de los procesamientos de información.	79
	6.2.7 Planteamiento de algoritmos. .	82
	6.3. Fase de diseño detallado.	82
	6.3.1 Metodologías de Diseño.	83
	- Método de Yourdon y Constantine.	
	- Método de Desarrollo de Sistemas de Jackson.	
	6.3.2 Plan de Desarrollo de Software.	88
VII.	CONCLUSIONES.	96
VIII.	BIBLIOGRAFIA.	98
IX.	GLOSARIO.	100



LISTA DE FIGURAS

FIGURA		Pag.
1	El Modelo de Cascada	9
2	Modelo de Cascada Modificado	11
3	Modelo de Prototipos	13
4	Modelo de Ciclo de Vida	15
5	Modelo de Versiones Suscesivas	17
6	Modelo de Espiral	19
7	Diagrama General del Paradigma Propuesto	30
8	Fase de Diagnóstico	31
9	Fase de Diseño Conceptual	32
10	Diagrama Sistémico causa efecto	41
11	Mapa de Ambientes	44
12	Diseño de la Definición Escencial	49
13	Modelo Conceptual	55
14	Propiedades de los Sistemas	58
15	Estructura de la Cruz Maltesa	66
16	Modelo de Datos Jerárquico	76
17	Cruz Maltesa (Parte superior)	78
18	Cruz Maltesa (Completa)	81
19	Modelo Propuesto	95

I. INTRODUCCION

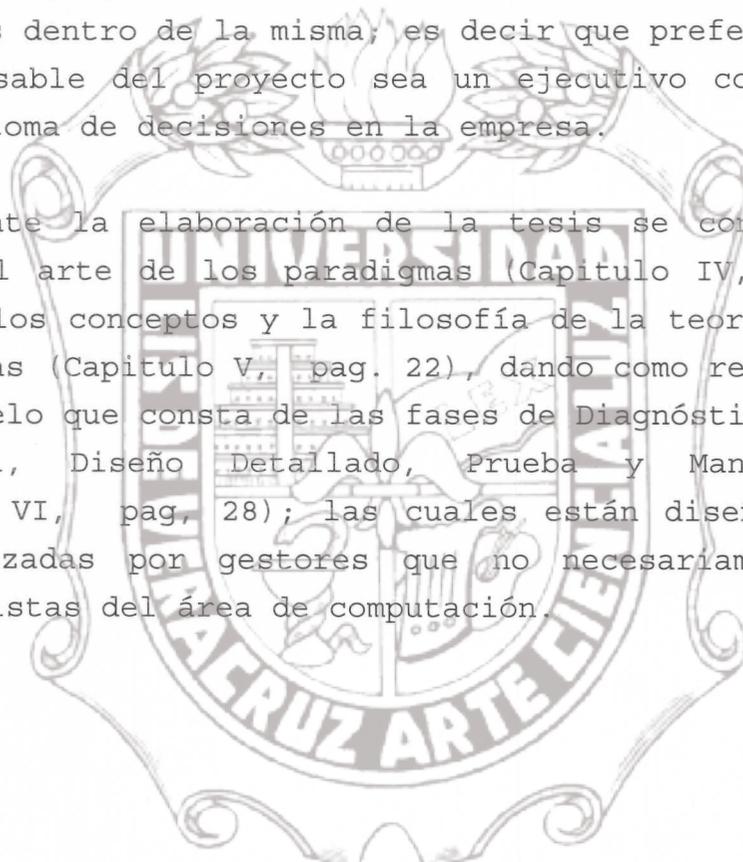
La Ingeniería de Software tiene como propósito el desarrollar productos de software de calidad, optimizando los recursos que intervienen en su elaboración e implementación. Su importancia en el área de computación y su impacto en las empresas es cada día mayor.

Entre los paradigmas existentes se encuentran los de Cascada, Desarrollo Incremental, Prototipos, Espiral, el recomendado por el Institute of Electrical and Electronic Engineering (IEEE), y algunas combinaciones entre ellos. Cada uno de ellos persigue el mismo objetivo y utiliza un enfoque dirigido a profesionistas con una gran experiencia en el área; éste enfoque se contrapone a la tendencia actual de que los gestores de proyectos de desarrollo de software sean profesionistas de diversas disciplinas, que han incursionado en el área de computación como parte de su formación o actualización profesional, contando con una visión global de la empresa y una gran responsabilidad por la importancia de sus decisiones.

En el presente trabajo se plantea un nuevo paradigma de ingeniería de software, basado en la teoría general de sistemas. Este posee características propias como: el utilizar las propiedades de los sistemas, el ser compatible con las técnicas de desarrollo que utilizan otros paradigmas, tratando de ser un modelo sencillo en sus conceptos y ágil en el desarrollo de cada una de las fases que lo componen.

El diseño del nuevo paradigma tiene la facilidad de poder ser utilizado a diferentes niveles de detalle y principalmente por gestores que tengan una visión global de la empresa y que cuenten con un alto nivel de toma de decisiones dentro de la misma; es decir que preferentemente el responsable del proyecto sea un ejecutivo con un alto poder de toma de decisiones en la empresa.

Durante la elaboración de la tesis se considera el estado del arte de los paradigmas (Capítulo IV, pag. 7), así como los conceptos y la filosofía de la teoría general de sistemas (Capítulo V, pag. 22), dando como resultado un nuevo modelo que consta de las fases de Diagnóstico, Diseño Conceptual, Diseño Detallado, Prueba y Mantenimiento (Capítulo VI, pag. 28); las cuales están diseñadas para ser realizadas por gestores que no necesariamente sean profesionistas del área de computación.



II. OBJETIVO

La presente tesis de grado pone de manifiesto su propósito desde el propio título de la misma " Paradigma de Ingeniería de Software basado en la Teoría general de Sistemas ", mas sin embargo es importante que quede asentado el objetivo central que persigue, el cual puede sintetizarse de la siguiente manera :

Diseñar un modelo de desarrollo de productos de software orientado a usuarios de nivel ejecutivo, utilizando los principios del enfoque de la teoría general de sistemas.

Teniendo el propósito de poder contar con una alternativa que contenga características propias que propicien el correcto desarrollo del software.

III. JUSTIFICACION

La Ingeniería de Software ha evolucionando tanto en su propia definición como en la creación de nuevos paradigmas, los cuales se plantean como una propuesta de solución a los problemas que se presentan durante el desarrollo de productos de software. Entre los modelos de desarrollo que actualmente existen se encuentran los de: Cascada, Desarrollo Incremental, Prototipos, Espiral y algunos otros propuestos por la IEEE. Todos ellos presentan ventajas y desventajas, pero en general, pretenden alcanzar el mismo objetivo, aún cuando estén compuestos de diferentes fases ó etapas. A continuación se describen algunas de esas situaciones:

- a) Es común observar que a medida que el sistema se desarrolla, se vuelve más grande y complejo y las técnicas utilizadas en los modelos son también más difíciles de aplicar. Hasta llegar a un nivel inoperante, ocasionando con ello que surja la necesidad de utilizar varios modelos de desarrollo para poder continuar con el sistema.
- b) Cada vez es más frecuente que ejecutivos de alto nivel y/o profesionistas de diversas áreas, sean responsables de proyectos de software, los cuales enfrentan una serie de dificultades en la utilización de los paradigmas de software, principalmente por la falta de experiencia o conocimiento en el área de desarrollo de productos de software.

- c) En la mayoría de las empresas que cuentan con centros o departamentos de desarrollo de software, existe la tendencia a tener preferencia por diseñar su propio modelo de desarrollo o bien combinar algunos paradigmas existentes.
- d) Generalmente la información que sustenta la definición del problema no llega a formar parte del proyecto de software, ocasionando que se dupliquen esfuerzos principalmente en la fase de planeación, y en otros casos, simplemente no se utiliza.

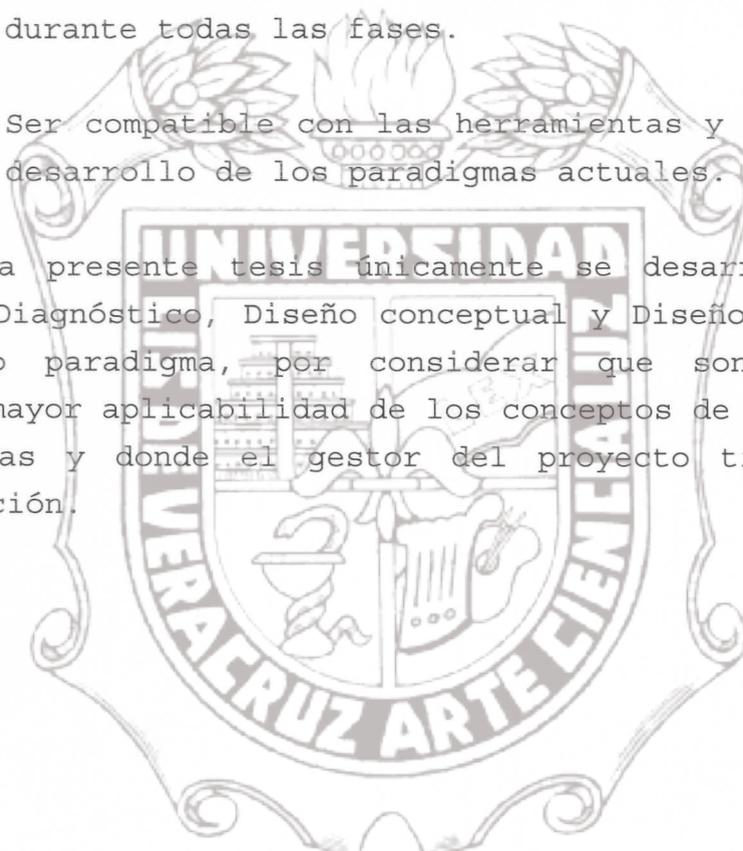
En esta tesis se aborda este problema diseñando un nuevo paradigma de ingeniería de software basado en el enfoque de ingeniería de sistemas, que permita utilizar las herramientas que se han desarrollado para otros modelos de ingeniería de software.

El propósito de iniciar el desarrollo de un nuevo modelo basado en la Teoría general de Sistemas es el poder contar con una alternativa que contenga características no consideradas hasta la fecha, las cuales se mencionan a continuación:

- Aplicar un enfoque holístico para el desarrollo de software.
- Orientado a usuarios de nivel corporativo.
- Partir desde la confrontación del problema hasta el mantenimiento del sistema de software.

- Mantener la relación entre el sistema de software y el ambiente del cual forma parte.
- Mantener un control del nivel de complejidad durante todas las fases.
- Ser compatible con las herramientas y métodos de desarrollo de los paradigmas actuales.

En la presente tesis únicamente se desarrollan las fases de Diagnóstico, Diseño conceptual y Diseño detallado del nuevo paradigma, por considerar que son las que permiten mayor aplicabilidad de los conceptos de ingeniería de sistemas y donde el gestor del proyecto tiene mayor participación.



IV. ESTADO DEL ARTE DE LA INGENIERIA DE SOFTWARE

4.1 Ingeniería de Software

El término Ingeniería de Software se introdujo por primera vez a finales de la década de los 60's, cuando se trató de abatir la crisis en la construcción de grandes sistemas de software. Actualmente existen muchas definiciones de ingeniería de software donde se han agregado conceptos de calidad y eficiencia, pero en esencia, todas se refieren de una u otra forma a la ingeniería de software como un enfoque sistemático para el desarrollo, operación y mantenimiento de software.

La Ingeniería de Software se integra por un conjunto de tres elementos: métodos, herramientas y procedimientos, que facilitan el proceso del desarrollo de software y suministran las bases para construir software de calidad de una forma productiva; lo cual se lleva a cabo mediante pasos o fases que en su conjunto se le denomina paradigmas de ingeniería de software y en si, conforman un modelo de desarrollo de productos de software.

Los métodos de la ingeniería de software describen el cómo construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño y estructura de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento.

Los procedimientos de la ingeniería del software definen la secuencia en la que se requieren los métodos, la documentación que se requiere, los controles que ayuden a asegurar la calidad y coordinar los cambios, y las guías que faciliten a los gestores de software establecer su desarrollo.

A continuación se describen algunos de los paradigmas mas utilizados.

4.2 Paradigma de Cascada.

El modelo de cascada se basa en que el desarrollo de software es un proceso lineal descendente de varios pasos, con posibles iteraciones entre cada uno de ellos, donde además se realiza validación o verificación durante su desarrollo.

La primera versión de este modelo es publicado en 1970 por Boehm en un artículo y posteriormente en su libro "Software Engineering Economics" en el cual propone un esquema del ciclo de vida del desarrollo de software, como lo muestra la figura 1.

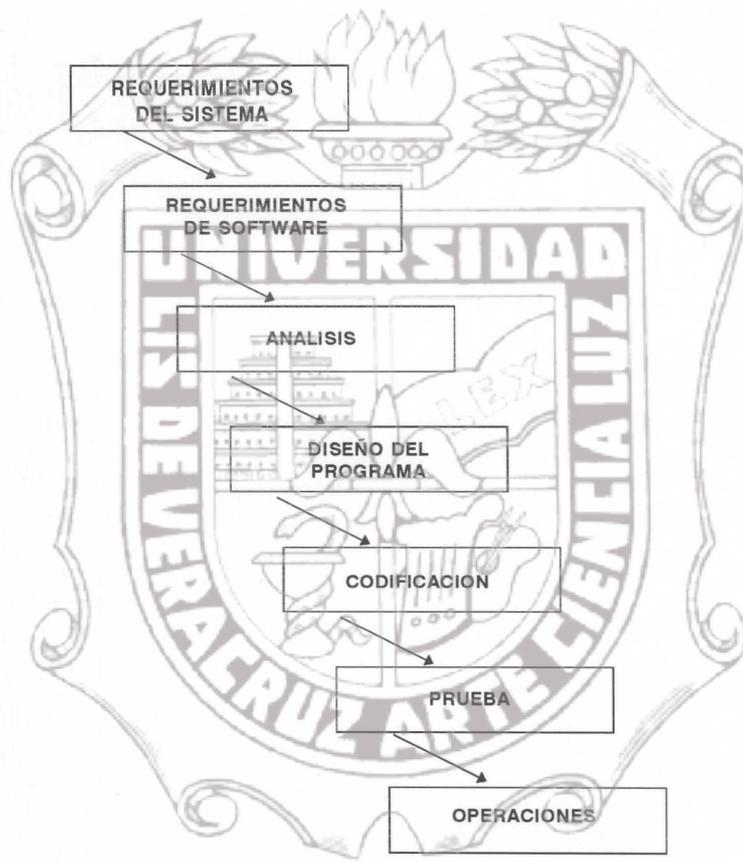


Figura 1. El modelo de cascada

Este primer esquema tuvo muchas objeciones antes de ser adoptado debido a sus limitaciones, que van desde no permitir la retroalimentación entre actividades continuas, hasta las concernientes al control de calidad y seguridad. Esto provocó que en 1975 Boehm realizara una variante al modelo, como se muestra en la figura 2, la cual incluye los conceptos de verificación y validación. A partir de entonces ha servido como base de otros modelos, por lo que actualmente se le conoce como el modelo clásico de vida del software.



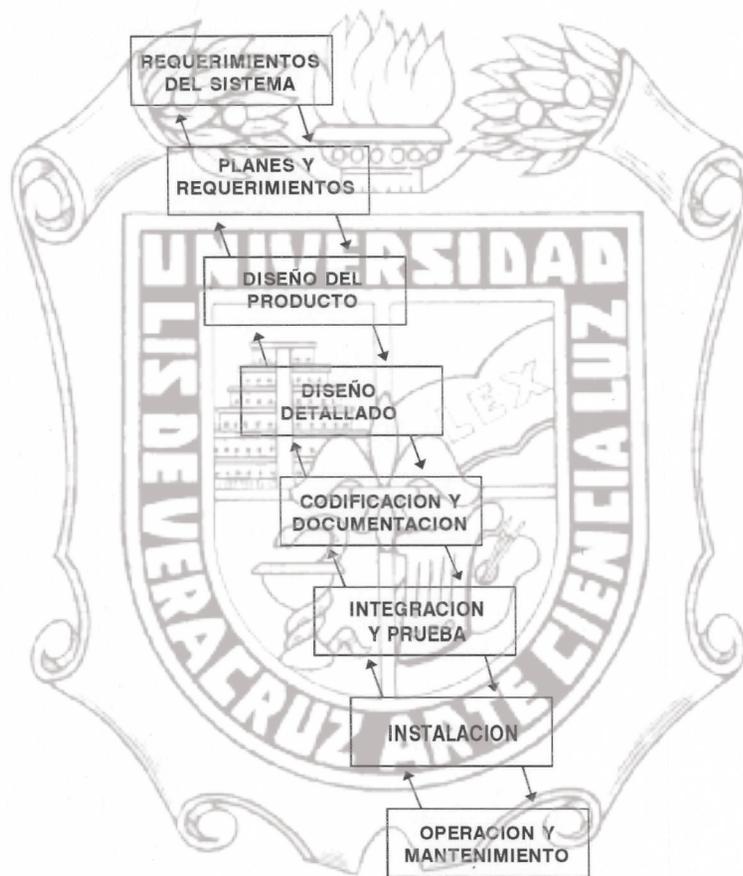


Figura 2. Modelo de Cascada Modificado

Limitantes :

Algunos de los inconvenientes de este modelo son el que los proyectos no necesariamente siguen el flujo secuencial propuesto, además de tener que establecer los requerimientos del sistema al inicio por única vez, no permite la corrección o modificación de posibles fallas que repercutirán durante la producción del software. La limitante más fuerte es que las iteraciones están restringidas a pasos continuos. En la práctica se requiere retroceder más de un paso en algunos casos. El desarrollo de prototipos se puede intentar, pero no está expresamente permitido, a fin de lograr una completa especificación adecuada para el diseño.

4.3 Paradigma de Prototipos.

La versión mas sencilla del modelo de prototipos se ilustra en la figura 3, el propósito de todo modelo basado en prototipos, es contar con una representación del software previa al producto final, con el propósito de mejorar o corregir fallas con antelación.

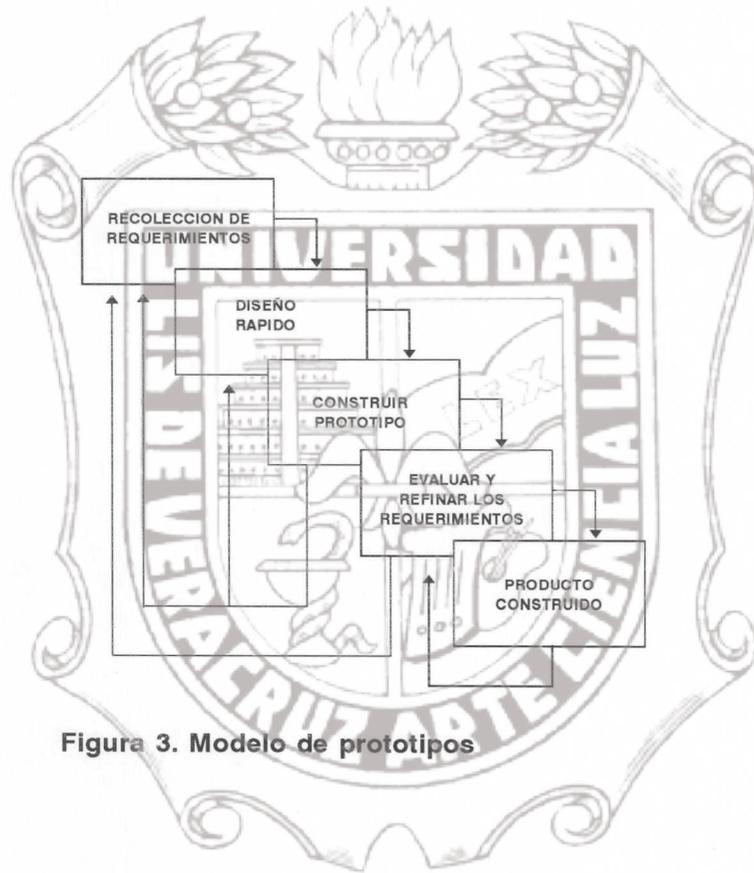


Figura 3. Modelo de prototipos

En 1985 los autores De Marco y Buxton incorporaron el desarrollo de prototipos al modelo de cascada, originando el modelo mostrado en la figura 4, mismo que posteriormente es modificado con el propósito de hacer la distinción entre el prototipo que se le presenta al cliente y el que es utilizado para la implantación .



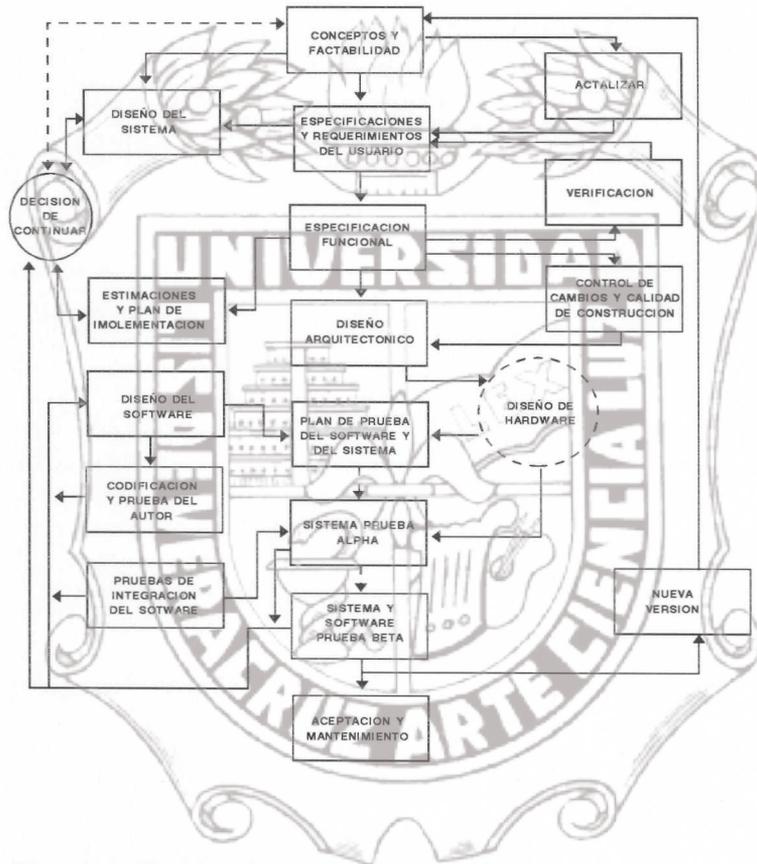


Figura 4. Modelo del ciclo de vida

Limitantes :

Algunos de los inconvenientes de usar prototipos en un modelo de desarrollo de software, son el riesgo de que el prototipo se tome como una primera versión del producto y se trasladen premisas o algoritmos que se utilizaron en la elaboración del prototipo al producto final sin un adecuado análisis, sin perder de vista que se puede cometer el error de que una vez que el prototipo cumpla con las especificaciones planteadas, el cliente quiera incrementar sus expectativas.

Una variante del paradigma de prototipos es la de versiones sucesivas, la cual consiste en refinar una versión hasta alcanzar el producto deseado. Un esquema básico de este modelo se muestra en la figura 5, donde se puede observar que es un proceso interactivo.



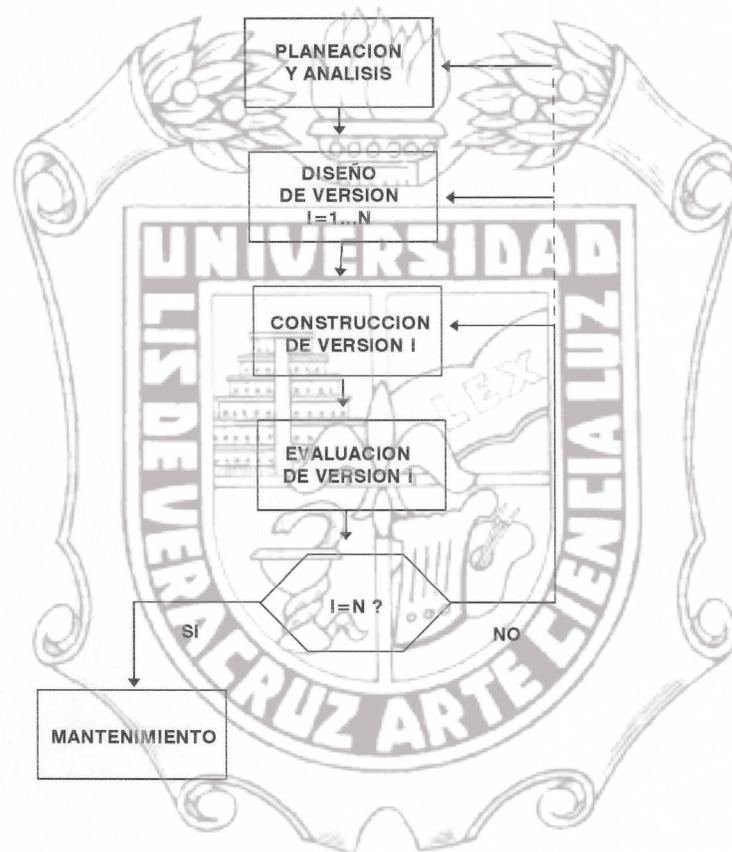


Figura 5. Modelo de versiones sucesivas

4.4 Paradigma de Espiral.

La característica principal del modelo de espiral es que su rango de opciones organiza las características deseables de los paradigmas, además de manejar el concepto de riesgo como un factor decisivo en el desarrollo. El modelo de espiral del proceso de software es la evolución de varios refinamientos al modelo de cascada, la combinación de varios paradigmas y la inclusión de la experiencia de expertos para el análisis de riesgos en el desarrollo.

En la figura 6, cada ciclo de la espiral se inicia con la identificación de: los objetivos de ese fragmento del producto a ser elaborado, los medios alternos para implementar esa porción del producto y las restricciones impuestas en la aplicación de alternativas. El siguiente paso consiste en evaluar las alternativas relativas a los objetivos y restricciones, en cuyo proceso se identifican áreas de incertidumbre que son fuentes importantes en el proyecto. Por lo que a continuación se debe formular una estrategia de costo-efectividad para resolver las fuentes de riesgo utilizando la técnica mas apropiada.

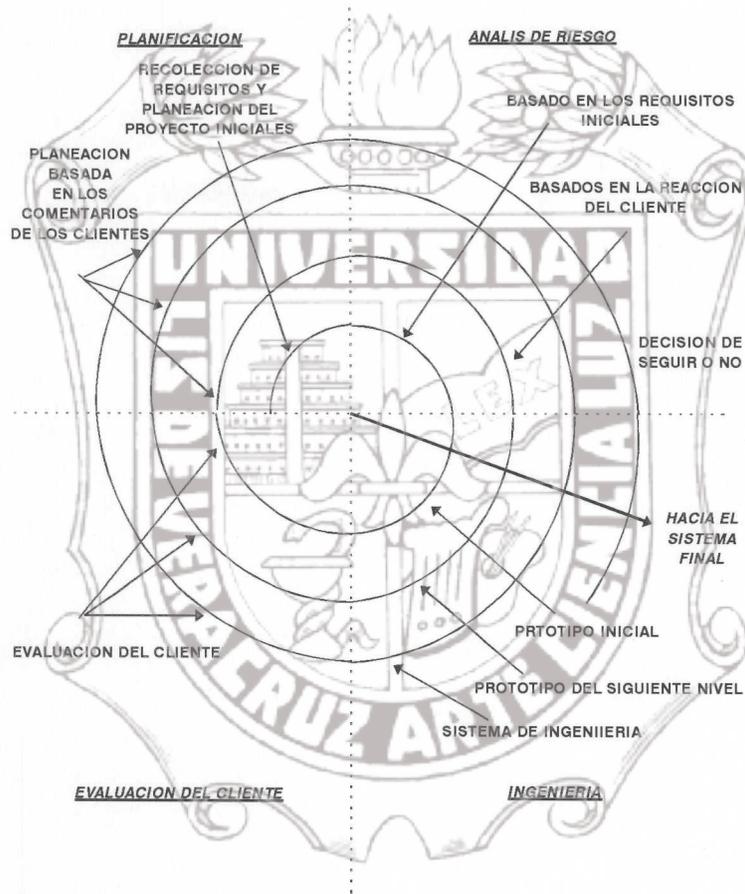


Figura 6. Modelo de espiral

El modelo de espiral se aplica tanto al esfuerzo de desarrollo como al esfuerzo para la extensión de software. Además, si el proceso de espiral falla en alguna prueba, entonces se termina el espiral. De otra forma, termina con la instalación del software desarrollado.

Limitantes :

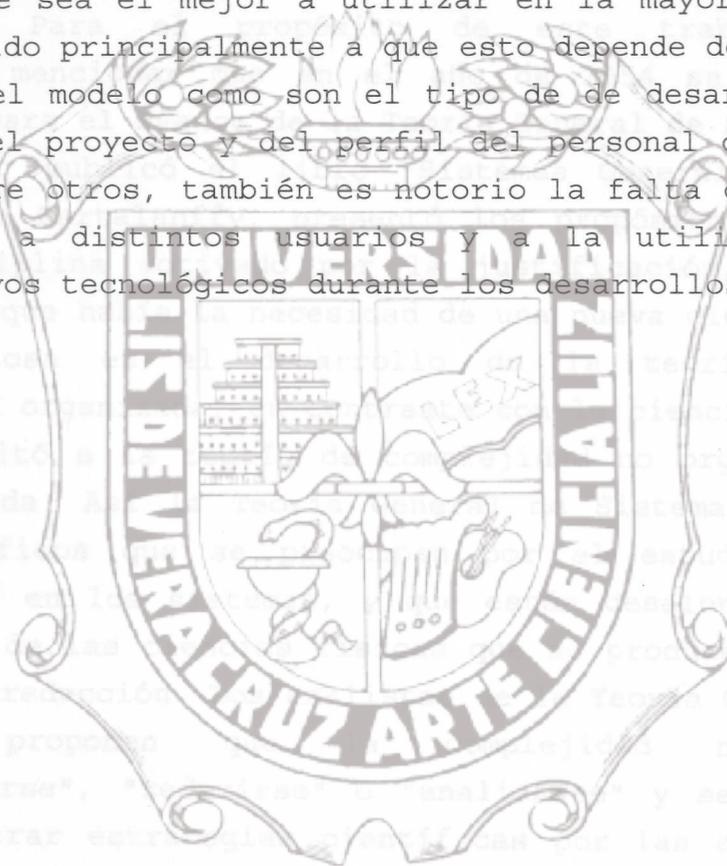
Los inconvenientes de utilizar este modelo son que el modelo de espiral tiene dificultades de aplicación cuando se realiza para trabajos que son muy rígidos en sus restricciones y limitaciones. El manejar y determinar riesgos dentro del modelo puede ser su mayor ventaja, pero de no hacerlo correctamente es contraproducente, debido a que principalmente utiliza juicios de personal experto, por lo que se requiere de elaboración adicional en los pasos del modelo, para asegurar que todos los participantes estén operando en un contexto consistente, así como guías y listas de chequeo para identificar los puntos más probables de riesgo.

Conclusión :

En términos generales se puede decir que los tres paradigmas anteriores y sus combinaciones son el universo actual de los modelos formales de desarrollo de software, los cuales tienen una orientación a solucionar problemas operativos y no los de tipo estratégico; así mismo el buen desempeño del modelo es sensible a la experiencia y conocimiento que tenga el analista en el área de computación en este tipo de proyectos, finalmente se observa que la evolución de los modelos de desarrollo de ingeniería de software han sido basados en la experiencia así como tratar de responder a los nuevos conceptos

(calidad, verificación, validación, mantenimiento, etc.) que ha incorporado la Ingeniería de Software.

Con base a lo anterior se puede resumir que no existe un modelo que cubra todas las características de los demás o que sea el mejor a utilizar en la mayoría de los casos, debido principalmente a que esto depende de factores externos del modelo como son el tipo de desarrollo, la magnitud del proyecto y del perfil del personal con que se cuenta entre otros, también es notorio la falta de modelos orientados a distintos usuarios y a la utilización de nuevos apoyos tecnológicos durante los desarrollos.



Actualmente La Teoría General de sistemas (TGS) llamada también Enfoque de Sistemas es una disciplina que tiene como ideas principales:

- El Todo es más que la suma de sus partes.
- El Todo determina la naturaleza de las partes.
- Las partes no pueden comprenderse si se consideran en forma aislada del Todo.

V. TEORIA GENERAL DE SISTEMAS

Los orígenes de la Teoría General de Sistemas puede remontarse probablemente, a los orígenes de la ciencia y la filosofía. Para el propósito de este trabajo será suficiente mencionar que en el año de 1954 se fundó la "Sociedad para el avance de la Teoría General de Sistemas", y en 1956 publicó el libro "Sistemas Generales" donde Ludwig Von Bertalanffy, presentó los propósitos de esta nueva disciplina motivado por la justificación - entre otras-, de que había la necesidad de una nueva ciencia, que fuera exitosa en el desarrollo de la teoría de la complejidad organizada, en contraste con la ciencia clásica que se limitó a la teoría de complejidad no organizada o desorganizada. Así la Teoría General de Sistemas reúne a los científicos que se preocupan por el estudio de la complejidad en los sistemas, y que están desalentados con el enfoque de las ciencias físicas que se produce mediante análisis y reducción. Los analistas de la Teoría General de Sistemas proponen que la complejidad no puede "simplificarse", "reducirse" o "analizarse" y se esfuerza para encontrar estrategias científicas por las cuales, se dejan intactas las interacciones internas y se estudia el sistema como un todo.

Actualmente La Teoría General de Sistemas (TGS) llamada también Enfoque de Sistemas es una disciplina que tiene como ideas principales:

- El Todo es más que la suma de sus partes.
- El Todo determina la naturaleza de las partes.
- Las partes no pueden comprenderse si se consideran en forma aislada del Todo.

- Las partes están dinámicamente interrelacionadas o son interdependientes.

Hoy continúan siendo válidas esas ideas, como se puede observar toma un enfoque holístico de sistemas que genera leyes particulares, mediante el hallazgo de similitudes de estructura (isomorfismos), a través de los sistemas, es decir que el enfoque de sistemas es un método de investigación, una forma de pensar que se basa en el sistema total y no solo en sus componentes; se esfuerza en optimizar la efectividad del sistema total en lugar de mejorar la eficiencia de los sistemas cercanos. Este enfoque logra mejorar los sistemas buscando las causas de su mal funcionamiento dentro de los propios límites de los sistemas, rehusándose a extenderlo y propiciar la investigación con diseños alternos, que abarcarían más allá de los límites de los sistemas inmediatos. Finalmente, el enfoque de sistemas coloca al planeador-desarrollador en el papel de tomador de decisiones, en vez de seguidor de un secuencia de pasos metodológicos.

5.1 Conceptos básicos de la teoría general de sistemas

Sistema. Es una estructura dinámica de personas, objetos y procedimientos, organizados con el propósito de lograr ciertas funciones.

Esta definición general proporciona un modelo de referencia global en el que pueden ubicarse todas las interpretaciones, sin embargo, no es suficiente para ubicar todos los tipos de sistemas que se pueden modelar.

La definición puede refinarse primero, derivando una clasificación en términos de tipos de sistemas y luego

desarrollando un conjunto de conceptos adecuados a cada uno de ellos. La clasificación que se utiliza en la TGS puede resumirse en:

- a) **Sistemas Naturales:** Sistemas físicos que integran el universo, en una jerarquía de sistemas. Desde los sistemas ecológicos hasta los sistemas galácticos.
- b) **Sistemas Diseñados:** Estos pueden ser tanto físicos (edificios, maquinaria), como abstractos (matemáticas, filosofía).
- c) **Sistemas de Actividad Humana:** Por lo general, describen cómo los seres humanos realizan una actividad determinada. Por ejemplo, los sistemas Hombre-Máquina, las empresas, etc.
- d) **Sistemas Sociales y Culturales:** La mayor parte de la actividad humana existirá en un sistema social, donde los elementos serán humanos y las relaciones serán interpersonales. Por ejemplo una familia, los Scout, la Iglesia.

Otra clasificación frecuentemente utilizada es dividir los sistemas en sistemas técnicos ("duros"), y sistemas sociales ("suaves").

- 1) **Sistemas Duros:** La teoría de los sistemas duros o rígidos implica, además del enfoque característico de sistemas, un rigor y una cuantificación estricta para el tratamiento de una situación dada.
- 2) **Sistemas Suaves:** En términos generales, la teoría

de sistemas suaves también implica el enfoque de sistemas, pero es menos rigurosa y mas cualitativa para el tratamiento de una situación a mejorar. Esta tendencia emerge de las ciencias sociales y ciencias del comportamiento.

Según Checkland, las metodologías de sistemas suaves han evolucionado de tal forma, que éstas son el caso general de un enfoque para la solución de problemas del cual los sistemas duros son casos específicos.

Entidades. Las entidades son partes o elementos de un sistema y pueden definirse como algo que tiene realidad física u objetiva y distinción de ser o carácter. Así pues, podemos atribuir ciertas propiedades o atributos a las entidades y por medio de éstos podemos diferenciar una entidad de otra. Además, podemos obtener ciertos valores de éstas propiedades a través de una evaluación. Las propiedades de las entidades definen el grado en que éstas pueden ser reconocidas. Una entidad sin propiedades no puede ser observada.

Agregado. Un agregado es un conjunto de entidades que pueden estar parcialmente interrelacionadas, pero en la cuál al menos una entidad no está relacionada con el resto de ellas.

Relación. La forma en la que dos o más entidades dependen entre sí. Una relación existe si el cambio en alguna propiedad de una entidad, resulta en un cambio en una propiedad de otra entidad. La existencia de una relación impone una limitante en los posibles modos de comportamiento de un sistema. Esto es una consecuencia

directa de la dependencia existe en una relación.

Estructura. Conjunto de relaciones que posee un sistema; la posición de cada una de las entidades con respecto a las otras dentro del sistema como un todo y por último conocer las dimensiones del sistema y de sus partes de una manera cuantitativa. El concepto de relación con el todo y el valor posicional asociado con éste hace posible además, distinguir entre el agregado y el sistema.

Estado. Estado de un sistema es el conjunto de propiedades relevantes que el sistema tiene en un momento del tiempo. La siguiente definición enfatiza su carácter dinámico, haciendo este concepto más claro: "El estado de un sistema conteniendo la información del pasado del sistema y su condición presente, es necesario y suficiente para predecir la salida o la probabilidad de cierta salida dada una cierta entrada".

Subsistema. Es un elemento o componente funcional de un sistema mayor, que tiene las condiciones de un sistema en sí mismo, pero que también tiene un papel de operación de un sistema mayor. Es decir que un subsistema es una parte de un sistema, que puede ser considerado como un sistema en sí mismo, que depende de su nivel de resolución.

Frontera. Límite del sistema y lo que además no se considera como parte del mismo. La selección de la frontera del sistema dependerá del propósito para el cual se está conceptualizando el mismo.

Ambiente. El ambiente es todo lo que está situado fuera de la frontera del sistema. Sin embargo, podemos distinguir

entre un Ambiente Total en el que se agrupa todo aquello que no pertenece al sistema y un Ambiente Relevante que es el conjunto de entidades fuera del sistema que lo afectan o son afectadas, por el sistema bajo estudio.

Ambiente Interno. El ambiente donde ocurren los procesos del sistema; el área de interdependencias internas representa el conjunto de elementos e interrelaciones que integran el sistema en estudio. En este ambiente se comprenden las actividades enfocadas a alcanzar los objetivos del sistema en cuestión.

Ambiente Transaccional. El ambiente donde ocurren procesos de intercambio entre el sistema y su ambiente. El área de interdependencia transaccional entre el sistema social bajo estudio y su ambiente. Representa el conjunto de elementos y sus propiedades relevantes que no son partes del sistema pero que ejercen o reciben influencia directa de éste.

Ambiente Contextual. El ambiente donde ocurren los procesos a través de los cuales los elementos del ambiente de un sistema se relaciona uno con otro. El área de interdependencias que pertenecen al ambiente mismo. Representa el conglomerado de entidades y sus interacciones, que ocurren en el entorno social del sistema y lo afectan de forma indirecta.

Sistema Abierto. Un Sistema Abierto es aquél que interactúa con su ambiente.

Sistema Cerrado. Un Sistema Cerrado es aquél que no interactúa con su ambiente.

Suprasistema. Suprasistema es un sistema cuya función global está contribuyendo al sistema y del cual forma parte.



VI. PARADIGMA DE INGENIERIA DE SOFTWARE BASADO EN LA TEORIA GENERAL DE SISTEMAS

En este capítulo se combinan los conceptos de Ingeniería de Software, Ingeniería de Sistemas y el ciclo de vida de todo proyecto de software, teniendo como resultado un modelo de desarrollo que consta de cuatro fases : Fase de diagnóstico, Fase de Diseño Detallado, y la Fase de Prueba y Mantenimiento, las cuales tienen la característica de estar constituidas por actividades comunes en la transición de una fase a otra, pudiendo expresarse en forma gráfica como una sobreposición entre fases como lo ilustra la figura 7. Asimismo cada una de estas fases cuenta con una serie de pasos que conducen al logro del objetivo de la misma y del proyecto.

Las fases del paradigma son descritas en forma general a continuación e ilustradas en las figuras 8 y 9 respectivamente:

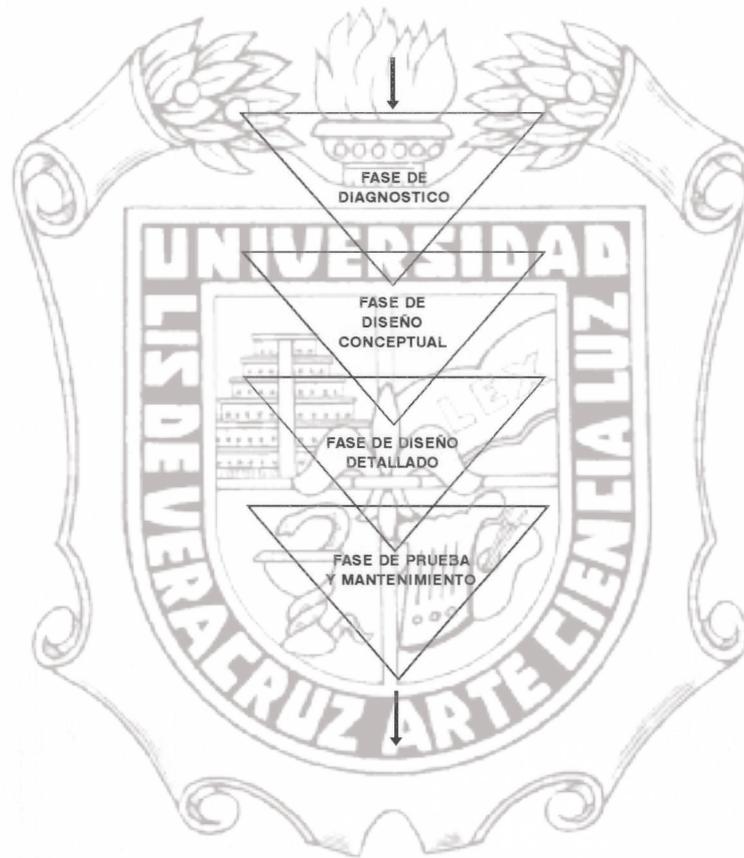


Figura 7. Diagrama general del paradigma propuesto



Figura 8. Fase de diagnostico

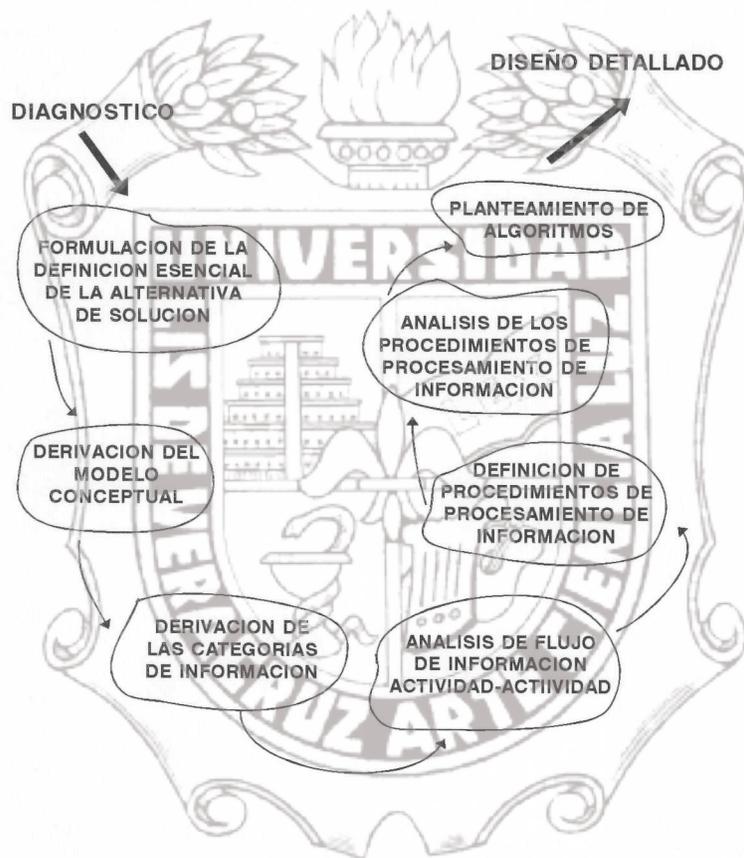


Figura 9. Fase de diseño conceptual

6.1 Fase de diagnóstico.

En esta fase se incluyen dos tipos de actividades. La primera esta formada por los pasos: La situación problemática no estructurada, la situación problemática analizada y la comparación de la situación problemática con los modelos conceptuales; son actividades que se refieren a la realidad, que necesariamente involucra gente en la situación del problema; la segunda los pasos : Formulación de la definición esencial de la alternativa de solución, y la conformación y prueba de los modelos conceptuales; son actividades del pensamiento de sistemas que quizá puedan o no involucrar a aquellas en la condición problemática, dependiendo de las circunstancias individuales del estudio.

En general, el lenguaje de los primeros pasos será el mismo que el lenguaje de la situación problema; la formulación de la definición esencial de la alternativa de solución y el de conformación y prueba de los modelos conceptuales; será mediante el lenguaje de sistemas, porque es en estos pasos donde la complejidad de la realidad, se desenmaraña y entiende como resultado de la traducción a un lenguaje de nivel superior (metalenguaje) de los sistemas.

Los pasos: la situación problemática no estructurada, y la situación problemática analizada; son fases de "expresión" durante la cual se hace un intento por construir la imagen lo más rica posible, no del "problema", sino de la situación en la cual se percibe que hay un problema. La pauta más útil es el interés por ensamblar una imagen hasta donde sea posible, sin imponer una estructura particular en ella. Esta consiste, en el análisis previo al registrar los elementos de estructura dentro de la situación y los elementos de proceso de cambio continuo y

al formar una visión sobre como la estructura y el proceso se relacionan entre sí, dentro de la situación que se investiga. El paso formulación de la definición esencial de la alternativa de solución, involucra el nombrar algunos sistemas que pudieran ser pertinentes al problema y el preparar definiciones concisas de lo que estos sistemas son en contraposición a lo que ellos hacen.

El objetivo es obtener una formulación explícita, cuidadosamente fraseada de la naturaleza de algunos sistemas que subsecuentemente se van a considerar como pertinentes para mejorar la situación del problema. Esto no se puede garantizar, por supuesto, pero la formulación siempre se puede modificar en interacciones posteriores, cuando el entendimiento se profundice. Estas definiciones en el paso Formulación de la definición esencial de la alternativa de solución, se denominan "definiciones esenciales", con lo que se planea indicar que ellas encapsulan la naturaleza fundamental de los sistema elegidos.

El paso conformación y prueba de los modelos conceptuales consiste en la creación de modelos conceptuales de los sistemas de actividad humana, nombrados y definidos en las definiciones esenciales. El lenguaje de construcción del modelo es muy simple, pero emerge como un lenguaje sutil y poderoso (se trata simplemente de los verbos que utilizamos en el lenguaje). Se ensambla un grupo estructurado de verbos que describen las actividades mínimas necesarias requeridas en un sistema de actividad humana, que corresponde con la que se describió en la definición esencial. La construcción del modelo se alimenta del modelo formal de un sistema, el cual se utiliza para

verificar que los modelos construidos no sean fundamentalmente deficientes.

Los modelos del paso conformación y prueba de los modelos conceptuales se introducen entonces en el paso comparación de la situación problemática con los modelos conceptuales, "en el mundo real" y se confrontan con las percepciones de lo que existe ahí. El propósito de esta "comparación", es generar un debate entre gente interesada en la situación problema, que en el paso subsecuente decidirá posibles cambios que simultáneamente satisfagan dos criterios: cambios argumentablemente deseables y al mismo tiempo viables, dada las actitudes de poder ser prevaletentes, ya que tienen que ver con la historia de la situación bajo examinación.

El paso definición de cambios deseables y posibles evalúan los posibles cambios y decide cuál o cuáles de ellos se implementarán en el diseño conceptual. En general, son posibles tres tipos de cambio: de estructura, de procedimientos y de actitudes. La metodología está diseñada para desarrollar los cambios que sean del tipo de procedimientos, es decir, los que caen dentro del desarrollo de un sistema de cómputo. Los pasos de esta fase se muestran a continuación:

6.1.1 La Situación Problemática no estructurada.

La metodología supone que la situación problemática no está estructurada y que difícilmente se conoce su ubicación. Es por ello que en este paso se pretende tener una visión de la situación en la cual es percibido el problema.

6.1.1.1 Confrontación con la situación problemática.

En este paso se pretende familiarizarse con la situación problemática y se identifiquen algunas características del problema. Como una guía para realizar la confrontación con la situación problemática, se sugiere responder a las siguientes preguntas :

*** Familiarización :**

- ¿Qué problemas existen?
- ¿Por qué se generaron?
- ¿Quién cree que son problemas?
- ¿Para quién es importante solucionarlos?

*** Planteamiento inicial:**

- ¿Son problemas o tan solo un síntoma del problema?
- ¿Se han identificado todo los síntomas?
- ¿Cómo se relacionan los problemas y los síntomas?

*** Identificación:**

- ¿Qué tipo de problema es?
- ¿Se puede clasificar de alguna forma?

Es recomendable que la confrontación de la situación problemática se realice mediante un equipo de trabajo, donde preferentemente la gente que participe tenga distintos roles en dicha situación problemática, para lograr obtener una mejor percepción de la situación; por

otro lado, cuidar de no imponer el punto de vista del analista.

Con la finalidad de ilustrar los pasos de la fase de diagnóstico y diseño conceptual, se ha tomado como Caso-Ejemplo un problema hipotético de un banco, el cual se supone que cuenta con servicios de Atención al cliente por la tradicional ventanilla, además de ofrecer el servicio permanente de Cajero Automático. Las características de la operación del Sistema Bancario y su problemática se dan a conocer, según lo requiera la misma metodología, en cada uno de sus pasos.

CASO-EJEMPLO.

Suponga que después de usar cualquier técnica de análisis grupal para el análisis de la situación, se obtuvieron las siguientes declaraciones para la familiarización del problema:

- a) Incremento significativo de la preferencia a los servicios de ventanilla, en comparación con los servicios por cajero automático.
- b) Los servicios que comparten la modalidad de ventanilla y los de cajero automático tienen distintas limitantes, principalmente con respecto a disposiciones de dinero en efectivo.
- c) El departamento de atención a los tarjetahabientes (usuarios), considera que existe un problema de

información al usuario, debido a que las características del servicio de cajero automático son reportadas por el usuario como fallas en el servicio.

- d) La atención al cliente por la modalidad de ventanilla es sensiblemente mas costosa que la del cajero automático.
- f) La imagen y política del banco respecto a la atención al cliente no es congruente en el caso del cajero automático, por no poder disponer de todo el dinero de saldo a favor en cualquier momento.

Como punto de comparación se muestran a continuación las respuestas cortas a las preguntas de familiarización del inciso.

¿Qué problemas existen?

- a) Incremento en la demanda de uso del servicio de ventanilla.
- b) Incremento en los costos del servicio a tarjetahabientes.
- c) Disminución de la imagen y calidad del servicio.

¿Por qué se generaron?

Por existir diferentes características entre el servicio de cajero automático y el de ventanilla, sin que el tarjetahabiente las conozca.

¿Quién cree que son problemas?

- a) Departamento de Servicio al tarjetahabiente.
- b) Departamento de costos.
- c) Departamento de políticas y lineamientos.

¿Para quién es importante solucionarlos?

- a) Departamento de Servicios al tarjetahabiente.
- b) Departamento de costos.
- c) Departamento de políticas y lineamientos.
- e) Departamento de ventanillas.

Es importante observar que las preguntas guías de la metodología, no necesariamente deben tener una respuesta directa, si no que las respuestas pueden estar contenidas en las declaraciones del grupo de trabajo.

6.1.1.2 Ubicación de la situación problemática.

Este paso pretende delimitar el campo de acción del estudio y ubicar las entidades que hasta el momento componen a la situación problemática y las relaciones entre ellos. Con la información obtenida hasta el momento, se desarrolla un diagrama o mapa que muestre los componentes de la situación problemática, resaltando la conectividad y la ubicación de las variables endógenas y exógenas.

Las entidades son representadas por figuras amorfas cerradas contenidas mediante flechas que indican el sentido de influencia y la variable de afectación entre entidades. Las variables exógenas no necesariamente provienen de una entidad especificada y están delimitadas por la frontera de

la situación problemática diagramada.

CASO-EJEMPLO.

Para lograr la ubicación del problema con mayor facilidad, se analizan los elementos de la estructura de acuerdo a la organización departamental del banco (Departamento de tarjetas de crédito, Departamento de Atención por Cajero Automático, Departamento de Políticas y Lineamientos, Departamento de Atención al cliente y Departamento de Atención por Ventanilla), digramándose de la siguiente manera: (ver Fig. 10)

Es posible lograr la ubicación del problema de otras formas como es la de jerarquías, por funciones, distribución física, etc., debiendo utilizar la que mejor represente la estructura de la situación problemática.



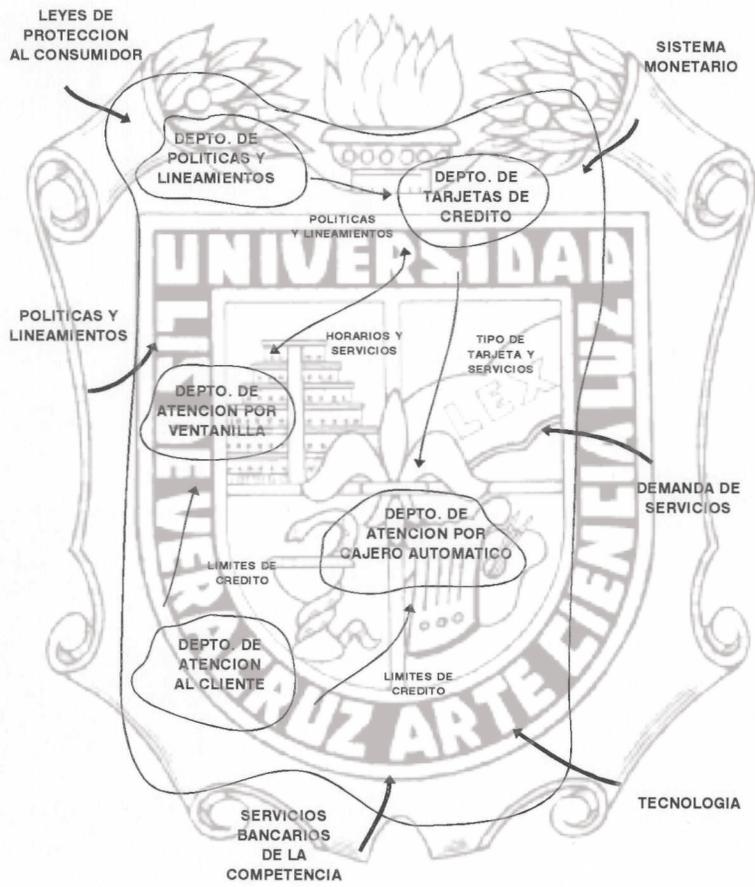


Figura 10. Diagrama sistémico causa efecto

6.1.2 Situación problemática analizada.

La función de este paso es mostrar la situación problemática de modo que un rango de elecciones relevantes posibles puedan ser descubiertas, es decir los sistemas seleccionados sean pertinentes a la solución del problema.

La situación problemática se analiza mediante el estudio de su estructura y el proceso, donde la estructura se examina en términos de distribución física, jerarquía de poder, estructura de reporte y del patrón de comunicaciones, tanto formal como informal. El proceso se examina en términos de las actividades básicas requeridas para decidir hacer algo, para llevar eso a cabo, para monitorear qué tan bien está hecho y sus efectos externos, y para implementar la acción correctiva adecuada. La relación entre estructura y proceso viene siendo el clima de la situación, la cuál es una característica esencial de la situaciones en las cuales se percibe el problema.

Ahora es necesario poder visualizar la situación problemática en una forma más estructurada, pero sin comprometerse a una situación particular. Se ha encontrado útil usar tres líneas de guía en este paso, las cuales consisten en contestar las siguientes preguntas específicas por separado.

- a) ¿ Cuáles son los elementos en la estructura organizacional que están relacionados con la situación problemática?
- b) ¿ Cuáles son los elementos en los procesos organizacionales que están relacionados con la

situación problemática?

- c) ¿Cuál es el clima organizacional que caracteriza a la situación problemática confrontada?

El resultado final idealmente es un panorama estructurado de la situación problemática, posible de ser ilustrado en un diagrama sencillo de sus ambientes interno, transaccional y conceptual.

La selección de los sistemas pertinentes es cuestión de juicio, pero como no es una selección única y para todo, el analista aprenderá de la efectuada y le será posible cambiarla basándose en el aprendizaje adquirido.

CASO-EJEMPLO.

Después del análisis de la situación problemática por los cuestionamientos de los incisos a), b), y c) del punto 2, se procede a representar el clima organizacional en función de los sistemas que se consideran pertinentes y que nos llevarán a una solución, en el caso del banco los subsistemas seleccionados son: acceso al sistema, verificación y validación, servicios y operación, status del sistema, interacción con el usuario y ejecución del servicio, los cuales se diagrama en su ambiente interno, transaccional y contextual como se muestra a continuación. (ver fig. 11)

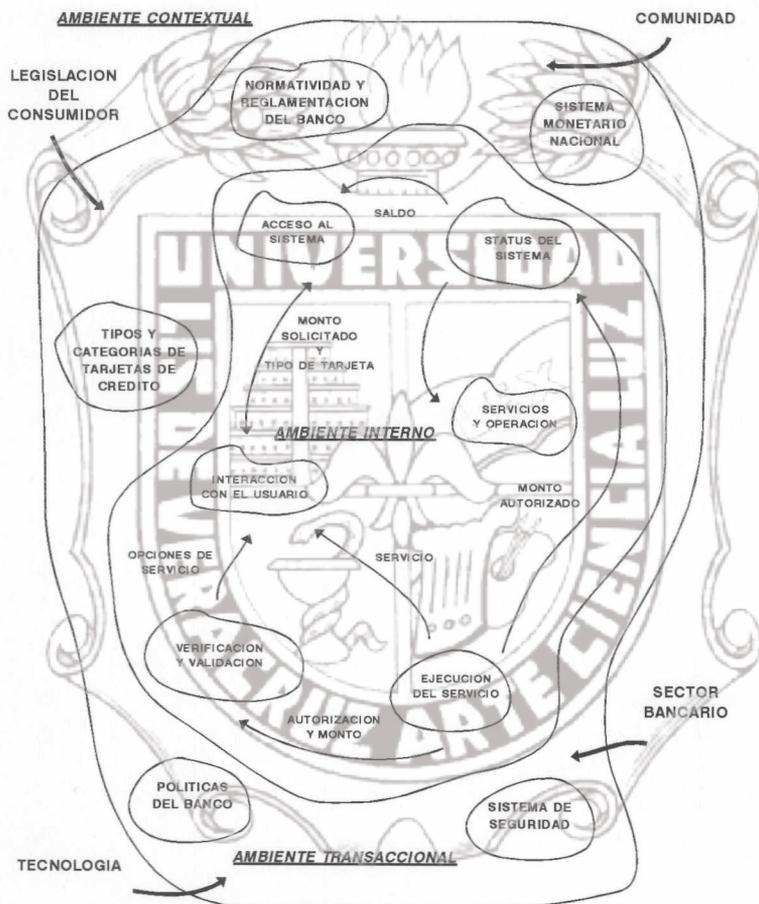


Figura 11. Mapa de ambientes

6.1.3 Definición esencial de los sistemas relevantes.

6.1.3.1 Definir los sistemas relevantes a la situación problemática.

Este paso es uno de los más importantes de la metodología. Se deberá seleccionar una forma específica de visualización de la situación problemática, con base en el panorama obtenido en el paso 2 y definir el problema. Esta selección se hace identificando y definiendo los sistemas relevantes a la situación problemática; esto es, se deben seleccionar los sistemas que serán relevantes para el logro de algún tipo de mejora visualizada. Estos sistemas serán generalmente sistemas de actividad humana y no necesariamente deberán coincidir con agrupaciones existentes en la organización.

Una vez que se han seleccionado los sistemas (relevantes), es necesario formular y discutir definiciones (esenciales) que expresen la naturaleza básica de cada sistema (relevante).

Las definiciones esenciales deben reflejar la forma seleccionada para visualizar la situación problemática; cada una de ellas consiste de una definición penetrante derivada de la riqueza de análisis, que sea reveladora para aquellas personas que estén involucradas en la operación del sistema o de los sistemas relacionados con el sistema relevante.

CASO-EJEMPLO.

Suponga que el servicio de mayor demanda en un cajero automático, es la disponibilidad de dinero en efectivo, adicionalmente, una de las políticas del banco es que el cliente pueda hacer efectivo el crédito autorizado y lógicamente, del saldo a favor en su caso.

Con base en lo anterior y la información que se ha obtenido en los pasos anteriores el analista visualiza un posible mejoramiento y lo define para su estudio como sistema relevante:

"Sistema de disponibilidad en efectivo del cajero automático del banco", donde el analista expresa la definición esencial del sistema relevante mediante:

"Sistema que atiende las 24 hrs. del día, las solicitudes de disponibilidad en efectivo, tomando en cuenta el saldo a favor en caso de existir".

6.1.3.2 Formulación de las definiciones esenciales de los sistemas relevantes.

Una definición esencial debe ser una descripción concisa de un sistema de actividad humana, la cuál tiene un punto de vista particular (percepciones) del sistema. A menudo es útil incluir un número de restricciones en la definición, y el efecto de éstas restricciones puede ser probado al comparar sobre el modelo.

La hipótesis basada en la intuición y la experiencia

formula que una definición esencial contiene cinco elementos explícitamente. El núcleo de una definición esencial de un sistema será un proceso de transformación (T), es decir, los medios por los cuales las entradas definidas se transforman en salidas definidas. La transformación incluirá el objeto directo de los verbos de actividad principal que se requieren subsecuentemente para describir el sistema. Habrá un dueño (D) del sistema que tenga un interés primario en el sistema y el poder para ocasionar que el sistema deje de existir. Los dueños pueden argumentar acerca del sistema.

Dentro del sistema habrá actores (A), los agentes que llevan a cabo o que ocasionan que se lleven a cabo las actividades principales del sistema. Especialmente, en su transformación principal dentro o fuera del sistema, habrá lo que denominamos consumidores (C) del sistema, que pueden ser beneficiarios o víctimas afectadas por las actividades del sistema. Los consumidores serán objetos indirectos de los verbos principales para describir al sistema.

En quinto lugar, habrá restricciones del medio (E) en el sistema, características de los medios del sistema y sistemas más amplios que éste tiene que asumir como "dadas".

Además de los anteriormente elementos mencionados existe uno que debido a su naturaleza, raramente está explícito en una definición esencial, pero que no es posible excluir; debido a la perspectiva imagen o marco que de significado a esta definición esencial particular la

cual es denominado como Weltanschauung (W) es decir un particular punto de vista, pudiendo existir más de un W en una definición esencial.

Los seis elementos descritos conjuntamente hacen una definición esencial bien formada, representándose por el mnemónico CATWDE.

La omisión de cualquiera de los elementos del CATWDE en la definición esencial deberá ser consciente o deliberada por alguna razón particular. Así mismo el que contenga los seis elementos no quiere decir que sea una buena definición esencial, sino que está formulada correctamente.

Todo el proceso de análisis de CATWDE se relaciona con la identificación de la definición esencial, no es un mecanismo para ampliar de manera inconsciente la definición. Es frecuente encontrar elementos descritos bajo los encabezados de CATWDE, que están en la mente del analista pero que no se encuentran en forma explícita en la definición esencial. Si el análisis de CATWDE, ha señalado ciertos elementos y éstos se consideran importantes, entonces la definición esencial se modificaría para incluirlos como se muestra a continuación: (ver fig. 12)

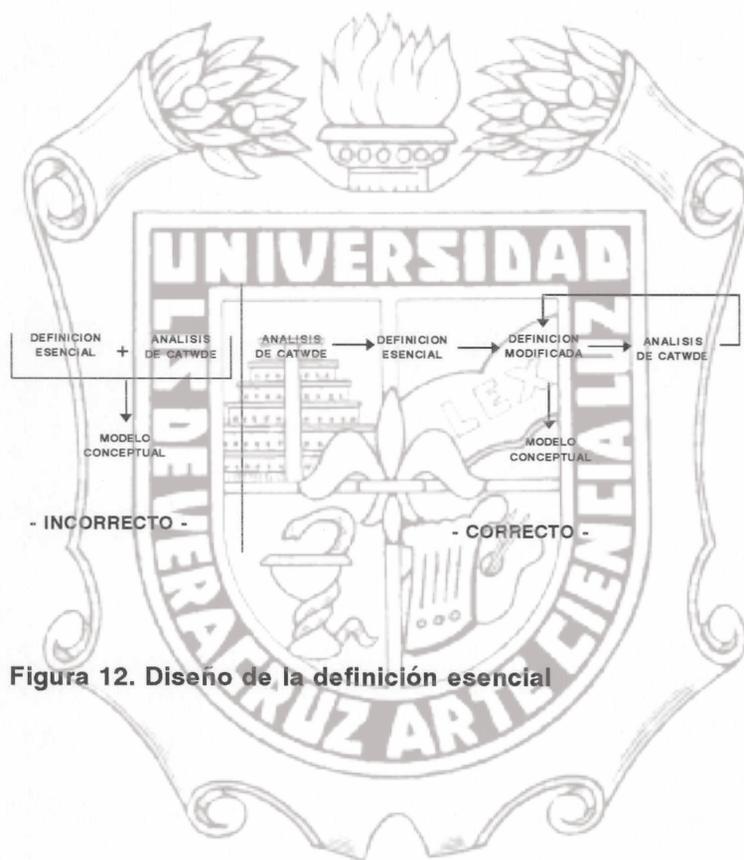


Figura 12. Diseño de la definición esencial

Después de todo, es la definición esencial la que está en proceso de modelado.

CASO-EJEMPLO.

Una vez que se ha formulado la definición esencial del sistema relevante es conveniente realizar un análisis de los seis elementos que debe de contener para que esté formulada correctamente.

Los elementos del CATWDE serían de la siguiente forma:

- a) Cliente: Banco.
- b) Actores: Subsistemas del cajero automático.
- c) Transformación: La atención de la solicitud de disposición de dinero en efectivo.
- d1) Weltanschauung-1: Automatizar el servicio de disposición de dinero en efectivo de acuerdo a la normatividad sin menoscabo de la calidad del servicio y seguridad.
- d2) Weltanschauung-2: Atender las solicitudes de disponibilidad de dinero en efectivo mientras no excedan el crédito autorizado cumpliendo la normatividad.

Se puede observar que el W-1 está dado por el punto de vista de operación del banco, mientras que el W-2 esta dado desde el punto de vista de atención al cliente.

- e) Dueño: Banco.
- f) Restricciones: Sistema monetario, estado de cuenta del solicitante, tipo de tarjeta, tecnología.

Con base en la definición de los elementos se modifica la definición esencial y nuevamente se hace el análisis del CATWDE, hasta lograr que sea completa y tenga congruencia entre los elementos. Suponga que finalmente se obtiene la siguiente definición: Un sistema bancario de servicio automático que atienda las solicitudes de disponibilidad de dinero en efectivo de los tarjetahabientes, por medio de los subsistemas del cajero automático en función de la normatividad, estado de cuenta y crédito del tarjetahabiente, y las políticas del banco.

6.1.4. Conformación y prueba de los modelos conceptuales.

6.1.4.1 Análisis Funcional.

En este paso se hace uso del principio que rige a todo sistema y que es el de estar caracterizado por un grupo de actividades interconectadas entre sí para el logro de los objetivos del sistema. Estas actividades en conjunto, constituyen el proceso de transformación que convierte las entradas del sistema en salidas.

La definición esencial se puede considerar como una descripción de un grupo de actividades humanas con un propósito determinado, concebido como un proceso de transformación. Con la cual construimos un modelo que

especifica la definición esencial. La definición es un reporte de que es el sistema; el modelo conceptual constituye un reporte de las actividades como debe ser para alcanzar la definición.

Las actividades del modelo se presentan en muy diversos niveles de agregación, pero con un poco de experiencia es posible distinguir un grupo de actividades con el mismo nivel de complejidad, que podríamos llamar las actividades necesarias y suficientes para el logro del objetivo del sistema. En este caso, al sistema seleccionado se le aplica un procedimiento para determinar dichas actividades, a las cuales se les denomina actividades sustantivas. Este proceso se conoce como análisis funcional.

El proceso de análisis funcional puede iniciar simplemente al listar todas las actividades que el sistema realiza. Es posible agrupar conjuntos de actividades que por naturaleza pertenecen a una actividad de mayor jerarquía. Las actividades resultantes deben examinarse con los siguientes criterios:

- ¿Son actividades del mismo nivel jerárquico?
- ¿La actividad es necesaria para el logro del objetivo del sistema?
- Es el conjunto de actividades suficiente para el logro del objetivo del sistema?

Se recomienda que el número de actividades sustantivas sea relativamente pequeño, digamos de 6 a 12 actividades como máximo.

Todo lo anterior podría generalizarse en la secuencia siguiente para conformar un modelo:

- a) A partir de la definición esencial y de los elementos del CATWDE de ésta, formar una impresión del sistema, visto como una entidad autónoma que lleva a cabo un proceso de transformación físico o abstracto.
- b) Ensamblar un número pequeño de verbos que describan las actividades más fundamentales en el sistema ya descrito. Tratar de mantener un nivel de resolución, evitando la mezcla de actividades definidas en diferentes niveles de detalle.
- c) Si se puede justificar a partir de la definición esencial-, estructurar las actividades en grupos de actividades similares.
- d) Conectar las actividades y los grupos de actividades con flechas que indiquen las dependencias lógicas.
- e) Indicar los flujos (concretos o abstractos) que sean esenciales para expresar lo que el sistema hace. Distinguir estos flujos de las dependencias lógicas, y en cualquier caso, mantener el número de flujos a un mínimo necesario.
- f) Verificar que la definición esencial y el modelo

conceptual juntos, constituyan un par de declaraciones de qué es el sistema y qué hace el sistema.

Una vez que se ha constituido este modelo, se puede ampliar, pasando a lo que podríamos llamar actividades de segundo nivel. Para esto, cada actividad es desglosada en subactividades sustantivas siguiendo el mismo procedimiento. Los niveles de detalle que se requieran están en función de la experiencia y conocimiento que tenga el analista del sistema, así como también del tipo de aplicación.

CASO-EJEMPLO.

Tomando como base la definición esencial y los elementos del CATWDE del paso 3, las actividades sustantivas del primer nivel en el modelo son:

- Controlar el acceso al sistema.
- Monitorear los saldos.
- Calcular los nuevos saldos.
- Interpretar solicitudes.
- Generar y activar respuestas.
- Almacenar saldos de tarjetahabiente.
- Especificar tipo de tarjetas.

- Almacenar el saldo disponible del cajero automático.
- Especificar límites de crédito.

El modelo quedaría representado de la siguiente forma: (ver fig. 13)



Figura 13. Modelo conceptual

6.1.4.2 Evaluación por propiedades de los sistemas.

No existen modelos validos e inválidos, solamente modelos conceptuales sustentables y modelos que son menos sustentables. Pero al menos sí es posible el verificar que los modelos conceptuales no sean fundamentalmente deficientes, y esto se logra mediante la comparación del modelo obtenido con un modelo general de cualquier sistema de actividad humana conocido como sistema formal, teniendo determinadas propiedades, por lo cual al análisis se le denomina " Análisis por propiedades de los sistemas", teniendo como base los siguientes conceptos :

El concepto de sistema asume que un conjunto de subsistemas operarán juntos a través de sus CONECTIVIDADES, para el logro de los objetivos del sistema del cual forman parte.

Por objetivo debe entenderse, una situación deseada que el sistema trata de alcanzar, como se expresa en una definición esencial. A este objetivo se le pueden fijar valores que se espera sean cumplidos en un determinado tiempo (METAS). Para lograr los objetivos del sistema, los componentes de éste realizan una serie de actividades que en su conjunto constituyen el proceso de transformación del sistema.

El tener un objetivo implica tener indicadores con los cuales se puede medir y evaluar el avance hacia el logro del mismo, es decir MEDIDAS DE DESEMPEÑO de su operación. La diferencia entre el valor deseado o meta de la medida de desempeño y el valor real medido, produce una discrepancia,

que motiva a aplicar acciones correctivas de CONTROL, para mantener el sistema en la dirección correcta.

El seguimiento y control de la operación motiva la necesidad de contar con un proceso para analizar la naturaleza de la discrepancia y proponer alternativas de solución y su correspondiente TOMA DE DECISIONES. Aplicar una acción de control implica la modificación del sistema, a través de la asignación de recursos. La propiedad de conectividad tiene una FRONTERA a través de la cual existirán flujos de información y de recursos. La FRONTERA del sistema delimita el área dentro de la cual el tomador de decisiones puede ejercer autoridad y control.

Una forma de probar el modelo conceptual es a través de las propiedades de un sistema, por lo que deberá de contener implícitamente o explícitamente (ver Fig.14):

- OBJETIVO.
- SUBSISTEMAS.
- CONECTIVIDADES.
- METAS.
- MEDIDAS DE DESEMPEÑO.
- FRONTERA.
- CONTROL.
- TOMA DECISIONES.

Es importante enfatizar que el propósito de un modelo conceptual no es ilustrar o demostrar "lo que debería existir en la realidad". El verdadero propósito de un modelo conceptual es estructurar una comparación entre la

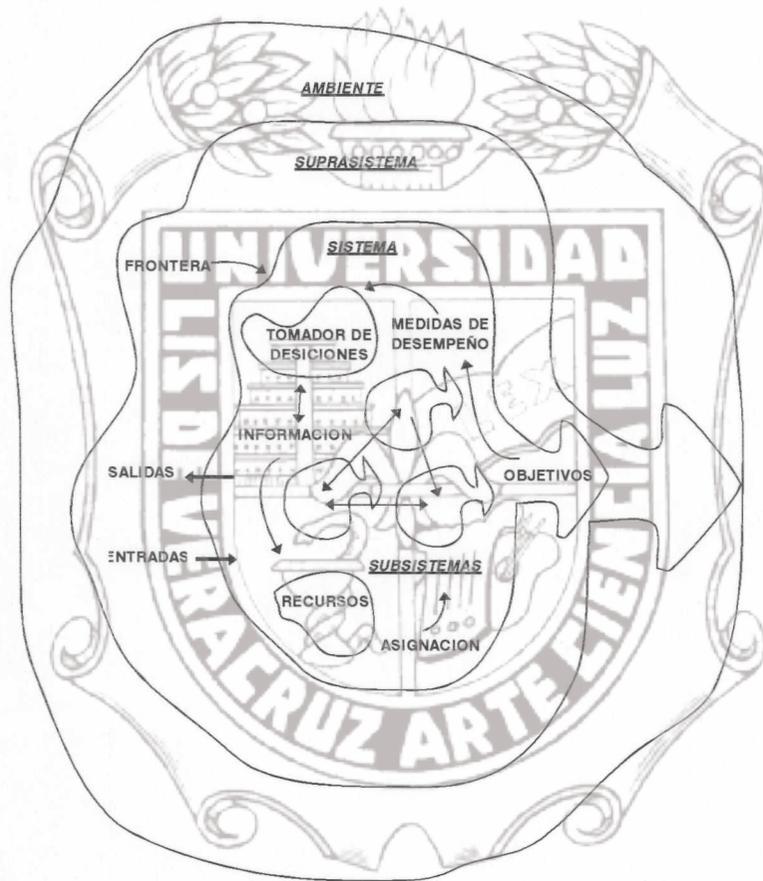


Figura 14. Propiedades de los sistemas

situación existente y la situación descrita por el modelo conceptual, construido en base a la definición esencial y así poder definir mejoras a la situación problemática existente.

CASO-EJEMPLO.

El análisis por propiedades que se aplica al modelo, consiste en identificar si existen los componentes del un sistema formal en el modelo conceptual.

Objetivo: Proporcionar el servicio de disponibilidad de dinero en efectivo por medio del cajero automático, bajo la normatividad del banco.

Medida de desempeño: Número de solicitudes debidamente atendidas.

Conectividades: Existen entre los subsistemas, (Ver fig. 13).

Toma de decisiones: Existe un subsistema de monitoreo y control donde se toman las decisiones.

Frontera: El sistema esta ubicado y delimitado por un ambiente, y suprasistema. (Ver figura No.13).

Flujo de información: Están implícitamente y explícitamente representadas por las conecciones entre los subsistemas del modelo.

Jerarquía de sistemas : El modelo está compuesto por

el sistema, sus subsistemas y un sistema mayor al que pertenecen. (Ver figura No. 13).

6.1.5 Comparación de la situación problemática con los modelos conceptuales.

Es un asunto de juicio, el decidir cuándo detener la construcción del modelo conceptual y pasar a la comparación con la realidad entre lo que existe ahí y lo que sugieren los modelos de sistemas, que se piensa son pertinentes para el problema. El propósito de esta comparación es el establecer un debate con las personas involucradas en el problema, con el fin de definir cambios posibles que puedan mejorar la situación existente.

El panorama estructurado de la situación problemática obtenida en el paso 4.1.2, se compara formalmente con los modelos conceptuales derivados en los pasos 4.1.3 y 4.1.4.

La comparación entre el modelo conceptual y la realidad puede hacerse de cuatro maneras diferentes:

- a) El modelo se usa como fuente de preguntas referentes a la situación existente. Estas preguntas se escriben y se contestan sistemáticamente y las respuestas son las que dan una orientación sobre los problemas y soluciones, a los participantes.
- b) Realizar un estudio retrospectivo, es decir reconstruir una secuencia de eventos en el pasado y ver que fue lo que ocurrió y lo que hubiera pasado si se hubiera usado el modelo.

- c) Otra forma consiste en identificar qué propiedades de los modelos conceptuales que son especialmente diferentes con la realidad y por qué.
- d) La comparación también puede hacerse construyendo un segundo modelo: éste modelo será ahora de la situación real y aquí se verán las diferencias que existen entre el modelo conceptual y el modelo de la situación real.

El objetivo de esta comparación es definir cambios potenciales que pueden implementarse en la realidad. Entonces, si los modelos conceptuales resultan ser muy similares a los existentes en la realidad, probablemente será necesario regresar al paso 6.1.3.2 y formular definiciones esenciales más radicales. Por otro lado, si los modelos conceptuales resultan demasiado apartados de la situación real, podría requerirse de la formulación de otra definición esencial más restringida.

CASO-EJEMPLO.

Utilizando el primer método de comparación se enfatiza la utilidad de una discusión general sobre la naturaleza del modelo y cualquier organización implicada en ella, cuando se relacionan con la naturaleza de lo que se cree que existe. Así, tienden a surgir asuntos estratégicos durante este tipo de discusión, en relación con la existencia de ciertas actividades antes de que los asuntos alcancen en un nivel detallado de procedimiento. La discusión mas útil de este tipo, ocurre cuando todos los participantes están familiarizados con el enfoque de

sistemas. Para el caso del banco supongamos que después de los cuestionamientos se obtuvieron las siguientes declaraciones como diferencias entre el modelo y la realidad:

- a) El sistema actual ofrece el servicio tomando en cuenta una limitada cantidad de información, que restringe el uso del cajero automático y es contraria a la política del banco, situación que en el modelo se ha superado.
- b) Los lineamientos y políticas forman parte del sistema en el modelo conceptual.
- c) El subsistema de monitoreo y control incluye el control de acceso del sistema.
- d) La interacción con el usuario se produce desde el subsistema de operación y el subsistema de monitoreo y control en el modelo conceptual.

6.1.6 Definición de cambios deseables y posibles.

La definición de los cambios deseables y posibles es un proceso de toma de decisiones que debe llevarse a cabo, considerando que los cambios pueden ser de tipo estructural, de procedimientos y de actitudes.

Para la toma de decisiones se deben identificar todas las alternativas disponibles y considerar los riesgos que implican, así como la evaluación de los mismos. Por otro lado, se recomienda utilizar una técnica de evaluación de alternativas, que permita la inclusión de factores

relevantes propios de la empresa, así como la ponderación de los mismos.

El objetivo final de este paso es la definición de los cambios más apropiados para la empresa y que sean a su vez factibles.

Es importante hacer notar que solamente mediante la selección de una alternativa de cambio que sea del tipo de procedimiento, - como lo es la modificación o desarrollo de un sistema de cómputo-, permitirá la continuidad de la metodología que se está aplicando, como parte del paradigma de desarrollo de Ingeniería de Software basado en la Teoría General de Sistemas.

6.2 Fase de diseño conceptual.

El diseño conceptual tiene como propósito desarrollar un modelo de solución con un alto grado de realismo y confiabilidad, el cual contenga los suficientes elementos, que permitan en la fase de diseño detallado realizar un diseño del producto de software y lograr un producto de software que prevea la realización de pruebas y la factibilidad de aplicar mantenimiento cuando sea necesario.

La fase de diseño conceptual es sin lugar a duda determinante en la metodología, dado que genera la solución a la problemática detectada previamente en la fase de diagnóstico además de ser la base para las siguientes fases (diseño detallado y prueba y mantenimiento).

Los dos primeros pasos de esta fase (ver figura No.

9), se llevan a cabo de la misma forma que los pasos formulación de la definición esencial de la alternativa de solución y la conformación y prueba de los modelos conceptuales de la fase de diagnóstico, y en el caso particular de que en el paso definición de cambios posibles y deseables, de la fase de diagnóstico, se decida desarrollar el modelo conceptual obtenido en esa fase, los dos primeros pasos de la fase de diseño conceptual se deben sustituir por los pasos mencionados de la fase de diagnóstico.

En el paso derivación de las categorías de información se analiza el tipo de información del modelo conceptual, para posteriormente agruparlas en categorías, las cuáles son analizadas junto con los flujos de información, en el paso análisis de flujo de información actividad-actividad.

En el paso definición de procedimientos de procesamiento de información, se considera que el sistema requerirá, para la realización de sus funciones, apoyarse en determinados procedimientos de procesamiento de información, pudiendo ser éstos automáticos (desarrollados o adquiridos) ó manuales. En este paso se definen y se validan con la información que requieren para realizar el procesamiento, así como la información que generan.

Una vez que se tienen todos los elementos se efectúa un análisis de procedimientos de procesamiento de información y del sistema en general a través de la Cruz Maltesa, tratando de obtener como resultado el planteamiento de desarrollo de determinados algoritmos que satisfagan los requerimientos del diseño conceptual.

La Cruz Maltesa es una matriz de cuatro áreas, donde la mitad superior contiene las actividades del modelo que se está desarrollando, en la etapa de diagnóstico (descrita en el paso 6.1.1.4), además de una indicación de los flujos

de información entre actividades. La mitad inferior contiene una declaración de los procedimientos del procesamiento de información, los cuales deberán ser los que finalmente se definan para el manejo de la información del modelo y que serán la base para el desarrollo a nivel de módulos o rutinas de programación.

La estructura de la Cruz Maltesa se muestra en la figura 15, donde la zona norte contiene un conjunto de actividades que constituyen el sistema relevante bajo estudio.

Las zonas este y oeste son idénticas y contienen las categorías de información, consideradas esenciales para el apoyo de las actividades en este nivel de resolución. La zona oeste (que representa las entradas) es la imagen reflejada del este (que representa las salidas).

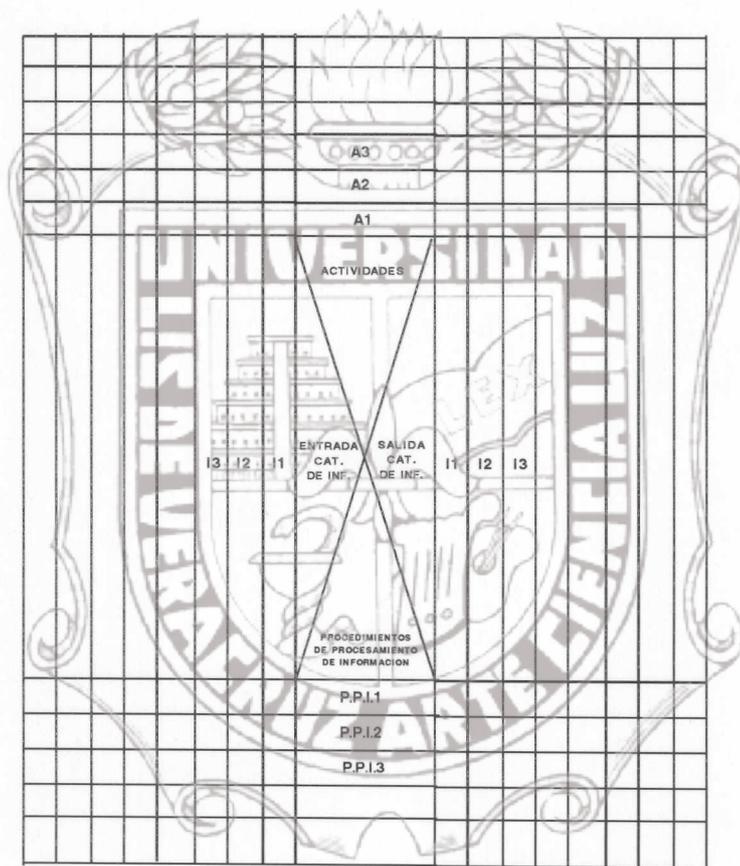


Figura 15. Estructura de la Cruz Maltesa

La zona sur es una lista de los procedimientos del procesamiento de información (automatizados y manuales) que deberán existir, por lo que podrán modificar los existentes o desarrollar nuevos procedimientos según sea el caso.

La Cruz Maltesa también se puede describir dividiéndola en dos partes. La parte superior representa el modelo conceptual y sus flujos de información, la mitad inferior representa los procedimientos de procesamiento de información requeridos.

Las zonas suroeste y sureste representan las entradas y salidas de los procedimientos de procesamiento de información, en este nivel de resolución particular.

Por lo tanto, puede obtenerse un mapeo de los procedimientos de procesamiento de información en el modelo conceptual, relacionando la mitad inferior con la mitad superior de la Cruz Maltesa.

Lo anterior nos permite definir los procedimientos de procesamiento de información con una visualización global clara del sistema y del impacto que tendrán en éste, que como consecuencia permite un mejor diseño de los mismos.

Para ilustrar el potencial de la utilización de la Cruz Maltesa, se describen los siguientes puntos de análisis:

- Examinar las columnas en la zona suroeste proporcionará los requerimientos para la disposición y operación de varias bases de datos. Por lo tanto, un gran número de entradas en una

columna particular podría constituir un argumento para la formación de una base de datos relevante para ese conjunto de datos particular, junto con la definición de los procedimientos de procesamiento de información.

- Un examen de las categorías de información en la zona noroeste, define las actividades que tienen la responsabilidad de monitorear y actualizar el contenido de datos.
- Las interacciones más importantes con los procedimientos de procesamiento de información se identifican mediante un examen de la matriz sureste, donde se puede detectar la duplicidad potencial del procesamiento de datos; necesita examinarse para lograr un desarrollo eficiente de los procedimientos de procesamiento de información.
- Además de usar la Cruz Maltesa como un medio para destacar las interacciones y duplicidad potencial de procesamiento de información, en la definición de los procedimientos de procesamiento de información, también pueden existir procedimientos de procesamiento de información que no requieran un desarrollo de un sistema informático.

En resumen, la Cruz Maltesa permite observar la información que requiere una determinada actividad, qué procesamiento la generó y a través de cuál(es) actividad(es). Así mismo, de esa actividad se conoce qué

información genera, mediante qué procesamientos, y por quién es utilizada.

Es pertinente aclarar que no es necesario producir una gran cruz que represente el nivel más alto de resolución, ya que esto puede producir una matriz de una dimensión inmanejable. Se recomienda crear una matriz para cada subsistema, donde el nivel de resolución sea del mismo nivel que el del modelo conceptual, y así poder visualizar las categorías de información en el nivel previo.

Considerando que la Cruz Maltesa utiliza las actividades sustantivas del modelo conceptual y sus flujos de información, complementándose con los procedimientos de procesamiento de información que se pretenden utilizar, se procede a describir la fase de diseño mediante los pasos que llevan a la conformación de la Cruz Maltesa, los cuales fueron ilustrados en la figura 9 y que se describen a continuación:

6.2.1 Formulación de la definición esencial de la alternativa de solución.

En este paso se utiliza nuevamente el CATWDE para la construcción de la definición esencial, la cual servirá de base a su vez, para la construcción del diseño de un sistema de software. Es de esperarse que la definición esencial elaborada en el paso 6.1.3 de la etapa de diagnóstico sea igual o muy similar, esto dependerá de qué cambios deseables y posibles conformen la alternativa de solución seleccionada. Para ello, se recomienda tratar de mantener la definición esencial original, a fin de que el modelo conceptual sea válido para el diseño.

CASO-EJEMPLO

En el caso-ejemplo, el paso 6.2.1 se considerará como el equivalente a los pasos 6.1.3 desarrollado en la fase de diagnóstico, por tratarse de la misma definición esencial.

6.2.2 Derivación del modelo conceptual.

De forma similar que en el paso 6.1.4 de la fase de diagnóstico, se debe desarrollar un modelo conceptual de la solución, a partir de la definición esencial del paso anterior. Por otro lado se recomienda que el modelo sea de un nivel de bajo detalle. De preferencia, se puede iniciar con un modelo de primer nivel que contenga únicamente las actividades sustantivas, asimismo el modelo conceptual puede ser similar o coincidir con el elaborado en la fase de diagnóstico y los cambios, (en caso de que los hubiera) deben ser congruentes y manteniendo el enfoque de sistemas.

CASO-EJEMPLO.

En el caso ejemplo, el paso 6.2.2 se considerará como el equivalente al desarrollado en la Fase de Diagnóstico en el paso 6.1.4, por tratarse de la misma definición esencial y su correspondiente modelo conceptual.

6.2.3. Derivación de las categorías de información.

En este paso se analizan cada una de las actividades del modelo conceptual, como si fueran procesos de transformación en los que se requiere determinada

información para realizar la actividad, y que a su vez, -al realizarla-, se produce o actualiza cierta información, adicionalmente se identifica cual información es necesaria para evaluar el desempeño de la actividad. Posteriormente se agrupan en categorías de información, pudiendo llegar a definir una estructura de datos jerárquica si se desea.

6.2.3.1 Identificación de la información

Información requerida.

Para cada actividad se identifica la información que necesita para poder llevarse a cabo.

Información generada.

Para cada actividad identificar la información que genera al realizarse.

Información de control.

Consiste en identificar la información necesaria para poder llevar a cabo un monitoreo de la medida de desempeño de cada una de las actividades.

CASO-EJEMPLO.

Para ilustrar la utilización de la Cruz Maltesa, se analizan el subsistema de operación y el subsistema de seguimiento y control. La identificación de la información de los subsistemas, se lleva a cabo por medio de la elaboración de una tabla de las actividades que se realizan

y la información que tienen como entrada, salida o que es necesaria para evaluar el desempeño de la misma actividad.

La siguiente tabla muestra los resultados de la identificación de la información.



TABLA DE INFORMACION REQUERIDA

ACTIVIDADES				
Control de acceso	Monitoreo de saldos	Interpretar solicitudes	Realizar cálculos	Activar respuesta
ENTRADAS				
Tipo tarjeta	Saldo - Sistema	Saldo - Tarjeta	Saldo - Tarjeta	Dictamen
Número confidencial	Saldo - Tarjeta	Saldo - Sistema	Saldo - Sistema	
Saldo - Sistema		Límite crédito		
Monto solicitado				
Saldo - Tarjeta				
SALIDAS				
Dictamen respuesta	Saldo - Sistema	Dictamen	Saldo - Tarjeta	Mensaje respuesta
Activa respuesta	Saldo - Tarjeta	Factibilidad	Saldo - Sistema	Saldo Sistema
Actualiza saldos				Saldo Tarjeta
MEDIDAS DE DESEMPEÑO				
Tiempo fuera de servicio	Movimientos no registrados	Dictamen erróneo	Operaciones con error	Respuestas erróneas
Emitir respuesta de servicio equivocada				

Accesos negados indebidamente	Saldos incorrectos			Actualización errónea
Autorización indebida				

6.2.3.2 Definición de las categorías de información.

Las categorías de información se definen en función del tipo de información y los datos que la conforman. Es pertinente no perder de vista que los datos procesados se vuelven información cuando se emplean en alguna actividad con alguno propósito, recordando que no es la forma ni el contenido de los datos lo que cambia, sino el uso o interpretación en cada actividad.

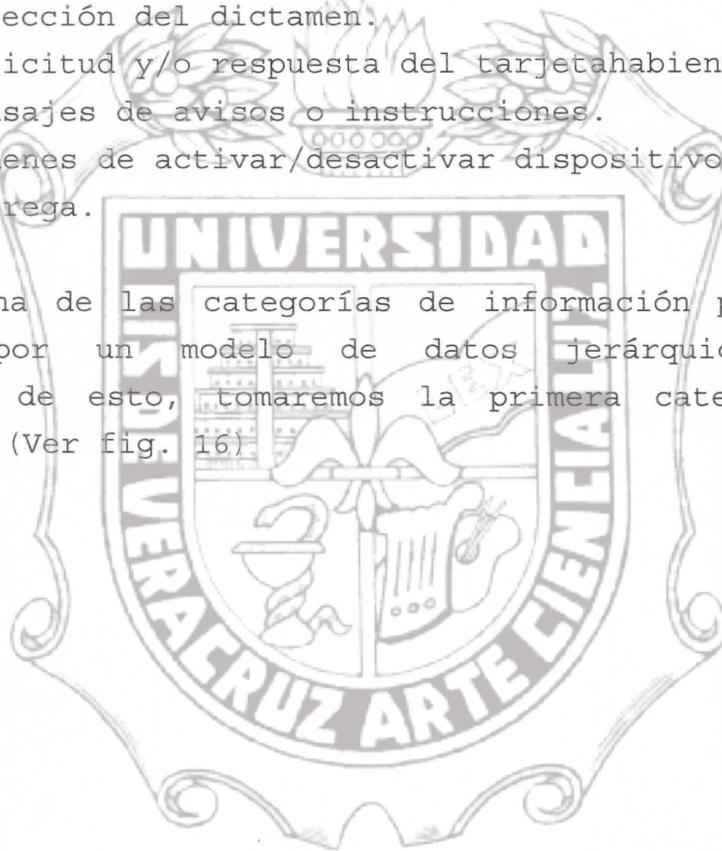
Los límites de una categoría de información están dados por los datos que representan, los cuales pueden ser claramente ubicados realizando una estructura de datos de forma jerárquica.

CASO-EJEMPLO.

Finalmente se obtienen las siguientes categorías de información, que se listan a continuación:

- 1.- Características de la tarjeta.
- 2.- Estado del sistema.
- 3.- Estado del tarjetahabiente.
- 4.- Criterios de validación.
- 5.- Selección del dictamen.
- 6.- Solicitud y/o respuesta del tarjetahabiente.
- 7.- Mensajes de avisos o instrucciones.
- 8.- Ordenes de activar/desactivar dispositivos de entrega.

Cada una de las categorías de información puede ser detallada por un modelo de datos jerárquico. Como ilustración de esto, tomaremos la primera categoría de información: (Ver fig. 16)



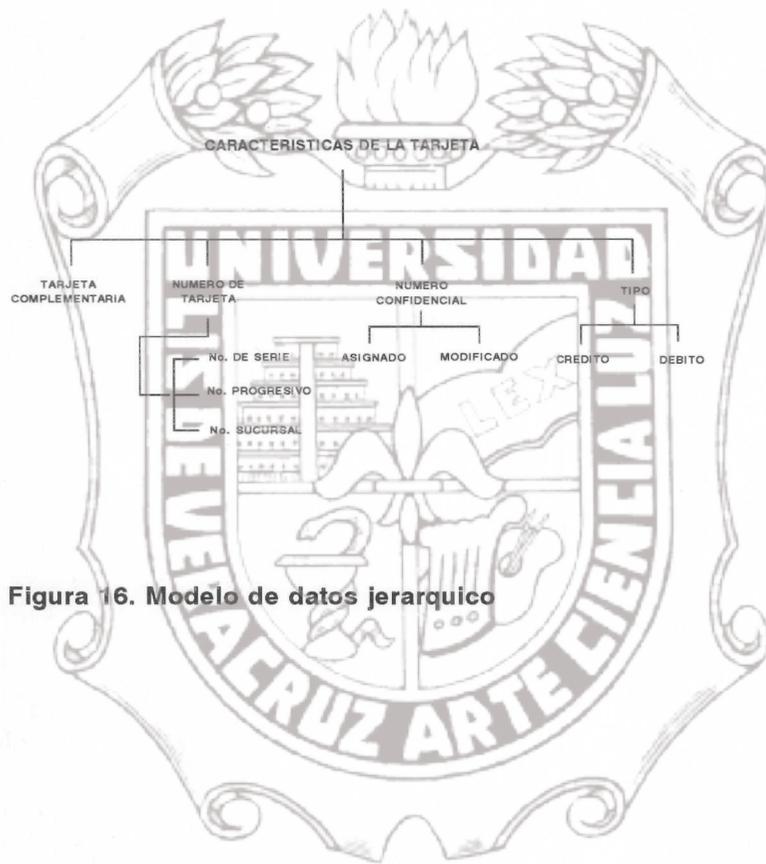


Figura 16. Modelo de datos jerarquico

6.2.4 Análisis de flujos de información entre actividades.

Con las actividades del modelo y las categorías de información, se realiza un análisis de la primera mitad de la Cruz Maltesa, donde se relacionan las actividades con las categorías (requeridas y generadas); con esto se logra una visualización clara del grado de participación y dependencia de las actividades, en función de la información.

Este análisis se centra en saber si individualmente o en conjunto, las actividades mantienen congruencia con respecto al manejo de la información, así como saber si existe alguna duplicidad o ausencia de la misma o bien si se pueden conjugar algunas categorías.

CASO-EJEMPLO.

Con la información hasta el momento se pueden determinar las zonas noreste y noroeste de la Cruz Maltesa y llevar a cabo su respectivo análisis de congruencia y consistencia entre sus elementos que la componen, como se ilustra a continuación: (Ver fig. 17)

6.2.5 Definición de procedimientos de procesamiento de información.

Definir los procedimientos de procesamiento de información (módulos, programas, rutinas o paquetes de software) que pueden ser aplicables, como apoyo a las actividades y que utilicen las categorías de información. En esta definición se pueden incluir procedimientos de proceso de información existentes y/o sistemas de procesamiento que no sean informáticos.

CASO-EJEMPLO.

Como resultado de los análisis, se logran definir cinco procedimientos de procesamiento de información informático: Toma de decisiones, Control, Operación e Histórico, para completar finalmente la Cruz Maltesa y estar en condiciones de verificar los procedimientos de procesamiento de información seleccionados e iniciar con el desarrollo de cada algoritmo para cada procedimiento definido.

6.2.6 Análisis de los procesamientos de información.

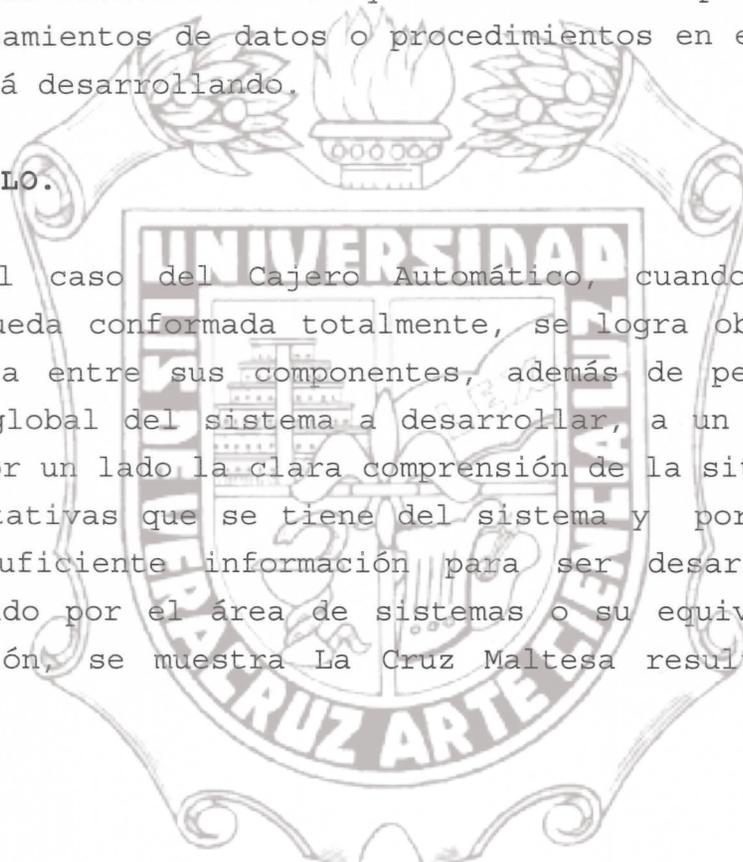
Primeramente se identifican los procedimientos de procesamiento de información definidos en el paso anterior con las categorías de información, esto es, debemos especificar qué categoría generará cada procedimiento de procesamiento de información y a su vez que categoría(s) de información utilizará para desempeñar su función.

Posteriormente, son examinadas las categorías de

información de apoyo que necesita cada actividad del modelo conceptual, poniendo especial atención a el medio por el cual se espera sea procesado, esto con la finalidad de tratar de identificar cualquier omisión o duplicación de los procesamientos de datos o procedimientos en el sistema que se está desarrollando.

CASO-EJEMPLO.

En el caso del Cajero Automático, cuando la Cruz Maltesa queda conformada totalmente, se logra observar la congruencia entre sus componentes, además de permitir un análisis global del sistema a desarrollar, a un nivel que permite por un lado la clara comprensión de la situación de las expectativas que se tiene del sistema y por el otro, con la suficiente información para ser desarrollado e implementado por el área de sistemas o su equivalente. A continuación, se muestra La Cruz Maltesa resultante. (ver fig. 18)



6.2.7 Planteamiento de algoritmos.

Para cada procedimiento de procesamiento de información nuevo o modificado, es necesario desarrollar un algoritmo de transformación de datos, haciendo resaltar su importancia e impacto en el sistema. Con este paso, se logra una definición de lo que se espera que sea desarrollado para su implementación, lo cual constituye lo mínimo necesario para hacer que los datos básicos proporcionen las categorías de información requeridos por los módulos de procesamiento de información.

6.3 Fase de diseño detallado.

En esta fase se ilustra para su aplicación, la utilización de algunos métodos de diseño detallado existentes. Se indican los pasos que contienen, resaltando la forma en que deben ser aplicados, a fin de conservar la continuidad de la metodología. Por otro lado, se plantea la necesidad de definir un plan de desarrollo del producto de software, con sus inherentes ventajas, pero haciendo hincapié en la participación del gestor del proyecto durante esta fase. Finalmente, se describen algunos atributos que debe poseer el producto de software al terminar esta fase y la forma en que el gestor participa en su obtención.

La Fase de Diseño Detallado tiene como propósito el construir el sistema de software que cubra las especificaciones de requerimientos, que a su vez contenga toda una serie de características de calidad y contemple en su diseño provisiones que garanticen la evolución que se espera del sistema a corto, mediano y largo plazo.

Es evidente que en esta fase la parte técnica tiene una gran relevancia y que deberá estar orientada y apoyada por parte del gestor del proyecto del software a fin de realizar un buen producto de software que resuelva la situación problema. La participación del analista (Ingeniero de Software) como el más alto rango próximo, responsable del proyecto del área de computación es el que deberá asegurar la integración en el producto de software de las ideas, objetivos, interés, etc. planteados por el gestor del proyecto, los cuales deberán ser un reflejo de las expectativas que tiene del sistema, desde la perspectiva de un nivel ejecutivo de la empresa. Esta fase la componen los siguientes dos apartados:

- 1.- Metodología de diseño existente. La continuidad del paradigma mediante métodos de diseño de detalle existentes.
- 2.- Plan de desarrollo del software. El seguimiento y participación en el desarrollo del producto de software por parte del gestor del proyecto.

6.3.1 Metodologías de diseño.

El diseño detallado consiste básicamente en la realización del sistema como un producto de software; es la parte técnica donde se define como deberá estar constituido el producto para poder cubrir los requerimientos del sistema, donde por lo general se realizan las actividades propias de diseño (diseño, codificación y prueba).

Existen varios métodos de diseño detallado, los cuales

se agrupan de acuerdo a: el flujo de datos, la estructura de datos y orientado a los objetos. Independientemente de la orientación del diseño que se seleccione, es de gran relevancia la información que se ha generado en la fase de diagnóstico y diseño. En la mayoría de los casos es posible suplir los primeros pasos de las metodologías de diseño detallado.

Con la finalidad de ejemplificar lo mencionado anteriormente e ilustrar la continuidad de este paradigma y su compatibilidad con los métodos de diseño detallado existentes, se analiza el caso de los métodos de Yourdon y Constantine, así como el método de Jackson.

- Método de Yourdon y Constantine.

El diseño orientado al flujo de datos propuesto por Yourdon y Constantine utiliza las características del flujo de información para derivar la estructura del programa. Donde un diagrama de flujo de datos se convierte en una estructura de programa usando una o dos técnicas de análisis. (Análisis de transformación o análisis de transacción).

Pasos:

- 1) Revisión del modelo fundamental del sistema. El modelo fundamental del sistema abarca un diagrama de flujo de datos a un nivel muy general y el soporte de la información.
- 2) Revisión y refinamiento de los diagramas de flujo de

datos para el software. Usando la información obtenida de una especificación de requerimientos del software, se deriva un diagrama de flujo de datos a un mayor nivel de detalle.

- 3) Determinar si el diagrama de flujo de dato tiene características de transformación o de transacción. En general, el flujo de información de un sistema puederepresentarse siempre como transformación. Sin embargo, cuando se encuentra una obvia característica de transacción, se recomienda un diseño diferente. En este paso, el diseñador selecciona una característica global del flujo de datos basándose en la naturaleza prevaleciente del diagrama de flujo de datos.
- 4) Aislar el centro de transformación especificando los límites del flujo de llegada y salida. Se identifican los flujos de llegada, como un camino en el que la información se transforma de externa a interna; el flujo de salida se transforma de interna a externa. Los límites del flujo de llegada y salida están abiertos a interpretación.
- 5) Realización del "primer nivel de factorización". La estructura del programa representa una distribución descendente de control. La factorización da como resultado una estructura de programa en la que los módulos de nivel superior toman las decisiones de ejecución y los modulos de nivel inferior ejecutan la mayoría de los trabajos de entrada, computacional y de salida. Los modulos de nivel intermedio ejecutan algún control y realizan moderadas cantidades de trabajo.

- 6) Ejecución del "segundo nivel de factorización". Este nivel se realiza mediante la conversión de las transformaciones individuales de un diagrama de flujo de datos, en los módulos correspondientes de la estructura del programa. Comenzando en el límite del centro de transformación y considerando las llegadas y salidas, las transformaciones se convierten en niveles subordinados de la estructura de software.

En este método, la información y el análisis que se realizan en los tres primeros pasos es similar al que se ha venido utilizando en la fase de Diagnóstico y Diseño, específicamente cuando se lleva a cabo la diagramación del diseño conceptual y el análisis de cada subsistema en una entidad de transformación, analizando la información que requiere para la misma y la información que genera, la cual es posteriormente agrupada en categorías de información para formar parte de la Cruz Maltesa.

En los pasos subsecuentes, también es útil el trabajo realizado en las dos fases anteriores pero en menor grado, mas bien como información de apoyo y referencia.

- Método de desarrollo de sistemas de Jackson.

El método de Jackson consiste en transformar una representación de la estructura de datos en una representación del software mediante la evaluación de las características de software; la representación de la estructura de las entidades se lleva a cabo mediante una notación propia, -conexiones por datos y conexiones por

vector estado-, denominada diagramas de Jackson. Estos son utilizados para hacer un modelo inicial donde se especifican las funciones que corresponden a las acciones que deberá realizar.

Pasos:

- 1) Entidad acción. De la declaración del problema se identifican las entidades (personas, objetos u organizaciones) que necesita el sistema para producir o usar información y las acciones (los sucesos que ocurren en el mundo real) que afectan a las entidades.
- 2) Estructura de entidad. Se ordenan en el tiempo, las acciones que afectan a cada entidad y se representan mediante diagramas de Jackson.
- 3) A partir del modelo inicial. Se crea un modelo de procedimientos que representa las entidades y las acciones; se definen las conexiones entre el modelo y el mundo real.
- 4) Se especifican las funciones que corresponden a las acciones definidas.
- 5) Temporización del sistema. Se establecen y especifican los aspectos de planificación del proceso.
- 6) Implementación. Se especifica el hardware y el software como un diseño.

En la metodología de Jackson, el primer paso de

entidad-acción, -donde con una breve descripción del problema en lenguaje natural-, se seleccionan las entidades (nombres) y las acciones (verbos), puede ser sustituida por el análisis de CATWDE (descrita en la Fase de Diagnóstico) donde se puede validar si la declaración inicial del problema tiene los elementos necesarios para que sirva como un adecuado punto de partida para el diseño detallado.

Así mismo el segundo paso de las funciones, cuyo propósito es expandir el diagrama de especificación del sistema, mediante la conexión, -por flujo de datos o por vectores de estado-, a un diagrama de estructura, para lo cual es necesario realizar un análisis de las actividades principales que deberá realizar el sistema, lo cual es factible mediante el estudio de las actividades sustantivas (mínimas necesarias) que se efectuó en la fase de Diagnóstico.

En el paso de modelo inicial se representan entidades y acciones, y en el cual se deben indicar las conexiones entre el modelo y el mundo real. Es esta última una de las características que mantiene el diseño conceptual del ambiente en el que se desempeña y del entorno al que pertenece el sistema que se está desarrollando.

6.3.2 Plan de desarrollo de Software.

La producción de software debe ser considerada como una actividad genérica que involucra una serie de subactividades que harán que al final, se obtenga un sistema de cómputo que satisfaga las expectativas que se tengan de él.

Como se comentó anteriormente, las actividades centrales de producción de software son el propio diseño, la codificación y prueba. Adicionalmente, se tiene que considerar que el sistema puede necesitar modificaciones, -ya sea antes o después de terminar-, por diversas razones. Con base en lo anterior, se puede decir que el diseño detallado considera y resume en un programa de cómputo, desde el conocimiento del problema hasta los planes a largo plazo que se tengan del sistema en la empresa.

Como todo administrador, el gestor del proyecto -y también el analista del sistema- debe apoyarse en un plan de desarrollo para lograr el objetivo de la realización de un sistema de cómputo de calidad, que satisfaga las necesidades para el cual fue elaborado.

En esta sección no se pretende exponer el contenido del plan ni la forma en que éste puede estructurarse. Esto dependerá de las características del proyecto, la cultura y desarrollo que tenga la empresa en el área de cómputo. Más sin embargo, es conveniente que se tenga especial atención en ciertos atributos del software, por lo que se recomienda poner mayor interés en las actividades correspondientes al plan de desarrollo, debido principalmente a su importancia en lo que se refiere a calidad y relevancia en la vida útil del sistema.

A continuación se presentan los atributos referidos y se comenta la participación por parte del gestor del proyecto en cada uno de ellos.

1) Uso de una metodología de diseño.

En la etapa de diseño se establecen las bases para el éxito y calidad del software, siendo esto el propósito de toda metodología de diseño, por lo cual el gestor del proyecto requiere solicitar al analista la especificación, justificación y uso de una metodología de diseño a fin de que la propia metodología establezca una secuencia lógica de desarrollo, mediante el conjunto de pasos que el propio método establece. Adicionalmente, se logra que el sistema pertenezca a un tipo definido de orientación de desarrollo, permitiendo aplicar técnicas, principios y herramientas compatibles a la orientación definida.

2) Ajustarse estándares determinados.

Los estándares para un producto de software deben ser previamente definidos por la empresa, por lo que en el desarrollo deberán sujetarse a ellos en caso de existir, en caso contrario, el gestor y el analista del proyecto deberán solicitar al comité técnico la definición de los estándares para su aplicación, así como su especificación en la documentación del software.

Es importante considerar que los estándares adoptados o definidos sean acordes a la proyección que pueda tener el sistema, aunado al tipo de sistema de software de que se trate, es posible usar desde estándares personales del grupo hasta necesitar cumplir con estándares internacionales de desarrollo.

3) Acreditar un plan de pruebas.

Las pruebas de un producto de software son básicamente, una revisión final de las especificaciones del diseño y la codificación. Dependiendo del propósito de la prueba, esta puede ser clasificada como: de verificación, de validación o del sistema.

La prueba de verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La prueba de validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. La prueba del sistema está constituida por una serie de pruebas diferentes, cuyo propósito es ejercitar profundamente el sistema desarrollado cuando es incorporado a su ambiente o integrado a un sistema mayor.

El gestor del proyecto participa en las pruebas de verificación mediante las pruebas de integridad del sistema que son pruebas llamadas de caja negra porque parten del principio de que conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguran que la operación interna se ajusta a las especificaciones del sistema. Por otro lado las pruebas de unidad son atendidas por el analista.

En las pruebas de verificación, el gestor participa con el analista en la revisión de la configuración (documentación del usuario, documentación de diseño,

listados, etc.); y en las pruebas Alfa, por ser pruebas que se llevan a cabo con el cliente en las instalaciones para las que se desarrolla el sistema, tiene como propósito descubrir fallas que el usuario final puede identificar.

Las denominadas pruebas del sistema caen fuera del alcance del proceso de Ingeniería de Software, como lo son las pruebas de seguridad, recuperación y resistencia. El resultado de estas pruebas en gran parte dependen del diseño del sistema, por lo que se debe de considerar o simular este tipo de pruebas durante el diseño.

4) Superar las revisiones técnicas.

La revisión técnica es una actividad que permite garantizar la calidad del software en desarrollo mediante una serie de revisiones y/o inspecciones que son llevadas a cabo por personal técnico, que primeramente confirman la traducción de los requisitos al diseño mediante el análisis de la arquitectura del software; posteriormente revisan la lógica procedimental de los algoritmos y como están implementados en el programa, finalmente revisan el código.

En este aspecto, el gestor debe solicitar la conformación de un grupo técnico para que realice la revisión técnica del software, la cual produzca un reporte donde se indique desde el objeto de la revisión hasta las conclusiones y recomendaciones. Estas últimas deberán ser apoyadas por el analista y el gestor para su aplicación.

5) Facilitar la aplicación de mantenimiento.

El mantenimiento del software se refiere a toda modificación que se lleva a cabo después de la fase de prueba, esto es después de que el sistema se ha implementado.

La razón por la cual se requiere la modificación determina el tipo de mantenimiento a efectuar, dando lugar a cuatro tipos de mantenimiento:

Correctivo: Cuando el propósito es eliminar los errores presentados durante el uso del sistema.

Adaptativo: Cuando se pretende mantener vigente el uso del sistema a pesar de los cambios en su ambiente (Sistema operativo, Hardware, Interfase, etc).

Perfectivo: Es aquel que se realiza con la finalidad de que el sistema realice nuevas funciones o mejorar las existentes.

Preventivo: Es aquel que se lleva a cabo con la finalidad de modificar el sistema, para facilitar la realización de los tres tipos de mantenimientos mencionados anteriormente.

La participación del gestor en la fase de diseño detallado consiste en propiciar que mediante la pruebas del sistema, se minimice la necesidad de aplicar mantenimiento correctivo, además de proporcionar la información relevante que permita dar a conocer al analista la posibilidad de necesitar aplicar a mediano y largo plazo el mantenimiento correctivo o adaptativo, a fin de realizar un diseño que facilite el mantenimiento, es decir que el mantenimiento sea contemplado como parte del diseño.

Conocidas cada una de las fases del paradigma propuesto se pueden resumir como lo muestra la figura No.19, donde se puede observar algunas de las características de la propia metodología en su conjunto, como es la facilidad de continuidad de una fase a otra, la simplicidad natural de la secuencia de las fases y la cobertura que tiene, que abarca desde una situación problemática no estructurada hasta la compatibilidad con el uso de técnicas de desarrollo de otras metodologías.

Finalmente es importante observar que la aportación principal de la metodología esta en la forma en que aborda el problema y desarrolla sus fases mediante un enfoque sistémico compatible con la Ingeniería de Software.

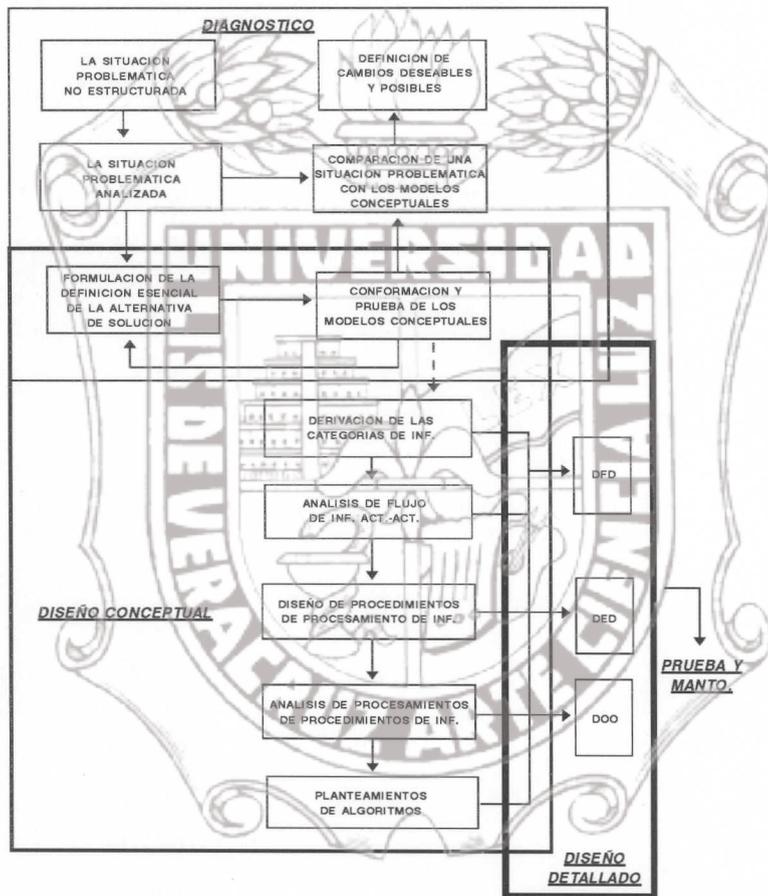


Figura 19. Modelo propuesto

VII. CONCLUSIONES

Las conclusiones que se presentan a continuación son el resultado de la conjugación de la experiencia acumulada obtenida durante la elaboración de la presente tesis y de la información consultada de diversas fuentes del área de cómputo, especialmente los relativos a Ingeniería de Software:

- 1.- Durante el ciclo de desarrollo de cualquier producto de software a través de sus fases, es factible y deseable la participación del gestor del proyecto, debido principalmente a la información adicional que puede aportar para la adecuada toma de decisiones, el punto de vista que debe tener y conocer muy de cerca las expectativas informales que la empresa pueda tener del proyecto.
- 2.- La aplicación de principios y técnicas de Teoría General de Sistemas, han demostrado ser útiles para un adecuado desarrollo de software, principalmente en las primeras fases; también se han encontrado similitudes y compatibilidad en el uso de métodos de diseño de desarrollo existentes.
- 3.- Los pasos generales del ciclo de vida del desarrollo de software (planeación, análisis, diseño y mantenimiento) continúan vigentes en todos los paradigmas, donde la diferencia que existe entre ellos estriba en la forma de agrupar las actividades en fases y en la forma en que se logra pasar entre cada

una de ellas, así como la preferencia de controlar, evaluar o bien desarrollar en mayor o menor medida, determinado concepto (riesgo, calidad, funcionalidad, etc.).

- 4.- La tendencia actual de desarrollo de sistemas complejos y de mayor magnitud, donde los objetivos pueden ser múltiples, ponen de manifiesto la estrecha dependencia que el gestor del proyecto y el analista del sistema deben de tener y de la cual en gran medida, depende el éxito del proyecto, resaltando su importancia en la toma de decisiones para canalizar los esfuerzos hacia dirección correcta.
- 5.- Después de considerar los principios básicos de diseño y la aplicación de heurísticas, los nuevos métodos tratan de facilitar la inclusión de conceptos de mantenimiento y calidad, donde estos últimos toman cada vez mas importancia, teniendo la característica de ser conceptos donde su importancia está en función de la concepción que tenga de ellos la empresa y que están implícitos en las políticas de desarrollo de software; por lo que resulta de gran importancia, la participación activa y directa del gestor del proyecto, así como la utilización de un paradigma que lo permita.

Finalmente, el paradigma de desarrollo de software propuesto en esta tesis conjunta una serie de elementos básicos con un enfoque holístico, logrando facilitar el desarrollo de productos de software y propiciando una responsable participación del gestor del proyecto.

VIII. BIBLIOGRAFIA

- Bertalanffy L. V., Teoría General de los Sistemas, Fondo de Cultura Económica, 1976.
- Blanchard S. B., Administración de Ingeniería de Sistemas, Limusa. 1993.
- Boehm W. B., Software Engineering Economics, Prentice-Hall, 1981.
- Cárdenas M. A., La Ingeniería de Sistemas filosofía y Técnicas, Limusa, 1978.
- Checkland P., Systems Thinking; Systems Practice, John Wiley Son, 1981.
- Wilson B., Systems: Concepts, Methodologies, and Applications, John Wiley Sons, 1990.
- Del Poso, N. F., La Dirección por Sistemas, Limusa, 1979.
- Fairley, E. R., Ingeniería de Software, McGraw-Hill, 1988.
- Gigch Van John P., Teoría General de Sistemas Aplicadas, Trillas, 1981. Gonzalo Cuevas Agustín, Addison-Wesley Iberoamericana, 1993.
- Jones, W. G., Software Engineering, John Wiley Sons, 1990.
- Macro Allen, Software Engineering, Prentice Hall, 1990.
- McDermid C. Donald, Software Engineering for Information Systems, Blackwell Scientific Publications, 1990.

Ramamoorthy V. C. and Vick C.R., Handbook of Software Engineering, Van Nostrand Reinhold Company, 1994.

Pressman S. Roger, Ingeniería del Software un enfoque práctico, McGraw-Hill, 1993.

Senn A. James, Análisis y Diseño de Sistemas de Información, McGraw-Hill, 1988.

Nommerville Ian, Software Engineering, Addison-Wesley, 1992.

Yourdon Edward and Constantine L. Larry, Structured Design, Yourdon Press, 1990.

Yourdon Edward, Análisis Estructurado Moderno, Prentice-Hall Hispanoamericana, 1993.



IX. GLOSARIO

CATWDE: mnemónico de las seis características cruciales que se deben incluir en una definición esencial.

Clima : Una característica de una situación del problema; es la relación entre los elementos de estructura y los elementos de proceso de esta.

Cruz Maltesa: Matriz de cuatro partes utilizada para analizar un modelo y los procedimientos de información requeridos.

Definición Esencial: Una descripción concisa y construida con precisión de un sistema de actividad humana que enuncia lo que es el sistema.

Enfoque de sistemas : Un enfoque que predica resolver los los problemas del sistema mayor, con soluciones que satisfacen no solo a los objetivos de los subsistemas, sino también la sobrevivencia del subsistema global.

Estado del Arte: Situación actual mas avanzada en la que se encuentra determinado genero.

Gestores: Persona encargada de la acción y efecto de administrar.

Hardware: Componentes físicos y permanentes de una computadora o de un sistema de proceso de datos.

Ingeniería de Software: Conjunto de actividades, métodos y practicas que guían a las personas en la producción de software.

Interfase: Dispositivo electrónico que permite la comunicación entre equipo o entre otra parte del mismo.

Modelo : Una construcción intelectual y descriptiva de una entidad en la cual al menos un observador tiene un interés.

Modelo conceptual: Una descripción sistémica de un sistema de actividad humana, construido sobre la base de una definición esencial.

Paradigma: Un proceso, un procedimiento, que puede utilizarse en forma repetida para abordar un tipo específico de problema.

Software: Conjunto de programas utilizables en una clase de computadoras.

Teoría General de Sistemas: Disciplina que proporciona fundamento y apoyos teóricos al enfoque de sistemas.

Weltanschauung: La forma en la cual un autor de decisión ve la totalidad de un problema.