

28  
21



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

---

---

FACULTAD DE INGENIERIA

PROPUESTA DE POLITICAS NORMAS Y  
PROCEDIMIENTOS PARA LA ELABORACION  
DE SISTEMAS DE INFORMACION CON  
ORIENTACION A OBJETOS BAJO  
LA NORMA ISO 9000

**TESIS PROFESIONAL**  
QUE PARA OBTENER EL TITULO DE  
**INGENIERO EN COMPUTACION**  
P R E S E N T A N :  
**GUSTAVO ADOLFO DE LA CRUZ GARCES**  
**ANGEL CESAR GOVANTES SALDIVAR**



DIRECTOR: ING. HERIBERTO OLGUIN ROMO

MEXICO, D. F.

1997

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

## ***Agradecimientos***

Agradecemos a la Universidad Nacional Autónoma de México, *UNAM*, el habernos permitido formar parte de ella, ya que gracias a la *UNAM*, muchos profesionistas han podido cumplir una meta más en sus vidas.

A la *Facultad de Ingeniería* ofrecemos nuestras más sinceras gracias, primeramente por la formación, conocimientos, experiencias, y apoyos recibidos durante nuestra estancia. A su vez agradecemos a todos y cada uno de los profesores que nos han formado, legándonos sus conocimientos y enseñanzas, y que han dejado huella en nosotros; especialmente a nuestro director de tesis el *Ing. Heriberto Olguín Romo* por su dirección, paciencia, experiencias y conocimientos compartidos durante la elaboración del presente trabajo.

*Al Programa de Tecnología en Computación*  
ya que bajo su compromiso y orgullo de pertenecer, me impulso siempre a dar más de mi en todos aspectos. Por su concepto de innovación al apoyo académico que han incrementado mi formación.

A mis amigos y compañeros por su retroalimentación de conocimientos.

*Angel César Govantes Saldivar*

*Al Instituto de Ingeniería* por la oportunidad de permitirme formar parte de éste, aplicando y desarrollando mis conocimientos aprendidos en la carrera, y por la experiencia adquirida.

A mis amigos y compañeros por su retroalimentación de conocimientos.

*Gustavo Adolfo de la Cruz Garcés*

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**PROPUESTA DE POLÍTICAS, NORMAS Y PROCEDIMIENTOS  
PARA LA ELABORACIÓN DE SISTEMAS DE INFORMACIÓN  
CON ORIENTACIÓN A OBJETOS; BAJO LA NORMA ISO 9000**

**GUSTAVO ADOLFO DE LA CRUZ GARCÉS  
ANGEL CESAR GOVANTES SALDIVAR**

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>3</b>
---------------------------	----------

## **CAPÍTULO I**

<b>CONCEPTOS SOBRE POLÍTICAS, NORMAS Y PROCEDIMIENTOS .....</b>	<b>7</b>
---	----------

<b>Introducción .....</b>	<b>9</b>
<b>1. Definiciones fundamentales .....</b>	<b>10</b>
<b>2. Política y su entorno .....</b>	<b>11</b>
<b>3. Proceso para la generación de políticas .....</b>	<b>14</b>
<b>4. Importancia de las normas .....</b>	<b>16</b>
<b>5. Organizaciones desarrolladoras de sistemas .....</b>	<b>17</b>
<b>6. Beneficios de políticas, normas y procedimientos .....</b>	<b>18</b>

## **CAPÍTULO II**

<b>NORMA ISO 9000 .....</b>	<b>21</b>
-----------------------------	-----------

<b>1. Orígenes de la Norma ISO 9000 .....</b>	<b>23</b>
<b>2. Objetivos de la Norma ISO .....</b>	<b>26</b>
<b>3. Estructura original de la serie ISO 9000 .....</b>	<b>29</b>
<b>4. Panorama y aplicación de las Normas .....</b>	<b>31</b>
<b>5. Principales funciones a cumplir de ISO .....</b>	<b>34</b>
<b>6. Principios de desarrollo de la Norma ISO .....</b>	<b>34</b>
<b>7. Beneficios de ISO 9000 .....</b>	<b>36</b>
<b>8. Conceptos elementales de calidad .....</b>	<b>38</b>
<b>9. El software en la Norma ISO 9000 .....</b>	<b>41</b>

**CAPÍTULO III**

<b>SISTEMAS DE INFORMACIÓN .....</b>	<b>47</b>
<b>Introducción .....</b>	<b>49</b>
<b>1. Perspectiva de los sistemas de información .....</b>	<b>51</b>
<b>2. Conceptos de sistemas de información .....</b>	<b>52</b>
<b>3. Objetivos de los sistemas de información .....</b>	<b>54</b>
<b>4. Fundamentos de los sistemas de información .....</b>	<b>55</b>
<b>4.1 Recursos utilizados por un sistema de información .....</b>	<b>57</b>
4.1.1 Recursos de hardware .....	58
4.1.2 Recursos de software .....	58
4.1.3 Recursos humanos .....	60
4.1.4 Recursos y manejo de información .....	60
<b>5. Puntos a destacar dentro de un sistema de información .....</b>	<b>65</b>
<b>6. Categorías de los sistemas de información .....</b>	<b>66</b>
6.1 Sistemas de información para el procesamiento de transacciones .....	67
6.2 Sistemas de información para el proceso administrativo .....	69
6.3 Sistemas de información para el soporte de decisiones .....	70
6.4 Sistemas de información ejecutivos .....	71
<b>7. Técnicas de sistemas de información .....</b>	<b>72</b>
7.1 Método del ciclo de vida para el desarrollo de sistemas .....	72
7.2 Método del prototipo de sistemas .....	75

**CAPÍTULO IV**

<b>ORIENTACIÓN A OBJETOS .....</b>	<b>79</b>
<b>Introducción .....</b>	<b>81</b>
<b>1. El por qué del término orientación a objetos .....</b>	<b>83</b>
<b>2. Metodología tradicional vs. Metodología con orientación a objetos .....</b>	<b>83</b>
<b>3. Principios de la orientación a objetos .....</b>	<b>87</b>
3.1 Objeto .....	88
3.2 Mensajes .....	92
3.3 Clase .....	95
<b>4. El modelado en la orientación a objetos .....</b>	<b>98</b>
<b>5. Perspectivas a considerar en el desarrollo con orientación a objetos...</b>	<b>100</b>
<b>6. Características y beneficios de las técnicas de orientación a objetos...</b>	<b>101</b>

**CAPÍTULO V**

<b>PROPUESTA DE POLÍTICAS, NORMAS Y PROCEDIMIENTOS PARA LA ELABORACIÓN DE SISTEMAS DE INFORMACIÓN CON ORIENTACIÓN A OBJETOS, BAJO LA NORMA ISO 9000</b>	<b>105</b>
1. Objetivo .....	107
2. Introducción .....	107
3. Políticas para la elaboración de sistemas .....	108
3.1 Objetivo .....	108
3.2 Políticas .....	108
4. Normas para la elaboración de sistemas .....	113
4.1 Objetivo .....	113
4.2 Normas generales .....	113
4.3 Normas para el análisis de sistemas .....	116
4.4 Normas para el diseño de sistemas .....	118
4.5 Normas para la programación y documentación de sistemas .....	122
4.6 Normas para la realización de pruebas .....	124
4.7 Normas para la implantación de sistemas y capacitación .....	125
4.8 Normas para el mantenimiento de sistemas .....	126
5. Manual de procedimientos para el desarrollo de sistemas con orientación a objetos (MPDSOO) .....	127
5.1 Introducción .....	127
5.2 Objetivo del MPDSOO .....	127
5.3 Plan de calidad para el desarrollo de sistemas .....	127
5.4 Etapas para el desarrollo de sistemas .....	129
5.5 Modelado con orientación a objetos .....	129
6. Análisis de sistemas .....	131
6.1 Objetivo .....	131
6.2 Descripción .....	131
6.3 Actividades del análisis .....	132
6.3.1. Estudio de factibilidad .....	132
6.3.2. Establecimiento del plan para el desarrollo de un sistema .....	133
6.3.3. Calendario de trabajo .....	135
6.3.4. Relación de documentos .....	135
6.3.5. Modelado en orientación a objetos .....	135
6.3.6. Acta de acuerdos .....	141
7. Diseño de sistemas .....	142
7.1 Objetivo .....	142
7.2 Descripción .....	142
7.3 Actividades del diseño .....	143

7.3.1. Selección e identificación de las herramientas de desarrollo y del hardware .....	143
7.3.2. Diagrama entidad-relación .....	144
7.3.3. Diseño de las bases de datos y archivos .....	144
7.3.4. Definición de procesos .....	145
7.3.5. Diseño de entradas y salidas .....	145
7.3.6. Diagrama estructural del sistema .....	146
7.3.7. Combinación de los modelos .....	146
7.3.8. Importancia de la selección de algoritmos .....	146
7.3.9. Desarrollo de la herencia entre clases .....	147
8. Programación y documentación de sistemas .....	149
8.1 Objetivo .....	149
8.2 Descripción .....	149
8.3 Actividades de la programación y documentación de sistemas .....	149
8.3.1. Integración de bibliotecas de funciones .....	149
8.3.2. Programación de módulos .....	150
8.3.3. Manual técnico .....	150
8.3.4. Manual del usuario .....	150
9. Pruebas de sistemas .....	151
9.1 Objetivo .....	151
9.2 Descripción .....	151
9.3 Actividades y tipos de pruebas .....	152
9.3.1. Pruebas con datos reales .....	152
9.3.2. Corridas en paralelo .....	152
9.3.3. Pruebas funcionales .....	153
9.3.4. Pruebas de desempeño .....	153
9.3.5. Pruebas de tensión .....	153
9.3.6. Pruebas de unidad .....	153
9.3.7. Pruebas de integración .....	154
9.3.8. Pruebas de sistema .....	154
9.3.9. Pruebas independientes .....	154
9.3.9. Pruebas beta .....	155
10. Implantación y capacitación .....	158
10.1 Objetivo .....	158
10.2 Descripción .....	158
10.3 Actividades para la implantación y capacitación .....	158
10.3.1. Configuración del desempeño .....	158
10.3.2. Registro de la implantación .....	158
10.3.3. Cursos de capacitación .....	157
10.3.4. Niveles de capacitación .....	157
10.3.5. Integración de sistemas al banco institucional de sistemas .....	157
10.3.6. Capacitación .....	158

<b>11. Mantenimiento de sistemas</b> .....	<b>159</b>
11.1 Objetivo .....	159
11.2 Descripción .....	159
11.3 Actividades para el mantenimiento del sistema .....	161
11.3.1. Identificación del tipo de mantenimiento .....	161
11.3.2. Solicitud de modificación .....	163
11.3.3. Análisis de la solicitud de modificación .....	163
11.3.4. Aprobación de la solicitud y elaboración de un programa de trabajo .....	164
<b>12. Tabla de formas o formatos para la documentación de sistemas</b> .....	<b>165</b>
<b>13. Formas o formatos para la documentación de sistemas</b> .....	<b>168</b>

## CAPÍTULO VI

<b>CONCLUSIONES</b> .....	<b>233</b>
Glosario de términos .....	237

## CAPÍTULO VII

### APÉNDICES

Apéndice A: Metodologías para el desarrollo de sistemas con orientación a objetos .....	A - 1
Apéndice B: Puntos a considerar para la elaboración de programas entendibles .....	B - 1
Apéndice Norma ISO .....	I - 1

### BIBLIOGRAFÍA

## INTRODUCCIÓN

## **INTRODUCCIÓN**

La información es uno de los recursos más valiosos de cualquier institución o empresa, de ahí la necesidad de obtenerla de manera rápida y sin perder la objetividad de la misma; a partir de esta necesidad se ha dado la tendencia a la informatización de empresas e instituciones. Bajo este contexto, tanto el software como el hardware se han dispuesto de forma tal que puedan organizar dichos datos, los cuales una vez ordenados conformarán información que puede ser crucial para la toma de decisiones.

Por otra parte si recordamos las proyecciones de aumento de población a nivel mundial, se espera que al llegar al próximo milenio existan aproximadamente 6230 millones de personas, las cuales representarán y generaran más información, que tanto empresas como gobiernos se darán a la tarea de organizar y mantener al día, con el apoyo indudable de los sistemas de información basados en cómputo.

En consecuencia se espera una gran tarea de desarrollo de sistemas tanto para programadores, analistas y líderes de proyecto. Que se verán en la necesidad de realizar estudios de análisis, diseño y programación, lo que generará en cientos de miles de líneas de código. Estos estudios sin la organización adecuada pueden resultar estratosféricamente complicados en tamaño, estructura y comprensión. Afortunadamente ya se han dado algunas soluciones para este problema, tal es el caso de la programación orientada a objetos, la cual se perfila como fuerte candidata para dar solución al gigantismo de los programas tradicionales.

En resumidas cuentas, se proyecta para la próxima década la elaboración de un gran número de sistemas de información, tanto integrales como aislados, que no sólo alcancen a cubrir la gran demanda, sino que lo hagan de manera eficiente y que además sean realizados en tiempos razonables. Por otra parte dichos sistemas se encaminarán a ser fundamentales para la organización de las sociedades, y representarán una dependencia en mayor o menor grado para el actuar de la sociedad misma, razón por la cual se acentúa el aspecto de confiabilidad.

Con las perspectivas anteriormente mencionadas, lo que proponemos en el presente documento, es apoyar en varios frentes el desarrollo de sistemas, tales como la organización interna del personal en cuanto a la elaboración del sistema, el aspecto de programación en torno a la orientación a objetos como solución al gigantismo de los programas, y el aspecto de calidad. Para esto último hemos considerado a la norma ISO-9000 como parámetro general en cuanto al desarrollo de un producto, para nuestro caso específico el software. El conjunto de normas ISO-9000 son de carácter complementario y de aplicación general para cualquier tipo de producción, y en este documento se les ha tomado como parámetros auxiliares de normatividad, considerando el trato especial que implica el desarrollo de software en diferencias y semejanzas.

El presente documento comprende cinco partes. En el capítulo uno presentamos las nociones fundamentales de lo que son las políticas, normas y procedimientos, para situar al lector en el contexto al cual nos queremos referir, incluyendo lo concerniente a responsabilidades de personal y de dirección, así como el proceso de generación de políticas, y por supuesto su importancia y beneficios. A continuación en el capítulo dos, nos dedicamos a la tarea de explicar en qué consiste la norma ISO-9000 en cuanto a orígenes, objetivos, estructura que guarda y sus beneficios. En el capítulo tres exponemos los principios, conceptos, objetivos, etc. de los sistemas de información y las categorías de los mismos. En el capítulo cuatro explicamos los conceptos básicos concernientes a la orientación a objetos, describimos las definiciones fundamentales en las que se basa dicha teoría en cuanto al análisis y diseño. El capítulo cinco: *Propuesta de políticas, normas y procedimientos para la elaboración de sistemas de información, con orientación a objetos, bajo la norma ISO 9000*; conforma el objetivo y aportación principal en el presente trabajo, apoyado por un conjunto de referencias a la norma ISO 9000 que se disponen al final de cada uno de los puntos expuestos, tanto en políticas como en normas y de la misma forma en la parte de procedimientos en cuanto a su descripción y actividades, así como los formatos requeridos para la documentación de cualquier sistema de información. Finalmente, realizamos las conclusiones y discernimientos de este trabajo.

Con objeto de suslentar algunos puntos concernientes al presente documento, hemos dispuesto al final del mismo un conjunto de apéndices que refuerzan y profundizan algunos de los conceptos expuestos.

Deseamos mencionar también que uno de los objetivos principales de este documento, es el de lograr una aportación para los diseñadores de sistemas de información cuya vigencia sea lo más prolongada posible. A diferencia de un sistema de información que puede ser perecedero debido al tipo de lenguaje o plataforma empleada; se ha buscado que el presente documento sea aplicable a cualquier desarrollo de sistemas sin que se vea frenado o entorpecido por cuestiones tecnológicas de hardware y software, pero, que tenga a bien consideráralas dependiendo de la institución o empresa.

Así pues bajo la justificación descrita en cuanto a problemática, consideraciones y estructura de este trabajo, esperamos que nuestra investigación y aportaciones se vean redituables en sistemas confiables y con menor tiempo de desarrollo para cualquier institución o empresa.

Angel Cesar Govantes Saldivar  
Gustavo Adolfo de la Cruz Garcés

Ciudad Universitaria, Julio 1997



**CONCEPTOS SOBRE POLÍTICAS NORMAS  
Y PROCEDIMIENTOS**

*Definir nuestro ideal es crear nuestro futuro.*



## **CONCEPTOS SOBRE POLÍTICAS NORMAS Y PROCEDIMIENTOS**

### **Introducción**

Después de dos décadas de incumplimiento a las promesas de productividad y metas de calidad para la aplicación de tecnología y nuevas metodologías en el software, las industrias y las organizaciones gubernamentales reconocen como uno de los problemas que les aqueja, su falta de habilidad para administrar el proceso de elaboración de software. Ni los beneficios de mejores métodos y herramientas pueden colaborar enteramente a mejorar el proceso de desarrollo de software, si éste se caracteriza por ser un proceso de desarrollo hecho a la medida, indisciplinado y caótico durante la elaboración de cualquier sistema.

De hecho, en una gran diversidad de organizaciones los proyectos de elaboración de sistemas son entregados a menudo con cierto retraso y el presupuesto estimado es comúnmente incrementado en más de lo previsto con anterioridad, se sabe que en la mayoría de las veces los proyectos se terminan debido a jornadas de trabajo prolongadas por parte del personal que labora en el desarrollo de software, más que por la ejecución de los métodos provistos por la propia organización que debería contar con un proceso establecido para el desarrollo de sistemas. La importancia radica en que ocurre muy frecuentemente, que las organizaciones no cuentan con medios organizacionales, de infraestructura y soporte necesario para ayudar a evitar o disminuir los problemas al elaborar cualquier sistema; de ahí la esencia en que las organizaciones cuenten e implanten políticas, normas y procedimientos para el desarrollo de sistemas.

Para la elaboración de sistemas en ausencia de un proceso establecido dentro de una organización, los resultados esperados dependen enteramente de la disponibilidad, participación y dedicación del mismo personal para el siguiente proyecto. Frecuentemente se sabe que se tiene éxito si se cuenta con la disponibilidad de individuos en específico, no importando contar con una base de

productividad a largo plazo y de mejoramiento de calidad en la organización. Un proceso de mejoramiento continuo sólo puede ocurrir si se enfocan y sustentan conjuntamente tanto la ingeniería del software como el empleo de prácticas administrativas durante el proceso, encaminados precisamente en la construcción de la infraestructura de un proceso para la elaboración de sistemas.

## **1. Definiciones fundamentales**

El criterio general que debe ser considerado por cualquier institución o empresa para guiar favorablemente el desarrollo de sistemas, se basa inicialmente en definir y comprender claramente los siguientes conceptos, la difusión de los mismos es recomendable en todas las áreas que constituyen la organización.

### ***Políticas***

Orientaciones o directrices que rigen la actuación de una organización en un campo determinado. Son leyes y reglamentos que gobiernan las operaciones, se consideran como actividades a seguir y cumplir por los integrantes de una institución o empresa.

### ***Normas***

Establecimiento de reglas o patrones uniformes a seguir durante el desarrollo de una actividad. Estas deben llevarse a cabo en las actividades, actos u operaciones de una institución o empresa. Son criterios previamente aceptados para aplicarlos a un producto final.

### ***Procedimientos***

Método de ejecutar algunas actividades. Modo de estructurar en la realidad algunas actividades guardando un cierto orden preestablecido, es decir, llevar a cabo las actividades en un orden sistemático; describiendo las instrucciones paso por paso con el fin de implantarlas en un proceso, a modo de obtener seguridad y eficiencia en la realización.

## **2. Política y su entorno**

La palabra *Política* como término o definición, ha sido usado de manera tan indiscriminada que hoy en día no se tiene un significado único. En la administración el término "política" se refiere a distintas ideas de carácter amplio y en ocasiones mutuamente incompatibles como puede ser el caso de una aspiración o un principio ético, un plan a mediano y largo plazo, una decisión organizacional, una práctica, un procedimiento administrativo de carácter general, una orden establecida, o bien, una regla genérica. Pero en general coinciden en que constituyen una orden indirecta por medio de la cual los niveles superiores de la organización tratan de determinar la conducta de los niveles inferiores. Se concluye, políticas son la expresión de acuerdos que sirven de guía y canalización de los razonamientos, decisiones y acciones de la gestión hacia la consecución de los objetivos de la empresa.

Sin embargo puede usarse el término política para describir una clase especial de decisiones y directivas. Una declaración de política puede invitar a la participación y al uso de la iniciativa individual de los miembros responsables de la organización.

Una política es una guía para la realización de acciones, sin grandes restricciones. De hecho, por medio del establecimiento de políticas se siente la influencia de las actitudes de la dirección para con toda la organización; es una forma de permitir al personal directivo de la organización tomar decisiones en cualquier momento ante una determinada situación, compensando así la imposibilidad en que se halla la dirección para tratar con cada empleado.

Algunos de los principios específicos para la declaración de políticas pueden tener las siguientes características:

- Afirmar un principio general, un objetivo o una meta a cumplir de la empresa, esta característica integra una decisión o una directiva general con una filosofía de administración establecida.

- **Comprometer a los representantes de la dirección a tomar en cuenta los principios establecidos. Esta característica persigue la consistencia de la conducta entre las diferentes partes de la organización, sin perder uniformidad.**
- **Invitar a la discreción, dentro de límites apropiados, por parte de aquellos que interpretan una decisión sobre políticas. Esta característica, respalda el uso del juicio para interpretar una decisión de política en una situación específica, diferenciándola del cumplimiento de una regla inflexible.**
- **Ser congruente con otras políticas de la empresa, así como con la elaboración de procedimientos con el conjunto necesario de reglas.**
- **Ser aplicable en alguna parte de la organización teniendo una serie de beneficios definidos.**
- **Debe contar con una responsabilidad definida, alguna persona o grupo de personas que se encarguen de la ejecución de la política establecida.**
- **Debe ser estable, teniendo una consistencia y fundamentación; sólo evitando frecuentes cambios de criterio existirá una orientación definida.**
- **Debe ser flexible para adaptarse a nuevas necesidades que surjan y que demande su modificación.**
- **Debe ser inteligible para evitar las malas interpretaciones causadas por las diferencias individuales de los integrantes de una organización.**
- **Debe ser práctica para el personal de la organización, ya que se necesita advertir que la política establecida responde a las necesidades que están viviéndose.**

**Las responsabilidades del personal son:**

Efectuar la realización del principio general de la política, así como de las provisiones detalladas de la misma.

Dar a conocer tanto a supervisores representantes y personal en general, sobre la política vigente, así como persuadir al resto del personal a seguirla.

Mantener el registro de datos objetivos con el fin de apoyar la aplicación de una política, a través de estadísticas.

Mantener comunicación con la dirección de la empresa o en su defecto con el encargado de la aplicación de la política, para resolver preguntas y problemas relacionados con la aplicación de esa política.

**Las responsabilidades de la dirección serán:**

Informar, a los subordinados inmediatos, sobre las leyes, reglamentos o normas que son compatibles con las políticas de la empresa.

Recomendar prácticas nuevas o mejoradas que puedan tomarse en cuenta para la ampliación de las políticas.

Obtener las conformidades evidentes que se obtuvieron en la aplicación de esa política.

En los casos que algunas áreas de la empresa tengan establecidas prácticas efectuadas por largo tiempo, que se han establecido sin la fuerza de un procedimiento escrito, la dirección local procederá hacia la ejecución de la política en forma gradual y según su criterio, y de forma inmediata si dicha costumbre se opone a la política general de la empresa.

Se debe hacer énfasis, en que una política a diferencia de una regla, permite una discreción en cuanto a su aplicación.

### **3. Proceso para la generación de políticas**

Aunque las políticas pueden ser resultado de una serie de estudios previos o bien surgir como fruto de la experiencia, también pueden surgir como resultado de la participación del personal de la empresa, así como la aplicación de nuevos estudios e investigaciones de la rama concerniente, para nuestro caso la ingeniería de software.

Para el caso de la participación del personal pueden enumerarse una serie de fases para la creación de una política.

*Iniciación del pensamiento sobre una política.* La sugerencia de que es necesario establecer o modificar una política, puede aparecer desde cualquier punto de la organización.

*Determinación de los hechos y análisis situacional.* En esta etapa cualquier miembro de la organización es capaz de hacer una contribución útil, derivada de su experiencia directa y especializada. Este conjunto de experiencias puede ser sintetizado posteriormente por medio de un comité, que se encargará de evaluar la viabilidad de la nueva política en cuanto a actitud, recursos, beneficios, etc..

*Hacer recomendaciones al jefe ejecutivo.* Los miembros del comité reportarán al jefe sus juicios y descubrimientos en cuanto a la nueva política que ejerce en la empresa. La libre discusión estará en función al grado de confianza entre el ejecutivo y el comité.

*Preparación de una declaración escrita sobre una política.* Hasta que una decisión sobre políticas haya sido puesta por escrito, no puede ser considerada como un compromiso firme y oficial de la dirección. Para este efecto, se requiere establecer apropiadamente un texto de fácil comprensión que sintetice los principios que

conforman la política y su acción. Es conveniente que el departamento más afectado por la nueva política sea el que prepare el borrador por escrito.

***Discusión de una declaración de política propuesta.*** En esta etapa es importante obtener una participación activa del personal que se verá involucrado en la nueva política, y en general con aquellos que la usarán como guía para la toma de decisiones parcialmente independientes. Un jefe ejecutivo que quiera proporcionar una amplia participación en la dirección, debe de percatarse que su personal se entere con anticipación de la nueva política, antes de que sea tarde para su modificación.

***Emisión de una declaración escrita de la política.*** La difusión simultánea a todos los representantes de la dirección significa una comunicación efectiva y es un vínculo que relaciona y compromete al resto de la empresa en la decisión oficial de aceptar la política como parámetro de decisión.

***Interpretación y vivencia de las políticas.*** Las declaraciones, principios, y en general las políticas, ganan significado según se manifiesta en el desenvolvimiento del personal, esto puede ser precedido si en los procesos previos al establecimiento de la política se tuvo una participación activa del personal, evitando con esto que las modificaciones de actitud y decisión permanezcan en rangos aceptables para el personal, así generan como resultados políticas viables y flexibles.

***Evaluación interna.*** Los representantes de la dirección, así como empleados afectados por una nueva política, pueden desarrollar la experiencia necesaria para evaluar su propiedad y utilidad. Cualquier dificultad debe ser reportada junto con sugerencias constructivas. Después de ser aprobadas definitivamente, se procederá a emitirse con formalidad por la alta dirección en la política oficial de la empresa.

***Revisión y evaluación oficial de una política existente.*** Los miembros de un comité de políticas pueden conjuntar preguntas a efectuar al jefe ejecutivo en torno a recursos corrientes, requerimientos y situaciones, o bien, si una política debe modificarse, reconstruirse, o bien, sustituirse por un procedimiento.

**Revisión de una declaración de una política.** En la revisión de una política existente, el jefe ejecutivo puede demostrar su voluntad para seguir iniciativas situacionales y compartir puntos de vista de otros miembros de la organización. De esta manera se puede ceder la iniciativa a otros miembros de la organización.

#### **4. Importancia de las normas**

El establecer una norma en el desarrollo de sistemas, específicamente en su ciclo de vida, implica adecuar un conjunto de métodos, técnicas y procedimientos en general, así como proveer de una estandarización de documentos para una correcta comunicación de las áreas participantes en la elaboración de un sistema. Todo esto permitirá cumplir con los requisitos especificados, asegurando con ello la obtención de la calidad a través de la realización propia del proceso.

La concordancia de un conjunto de normas hacia un parámetro definido, que puede ser de carácter técnico, o bien de procedimiento, y que sean establecidas en un ámbito definido (local, regional, nacional o internacional), tienen el propósito de ser aceptadas y respetadas dentro de la organización. De esta manera es posible el establecer cooperación en el desarrollo de proyectos, sin tener que redefinir las técnicas o los procedimientos de los equipos de trabajo, evitando con ello la pérdida de tiempo. Como ejemplos podemos citar en cuando a las normas de carácter técnico, las que define el CCITT en aspectos de comunicaciones, o bien las de procedimientos que define la ISO.

En general los esfuerzos de depuración y corrección de errores, son necesarios debido a errores cometidos en las primeras fases del ciclo de vida del sistema, debido principalmente a que la terminología y documentación no se encuentran normalizadas. Es por ello que el establecer normas específicas ayudan a evitar la prolongación en los plazos de entrega de sistemas, además de auxiliar a conseguir un producto de mayor calidad y menor costo.

Bajo una normatividad establecida es posible tener una mejor comunicación técnica entre el personal de desarrollo, debido a que se agiliza la localización de errores en códigos realizados, así como en la aplicación de técnicas y metodologías en las etapas de análisis, diseño y desarrollo. Además el personal de desarrollo de sistemas, puede tener la capacidad de desarrollar otro proyecto con costos mínimos de adaptación y aprendizaje, ya que una normatividad establecida respalda a toda la organización.

#### **5. Organizaciones desarrolladoras de sistemas**

Cualquier organización que desee mejorar su proceso de desarrollo de sistemas, debe establecer tanto metas alcanzables como razonables; sin embargo, también requiere primordialmente identificar la diferencia de las actividades entre las organizaciones con proceso de elaboración de sistemas definido, con aquellas que no cuentan con ninguno. A continuación se mencionan las características de cada una.

##### **Organizaciones desarrolladoras sin ningún proceso definido**

- El proceso de elaboración de sistemas es generalmente improvisado por líderes y desarrolladores durante el transcurso del sistema.
- Aun cuando el proceso de desarrollo de software halla sido especificado, éste no es rigurosamente llevado a cabo.
- Se caracteriza por ser una organización reaccionaria.
- Los líderes de proyecto se enfocan a resolver crisis inmediatas.
- La calendarización y presupuestos para el desarrollo de sistemas son excedidos frecuentemente, debido a que éstos no están sustentados en estimados realistas, además de que continuamente se definen requerimientos y por ende se retoman las etapas de análisis y diseño.
- Frecuentemente se compromete la funcionalidad y calidad del sistema al tratar de cumplir con el calendario establecido.

- No cuentan con un objetivo base para juzgar la calidad de un sistema o para problemas que se presenten en el proceso. Por lo tanto, es difícil predecir la calidad del sistema.
- Las actividades para el aseguramiento de calidad tales como revisión y pruebas son a menudo restringidas o incluso llegan a eliminarse, sobre todo cuando se terminan los tiempos establecidos.

#### **Organizaciones desarrolladoras con un proceso definido**

- Se dispone de un proceso de administración de desarrollo y mantenimiento de software.
- Existe una comunicación efectiva entre el staff existente, los nuevos empleados y el cliente.
- Las actividades son realizadas de acuerdo al proceso planeado.
- El proceso es actualizado continuamente cuando es necesario, las mejoras son implantadas en base a pruebas piloto y aun análisis de costo/beneficio.
- Los roles y responsabilidades se encuentran claramente definidos en el proceso, para quienes desarrollan el proyecto (sistema).
- Los gerentes y líderes de proyecto se encargan de verificar la calidad del producto de software, así como de la satisfacción de los clientes.
- Calendarios y presupuestos están realizados en base a la experiencia histórica de proyectos efectuados y en tiempos realistas.
- El proceso es disciplinado y consistentemente realizado por todos los participantes, debido a que entienden el valor de efectuar el proceso correctamente.

#### **6. Beneficios de políticas, normas y procedimientos**

Aunque en general los directivos e ingenieros de software frecuentemente conocen los problemas que les aquejan, no terminan por ponerse de acuerdo en cuáles mejoras merecen mayor atención. Sin una estrategia organizada para el

mejoramiento, es difícil alcanzar un consenso entre los directivos y los profesionales de la ingeniería para identificar las actividades prioritarias a efectuar. Así es necesario constituir políticas, normas y procedimientos dentro de una empresa, con el objeto de establecer un proceso inicial para la elaboración de sistemas.

Las políticas, normas y procedimientos permiten a las organizaciones contar con una guía para controlar sus sistemas, en cuanto a su desarrollo y mantenimiento, así como permiten evolucionar la cultura de ingeniería de software y sirven de guía para buscar la excelencia en la elaboración de toda clase de sistemas. De hecho, la elaboración de sistemas se concibe como un conjunto de actividades, métodos, prácticas y transformaciones que emplea el personal de una institución o empresa para desarrollar y mantener la elaboración de sistemas. Ésto incluye desde la planeación del sistema por parte de la dirección a través de su dirección de informática, el diseño de documentos, creación del código, realización de casos prueba y el mantenimiento posterior del sistema.

Las políticas, normas y procedimientos también guían a las organizaciones de software con el apoyo de estrategias de mejoramiento de calidad de un sistema en dos rubros, el ciclo de vida del sistema y el aseguramiento de calidad. En lo que se refiere a ciclo de vida del sistema se consideran los puntos siguientes:

- Apoyo en propuestas y revisión de contratos, asegurando que todos los requerimientos de un sistema estén bien especificados y que la organización desarrolladora tenga la capacidad de cumplirlos.
- Especificación clara en el análisis de requerimientos del sistema a elaborar, asegurando principalmente que sean revisados y acordados con el cliente.
- Durante el análisis y diseño considerar a detalle todos los elementos e interfaces para la elaboración del sistema, verificando que satisfagan la especificación deseada.
- Realización e integración de pruebas al sistema, demostrando que el sistema integrado funciona correctamente y satisface las expectativas del cliente.

- **Instalación y liberación del sistema, poniéndolo a disponibilidad del cliente.**
- **Capacitación a usuarios en el uso del sistema liberado, de tal forma que puedan operarlo y beneficiarse de los resultados obtenidos.**
- **Mantenimiento del sistema, permitiendo que se asegure su operación para el cumplimiento de los nuevos requerimientos del cliente.**

En lo que respecta al aseguramiento de calidad se consideran los puntos siguientes:

- **Establecimiento de reglamentos generales dentro de la organización, que sirvan para ordenar en su mayoría, las actividades involucradas al elaborar sistemas.**
- **Organización y control de proyectos, de tal forma que exista un seguimiento al desarrollar sistemas.**
- **Recavación y sistema de control de registros, a través de la documentación de todas las actividades consideradas en el manual de procedimientos para la elaboración de sistemas.**
- **Seguridad, respaldos y almacenamiento de todas las facetas del sistema, asentándolas en las formas o formatos.**
- **Realización de acciones correctivas, de tal forma que aseguren que el sistema cumple con las especificaciones y es efectivo en el logro de resultados.**
- **Capacitación y entrenamiento del personal, con el fin de asegurar su competencia y motivación para la consolidación del sistema.**

El beneficio principal de una organización de software, al contar con un proceso para la elaboración de sistemas, es la institucionalización; ésto por medio de la instauración de políticas, normas y procedimientos en sus estructuras organizacionales. La institucionalización también proporciona el vínculo para la construcción de una infraestructura y del establecimiento de una cultura corporativa en la institución o empresa, que precisamente soporte los métodos, prácticas y procedimientos a efectuar con el fin de reforzar, mantener y mejorar el proceso de elaboración de sistemas. Finalmente, una vez establecido el desarrollo de sistemas en una institución o empresa, éste redundará principalmente en la elaboración de sistemas posteriores.

## **NORMA ISO 9000**

*Si logras mostrarle a una persona lo malo que ella está haciendo, procura hacer entonces lo bueno. La gente cree sólo lo que mira. Deja que vean tus obras buenas.*

*Henry D. Thoreau.*



## **NORMA ISO 9000**

### **1. Orígenes de la Norma ISO 9000**

En una organización un factor trascendental es la calidad que pudiesen tener sus productos o servicios. Examinando el esquema mismo de la empresa podemos encontrar funciones como tensiones internas, factores sociales o defectos intrínsecos en las actividades de la propia organización que perjudican notablemente para alcanzar la calidad en los productos o servicios que se desean.

En nuestros días en un mundo de libre comercio la calidad es un factor fundamental para poder mantener la competitividad de una empresa. La decisión de armonizar las normas de calidad y los lineamientos específicos de una organización es esencial, ya que de no llegar a un consenso la empresa corre el riesgo de quedar excluida de los mercados tanto nacionales como internacionales.

Durante varios años los proveedores se vieron obligados a cumplir con principios filosóficos; así como, con métodos y sistemas de garantía de calidad, en algunos casos, con enfoques diferentes para cada cliente. Sucedió, y aun ocurre, que era necesario tener manuales y procedimientos que cumplieran con tantos sistemas como clientes potenciales existieran.

La Norma ISO 9000 fue creada por la Organización Internacional para la Normalización mejor conocida como ISO, conformada por los cuerpos normalizadores de aproximadamente cien países; en conjunto forman una federación mundial con aportación de sus respectivas normas nacionales.

ISO es una organización no gubernamental oficialmente establecida el 23 de febrero 1947. La misión principal de ISO es promover el desarrollo de la normalización y de las actividades relacionadas en el mundo con el intercambio internacional de bienes y servicios, también pretende la cooperación en los campos de actividades intelectuales, científicas, tecnológicas y económicas.

Entre los antecedentes que se tienen con respecto a la normalización y sus orígenes se encuentran los siguientes hechos; cabe mencionar que destaca la industria militar aeroespacial en su uso.

- ◆ En el año 1926 se establece la Federación Internacional de la Asociación Nacional de Normalización, el énfasis que daba a sus normas se basaba principalmente en la ingeniería mecánica.
- ◆ Después de la Segunda Guerra Mundial delegados de 25 países decidieron crear una organización internacional que facilitara la coordinación y unificación internacional de normas industriales, es como nace ISO.
- ◆ La primera norma ISO fue publicada en 1951 la cual estuvo titulada "Norma de referencia de temperatura para medidas industriales".
- ◆ El primer documento que tuvo mayor significancia fue la norma MIL-Q-9858, editada por el departamento de defensa de los E.U.A en 1959, donde define un programa de administración de calidad.
- ◆ En 1967 la Comisión de Energía Atómica (AEC) de E.U. editó el documento: "General Design Criteria for Nuclear Plants", el cual daba los criterios para desarrollar e implantar un programa de aseguramiento de calidad.

- ◆ En 1969 la Comisión de Energía Atómica (AEC) publicó el apéndice B del 10 CFR 50 "CFR-Code of Federal regulation quality assurance criterial for Nuclear power plants" éste documento dió a conocer al mundo los requisitos básicos de aseguramiento de calidad.
- ◆ En 1970 se publicó la norma ANSI N-45.2 "Quality Assurance Program Requirements for Nuclear Power Plants", ésta establece los requisitos del programa de aseguramiento de calidad de una manera más comprensible.
- ◆ En 1971 en la sección III del código ASME se incluyó en su sección NCA el artículo 4000 "Quality Assurance", que indica los requisitos de aseguramiento de calidad aplicables a fabricantes y montadores, cuyas actividades se relacionan con la sección III de este código.
- ◆ En 1975 el organismo internacional de energía atómica desarrolló sus guías de aseguramiento de calidad. El código N° 50-c.QA "Aseguramiento de calidad para la seguridad en las centrales código de prácticas". En este se establecen los 13 criterios de aseguramiento de calidad.
- ◆ En 1977 cierto número de países de la Comunidad Europea habían hecho sus normas nacionales para operar sistemas de control de calidad en la industria manufacturera y de servicios. En 1978 se publicó la primitiva versión de las norma serie ISO 9000 y rápidamente fueron adoptadas por diferentes países con sus denominaciones normativas propias.
- ◆ En 1979 la Organización Internacional de Normalización (ISO) con sede en Ginebra, Suiza, por iniciativa de varios países formó un comité técnico (TCQ176)

con el objeto de desarrollar una norma única para la operación y administración del aseguramiento de la calidad. El objetivo a cumplir que fuera la base mínima para garantizar la calidad de una empresa frente a cualquier cliente en cualquiera de los países miembros, mediante la certificación y registro único de un sistema de calidad por un cuerpo certificador independiente, debidamente acreditado, de esta manera se pretendía eliminar la multiplicidad de auditorías de evaluación.

- ◆ En 1987 la Organización Internacional de Normalización (ISO) crea la norma para un sistema de calidad mejor conocida como ISO 9000, basada principalmente en la norma británica BS-5750. En ese mismo año la Comunidad Europea (CE) notificó a los Comités Europeos de Normas (CEN) y Europeo de Normas Electrónicas (CENELEC) que adoptaran las normas internacionales de la ISO 9000 como las normas europeas y las designaran con el nombre de EN 29000.
- ◆ En 1991 se publica la primera y única versión que muestra la guía para la aplicación de ISO 9001 en el desarrollo de software. Es en ese mismo año cuando México adopta ciertas partes de la ISO 9000 para desarrollar sus normas nacionales conocidas como NMX-CC.

## **2. Objetivos de la Norma ISO**

La ISO busca desarrollar un estándar industrial que facilite el comercio internacional. La ISO 9000 no es un estándar para un producto, por el contrario es un estándar aplicable a un sistema de calidad. Esto involucra al proceso en que se crea un producto o un servicio. De hecho se puede aplicar a cualquier proceso en cualquier parte del mundo. El objetivo principal de ISO 9000 que se maneja en el mundo de los negocios es: "Conocer las expectativas y requerimientos del cliente". De hecho, cuando el comité técnico encargado de la normalización de aseguramiento y administración de calidad inició sus trabajos, se enfrentó con el reto de alcanzar un consenso entre las diferentes filosofías y conceptos de calidad existentes. Dentro de los objetivos esenciales se encontraron:

1. Desarrollar un código mínimo de prácticas de administración de calidad, aplicable a cualquier tipo de: empresa, aseguramiento y administración de calidad.
2. Estimar lo que una empresa está obligada a realizar como mínimo para poder responder a los requerimientos de un mercado competitivo.
3. La norma es la base para poder establecer acuerdos sobre las responsabilidades de proveedores y compradores respecto a la calidad de los bienes o servicios intercambiados.

ISO 9000 reconoce principalmente tres fases dentro del negocio que afectan la calidad del mismo, de hecho es un ciclo que nunca termina: la planeación, el control y la documentación. Entre los puntos a considerar en cada fase se encuentran:

- ◆ Aquellas actividades que afectan la calidad deben ser *planeadas*, esto con el fin de llevar a cabo esas metas; la autoridad y responsabilidad son definidas y entendidas.
- ◆ Aquellas actividades que afectan la calidad deben ser *controladas*, esto con el fin de conocer esos requerimientos específicos, anticiparse a los problemas y tomar acciones correctivas para que sean planeadas y llevadas a cabo.
- ◆ Aquellas actividades que afectan la calidad deben ser *documentadas*, esto con el fin de entender los objetivos de calidad y los métodos, la interacción igualitaria dentro de la organización, la retroalimentación para el ciclo de planeación y la evidencia objetiva del sistema de calidad para aquellas partes que la requieran tal como los clientes.

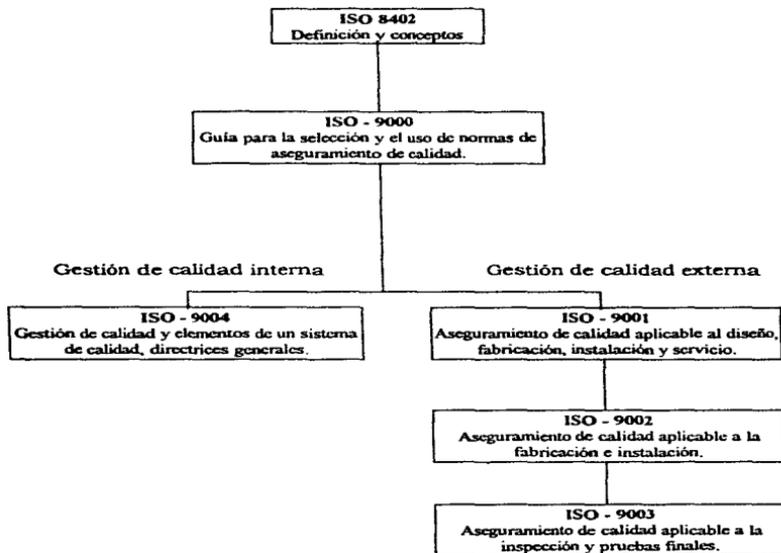
Todas las organizaciones buscan elaborar un producto o servicio para satisfacer una necesidad o un requerimiento de un cliente. Estos requerimientos se incorporan a las especificaciones particulares del sistema para diseñar o elaborar el producto o

**servicio.** En términos generales una organización debe buscar el alcanzar continuamente los siguientes objetivos referentes a la calidad en la elaboración de sus productos:

1. La organización debe lograr y mantener la calidad del producto o servicio elaborado para alcanzar continuamente los requerimientos establecidos por el comprador.
2. La organización debe ofrecer confianza a su propia administración de que la calidad pretendida se está logrando y manteniendo.
3. La organización debe ofrecer confianza al comprador de que la calidad pretendida se está o será alcanzada en el producto entregado.

### 3. Estructura original de la Serie ISO 9000

En forma general la Norma ISO 9000 consta de un vocabulario y cinco normas tal como se muestra en el siguiente diagrama:



La Organización Internacional de Normalización (ISO) define dos conceptos básicos en su serie ISO 9000, tales conceptos definen la aplicación propia de la norma según el contexto que se desee aplicar. Los conceptos se describen como:

- **Casos no contractuales**

En los casos no contractuales se proporciona los lineamientos para implantar los elementos necesarios de un sistema de calidad con el fin de asegurar a la misma empresa de que todos los factores técnicos, administrativos y humanos que influyen en la calidad de sus productos/servicios estén bajo control. Para estos casos se utiliza la norma ISO 9004.

- **Casos contractuales**

En los casos contractuales se permite al proveedor y al comprador de los productos/servicios seleccionar el modelo del sistema de calidad que más se ajuste a las condiciones contractuales de compra y venta. Las normas contractuales son la base para que el comprador o cliente pueda evaluar el sistema de calidad del proveedor, es decir, asegurar el nivel del producto o servicio. Para los casos contractuales se utilizan la ISO 9001, 9002 y 9003.

En la siguiente tabla se muestra una lista de las normas vigentes de aseguramiento de calidad emitidas por ISO.

<b>Norma ISO</b>	<b>Título</b>
ISO 8402:1994	Administración y aseguramiento de la calidad. Vocabulario
ISO 9000-1:1994	Estándares y aseguramiento de calidad - Parte 1: Guías para selección y uso
ISO/DIS 9000-2	Estándares de administración y aseguramiento de calidad - Parte 2: Guías de aplicación de ISO 9001, ISO 9002 e ISO 9003
ISO 9000-3:1991	Estándares de administración y aseguramiento de calidad - Parte 3: Guías para la aplicación de ISO 9001 al desarrollo y mantenimiento de software.
ISO 9000-4: 1993	Estándares de administración y aseguramiento de calidad Parte 4: Guía para programas de confiabilidad ( <i>Dependability</i> )
ISO 9001:1994	Modelo para aseguramiento de calidad en el diseño, desarrollo, producción, instalación y servicio.
ISO 9002:1994	Modelo para aseguramiento de calidad para producción, instalación y servicio.

Norma ISO	Título
ISO 9003:1994	Modelo para aseguramiento de calidad para inspección final y pruebas.
ISO 9004-1:1994	Elementos de administración y sistemas de calidad - Parte 1 Guías
ISO 9004-2: 1994	Elementos de administración y sistemas de calidad - Parte 2: Guías para servicio.
ISO 9004-3: 1993	Elementos de administración y sistemas de calidad - Parte 3 Guías para materiales procesados
ISO 9004-4: 1993	Elementos de administración y sistemas de calidad - Guías para mejora de la calidad
ISO 10005: 1995	Administración de calidad - Guías para planes de calidad
ISO 10007: 1995	Administración de calidad - Guías para administración de la configuración.
ISO 10011-1: 1990	Guías para auditar sistemas de calidad - Parte 1: Auditorías
ISO 10011-2: 1991	Guías para auditar sistemas de calidad - Parte 2: Criterios de calificación para auditores de sistemas de calidad.
ISO 10011-3: 1991	Guías para auditar sistemas de calidad - Parte 3: Administración de programas de auditoría.

#### 4. Panorama y aplicación de las Normas

##### **ISO 8402 Vocabulario**

Es la norma internacional que define los términos usados en toda la serie ISO, con el fin de que exista mutua comprensión en las comunicaciones internacionales. Los primeros términos que define son: calidad, políticas, aseguramiento, sistema, grado o clase, administración, etc..

##### **ISO 9000 Administración de calidad y normas de aseguramiento de calidad, guía y procedimientos para su selección y uso.**

Aplicación: Todas las industrias

Esta norma ayuda a preparar nuestros sistemas gerenciales internos de calidad y a seleccionar el modelo específico con base a las normas 9001, 9002 y 9003. También permite conocer la filosofía de las normas de sistemas de calidad, sus características, modelos y en qué condiciones utilizarías, los requisitos de documentación y auditoría.

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

**Aplicación:** Firmas de ingeniería y construcción, Fabricantes que diseñan desarrollan, producen e instalan productos y servicios.

Esta norma permite asegurarla a los clientes, que la calidad de los productos de una compañía se conforman con requerimientos especificados durante todas las etapas, incluyendo desde el diseño, desarrollo, instalación y servicio. Se aplica particularmente cuando hay un contrato que requiere un diseño específico y cuando los requerimientos del producto son establecidos en términos de su comportamiento.

A continuación se mencionan los 20 puntos que considera ISO 9001 para el aseguramiento de calidad.

1. Responsabilidades gerenciales.
2. Principios del sistema de calidad.
3. Revisión de contrato.
4. Control de diseño.
5. Control de documentos y datos.
6. Adquisiciones.
7. Control de productos suministrados por el cliente.
8. Identificación y seguimiento del producto.
9. Control de proceso.
10. Inspección y prueba.
11. Control de equipos de inspección, medición y prueba.
12. Estado de inspección y prueba.
13. Control de producto no conforme.
14. Acciones correctivas.
15. Manejo, almacenamiento, empaque, conservación y entrega.
16. Control de registros de calidad.
17. Auditorías internas de calidad.
18. Capacitación.
19. Servicios.
20. Técnicas estadísticas.

**ISO 9002 Modelo para el aseguramiento de la calidad en producción, instalación y servicio.**

Aplicación: Compañías en el proceso químico, no involucradas en el diseño de productos y servicio posventa.

Esta es la norma más común para fabricantes y se aplica cuando ya existe un diseño o especificaciones establecidas, las cuales constituyen los requerimientos especificados del producto. También supone que el sistema de calidad establecido demuestra que el proveedor puede continuar fabricando el producto de acuerdo con lo estipulado. Es menos rígida que la 9001.

**ISO 9003 Modelo para el aseguramiento de la calidad en inspección y pruebas finales.**

Aplicación: Talleres pequeños, divisiones dentro de una compañía, distribuciones de equipo de prueba que inspeccionan productos.

Esta norma es útil para demostrar la capacidad en actividades de inspección y pruebas, donde el comportamiento del producto o servicio tiene que cumplir con ciertos requerimientos.

**ISO 9004 Administración de la calidad y elementos del sistema de calidad.**

Aplicación: Todas las industrias.

Esta norma es la declaración más exhaustiva de lo que la norma constituye. Se podría decir que un sistema de calidad básico conformado según ISO 9004, podría ajustarse para que cubriera la 9001, 9002 y 9003. Esta norma muestra énfasis sobre los clientes, el establecimiento de responsabilidades y la importancia de evaluar riesgos y beneficios.

## **5. Principales funciones a cumplir de ISO**

La Organización Internacional de Normalización (ISO) se encuentra integrada por representantes de los países miembros, éstos conforman el cuerpo de miembros de ISO. Los integrantes de ISO tienen cuatro puntos principales que cumplir, los cuales son:

- ◆ Informar en sus respectivos países sobre los intereses potenciales que surjan de las iniciativas y oportunidades de la normalización internacional.
- ◆ Participación concertada en los acuerdos de normatividad de los intereses del país que estén representando durante las negociaciones internacionales.
- ◆ Asegurar la participación de un secretariado en los comités técnicos y subcomités donde su respectivo país tenga un particular interés.
- ◆ El apoyo financiero del país que representan para las operaciones centrales de ISO, a través del pago de los derechos de los miembros.

Es importante recalcar que un miembro correspondiente es usualmente una organización dentro de un país, la cual no desempeña formalmente la actividad de desarrollo de normas, es decir, no toman parte activa en el trabajo técnico. Estas organizaciones sólo se limitan a mantenerse informadas sobre los trabajos realizados y que son de interés para ellas.

## **6. Principios de desarrollo de la Norma ISO**

La Organización Internacional de Normalización (ISO) realiza la elaboración de sus normas y estándares a través de 2700 comités técnicos, de subcomités y de grupos de trabajo. Dichos comités están constituidos por representantes calificados de la industria, investigación, institutos, autoridades gubernamentales, consumidores y

organizaciones internacionales, que en conjunto y con patrones iguales solucionan problemas globales de estandarización y normalización.

Una de las responsabilidades importantes para la administración de un comité de estandarización es la aceptación de normas de los cuerpos nacionales, como por ejemplo AFNOR, ANSI, CSBTS, JALEO, SIS, etc. que de hecho constituyen el cuerpo de miembros de ISO.

Al desarrollar cualquier norma internacional de ISO, ésta debe cumplir con los siguientes principios, que se consideran como fundamentales.

**1. Consenso**

Considerar durante el desarrollo de sus estándares todos los intereses de: fabricantes, vendedores, clientes, grupos de consumidores, laboratorios de prueba, gobiernos, profesionales de ingeniería y centros de investigación.

**2. Industria mundial**

Busca satisfacer completamente, tanto a las industrias como a los clientes alrededor del mundo, basándose en proponer soluciones globales.

**3. Voluntariedad**

La estandarización internacional de los mercados mundiales se basa sobre compromisos voluntarios de los intereses del propio mercado.

Cuando se desarrolla un estándar para una industria, el proceso que se sigue consiste en que dicha industria comunique sus necesidades a su cuerpo de normalización nacional. Posteriormente se propone como un nuevo trabajo a ISO, si el trabajo es formalmente aprobado se pasa a la primera fase, la cual consiste en la definición de la estructura técnica del estándar. Durante esta fase participan grupos de trabajo y expertos técnicos del país interesado en la norma.

Una vez cubiertos los aspectos técnicos del estándar se pasa a la segunda fase, en la cual se llevan negociaciones entre los diversos países para detallar las especificaciones del estándar, esta fase suele conocerse como la fase de consenso.

La fase final es la autorización formal del estándar, para dicha aprobación se estipula el voto a favor de las dos terceras partes de los miembros de ISO que participaron en la elaboración del estándar y el 75% de los demás miembros. Una vez cubiertos estos puntos la norma es publicada como un estándar internacional de ISO.

## **7. Beneficios de ISO 9000**

La estandarización de los sistemas de calidad bajo ISO 9000 tiene diferentes implicaciones según el contexto de que se trate, muchas de ellas pueden ser estratégicas a corto o largo plazo. Cabe aclarar que la intensidad de los beneficios que se puedan obtener están en función de cómo las normas son aplicadas. De hecho la estructura de ISO 9000, por si misma, proporciona dos aplicaciones principales a obtener:

- ◆ **Propósitos de administración de calidad.** Al adoptar las normas se obtiene con facilidad una descripción detallada del plan a seguir de un sistema de calidad interno.
- ◆ **Propósitos contractuales.** Se demuestra que un sistema de calidad es una condición de contrato con un cliente.

Lo principal en un sistema de calidad es conocer los requerimientos de calidad de los clientes, con ello existe indirectamente un buen tramo averiguado para la aplicación de las normas. En términos generales se puede decir que los sistemas de calidad exhiben atributos en común aplicables a cualquiera de los cuatro grupos genéricos. La filosofía que se adquiere al cumplir cabalmente con una norma ISO se puede resumir en los puntos siguientes:

- ◆ Una filosofía de prevención antes que el descubrimiento.
- ◆ Revisión continua de los puntos o procesos críticos, acciones correctivas y los efectos o resultados.
- ◆ Comunicación consistente dentro de los procesos, incluyendo la facilidad de comunicación entre proveedores y clientes.
- ◆ Control eficiente de documentos críticos
- ◆ Conciencia de calidad entre todos los empleados.

Entre los beneficios de registrarse bajo ISO 9000 se encuentran los siguientes puntos.

◆ **Acceso a mercados**

El acceso a mercados es el beneficio más crítico de la certificación. Esto facilita en mantener o crear relaciones con clientes. De acuerdo con John Symonds la cantidad de empresas en el mundo que se están certificando en Europa y cada vez más en otros partes del mundo va en aumento. En octubre de 1993 había 45,000 empresas certificadas con un incremento del 70% en 9 meses. Se espera que en el 2000 existan de uno a dos millones de empresas certificadas.

◆ **Un proveedor de competencia**

Debido a la globalización que se esta teniendo en los últimos años muchas empresas toman como arma la certificación esto con el propósito de obtener la confianza de sus clientes en todos los sentidos. Otro punto a considerar en la certificación es la confianza de clientes, ya que en vanas industrias surge como un medio de supervivencia

- ◆ **Reducción potencial de auditorías**  
En muchas industrias se espera que a través de la certificación se puedan evitar docenas de auditorías de calidad que se les practican.
- ◆ **La adquisición de una política corporativa de la empresa.**
- ◆ **Ventaja al participar en concursos por algún contrato.**
- ◆ **Facilita el intercambio comercial y la transferencia de tecnología.**
- ◆ **La confianza de adquirir un producto con calidad y a precio razonable.**
- ◆ **Incrementa la distribución eficiente del trabajo y el fácil mantenimiento.**
- ◆ **Gran compatibilidad e interoperabilidad de bienes y servicios.**

### **8. Conceptos Elementales de calidad**

La necesidad de conocer el lenguaje y terminología que utiliza la norma ISO 9000 es fundamental para cualquier organización que desee incorporar la norma a su institución o empresa, ésta permitirá comprender y entender de manera más amplia los estándares a seguir para su correcta aplicación durante las etapas de desarrollo de cualquier producto. Por otro lado dichos términos se utilizan en un sentido más específico en comparación de los conceptos del diccionario común.

En la norma ISO 9000 cada término es cuidadosamente definido. Cada sección de la norma ofrece la definición de los términos utilizados, y dichas definiciones fueron reunidas y presentadas en su norma ISO 8402, "International Standard Quality Vocabulary". A continuación mencionaremos los conceptos fundamentales al hablar de calidad al diseñar cualquier producto o servicio.

### **Elemento**

Cualquier ente que puede ser descrito y considerado individualmente. Como ejemplo: un proceso, producto, organización, sistema, persona, etc.

### **Producto**

Es el resultado de actividades o de procesos, inclusive la combinación de los mismos. Estos pueden clasificarse en dos tipos generales que son:

- **Productos tangibles:** Son aquellos que son palpables, no abstractos. Como por ejemplo, los ensambles, materiales procesados, etc.
- **Productos intangibles:** Son aquellos que son términos abstractos en su concepción. Como por ejemplo: conocimientos, conceptos o información.

### **Calidad**

Según Perry L. Johnson la define: La ISO 9000 la interpreta como una integración de las características y rasgos distintivos que determinan la satisfacción de las necesidades y expectativas de los clientes.

### **Calidad**

Según la Norma ISO 9000: Conjunto de características de un elemento que le confieren la aptitud para satisfacer necesidades explícitas e implícitas.

La satisfacción de esas necesidades deben identificarse como criterios a cumplir para satisfacer las expectativas del cliente. Dichos criterios pueden incluir aspectos de desempeño, facilidad de uso, facilidad de mantenimiento, etc.. Cabe aclarar que según el contexto que se aplique, las necesidades deben ser interpretadas, así en un ambiente contractual las necesidades son especificadas y en otros casos son identificadas y definidas. Un aspecto en recordar es que las necesidades son cambiantes con el tiempo, por lo que se recomienda una revisión periódica de estas.

Toda empresa o institución al ofrecer servicios de calidad es conveniente que considere la norma ISO 9004 con el fin de asegurar y fortalecer su sistema de calidad

interno, de hecho la aplicación es voluntaria. La ISO 9004 define la implantación de un *sistema de calidad* interno, precisamente basado en la filosofía de aseguramiento de calidad y documentado en su manual de calidad. En este sistema se deben incluir todas las políticas, procesos y procedimientos necesarios para asegurar la calidad. También menciona que para garantizar la calidad de los productos y servicios se debe estructurar un *plan de calidad*, el cual deba incluir todas las inspecciones, pruebas y verificaciones necesarias a lo largo de su ciclo de vida, es decir, desde su concepción y diseño hasta su instalación y servicio.

El principio fundamental de un sistema de calidad supone que si se tienen planeadas y documentadas todas las acciones necesarias para que un producto o servicio se haga con calidad, y estas acciones efectivamente se hacen tal como están documentadas, existe la confianza de que el producto o servicio será de la calidad planeada.

En la siguiente tabla se describen las definiciones fundamentales sobre un sistema de calidad bajo el contexto de la norma ISO 9000.

Sistema de calidad	Es la estructura organizacional, los procedimientos, los procesos y los recursos necesarios para implantar la administración de calidad. De hecho está diseñado para la administración interna de la empresa o institución.
Administración de la calidad	Conjunto de actividades de la función general de administración que determina: la política de calidad, objetivos, responsabilidades e implantación de estos, por medios tales como: planeación de la calidad, control de calidad, aseguramiento de la calidad y mejoramiento de la calidad dentro del marco del sistema de calidad.
Planeación de la calidad	Actividades que determinan los objetivos y requisitos para la calidad. Esos requisitos incluye los elementos para la implantación del sistema de calidad.
Control de calidad	Técnicas y actividades de carácter operacional, utilizadas para cumplir los requisitos para la calidad

<b>Aseguramiento de calidad:</b>	Conjunto de actividades planeadas y sistemáticas implantadas dentro del sistema de calidad, demostradas según se requiera para proporcionar confianza adecuada de que un elemento cumplirá los requisitos para la calidad.
<b>Mejoramiento de la calidad</b>	Aquellas acciones tomadas en la organización para incrementar la efectividad y la eficiencia de las actividades o procesos. El fin es proveer de beneficios adicionales tanto para la organización como para sus clientes.
<b>Manual de calidad</b>	Documento donde se formaliza y describe las políticas de calidad y sistema de calidad de la empresa.
<b>Plan de calidad</b>	Un documento especificando las prácticas, recursos y secuencia de actividades relativas a la calidad de un producto, proyecto o contrato.

## 9. El Software en la Norma ISO 9000

Con el progreso constante de la tecnología de información, la cantidad de productos de software se ha incrementado, y su administración durante la elaboración de sistemas se ha vuelto un proceso esencial. De ahí la importancia de ISO en establecer un sistema de administración de calidad que permita proveer una guía para el aseguramiento de calidad en el software.

En primera instancia al hablar de software, debemos conocer e identificar cómo la norma ISO 9000 clasifica a los distintos productos. La norma ISO 9000 para su mejor comprensión de los mismos los refiere a cuatro grupos genéricos y son los siguientes:

- ◆ **Hardware** - Producto tangible con características distintivas. Compuesta de productos consistentes en piezas manufacturadas o partes, ensamblaje de componentes.
- ◆ **Software** - Creación intelectual constituida por información escrita o registrada en diferentes medios de soporte. Esta puede encontrarse en forma de: conceptos,

transacciones, procedimientos o programas de cómputo. Como ejemplo esta el software de computación

- ◆ **Materiales procesados (finales o intermediarios) - Producto tangible generado por la transformación de materias primas en un estado requerido. El estado del producto puede presentarse como sólido, líquido, gas o combinaciones inclusive.**
- ◆ **Servicios - Productos intangibles relativos a actividades tales como: planear, evaluar, entrenar, operar o mantener un producto tangible.**

Se observa claramente que la Norma ISO 9000 articula al *desarrollo de software* dentro de dos rubros descritos anteriormente, por una parte para la elaboración de software se necesita establecer un plan a seguir y sus actividades inmersas en las distintas fases de desarrollo del software (sistema informático), esto permitirá sentar las bases para el aseguramiento del software. Dicho software surge de la necesidad de resolver un problema específico en una área de una institución o empresa. Por otro lado el desarrollador de software cumple con la función de ofrecer un servicio debido a que para realizar el sistema informático debe contemplar el establecimiento de un contrato, la factibilidad del sistema, la planeación de actividades, las pruebas necesarias y la capacitación de personal técnico-operativo.

Se puede apreciar explícitamente que ISO al identificar los requerimientos para un sistema de calidad genérico en situaciones contractuales establece el uso de la Norma ISO 9001, que abarca el ciclo completo de la elaboración de un producto desde su diseño, desarrollo, instalación y servicio. En cuanto a los servicios ISO manifiesta que se considere además a la Norma ISO 9004. Sin embargo debido al proceso de desarrollo y mantenimiento de software que es relativamente distinto de otros tipos de procesos industriales, la Organización de Estándares Internacionales decide agregar como apoyo la ISO 9000-3.

La norma ISO 9000-3 es una guía para facilitar la aplicación de ISO 9001, en aquellas organizaciones que desarrollan, proveen y suministran mantenimiento de software. En dicha norma se establecen las posibles diferencias comparándola con la ISO 9001, de hecho se cuenta con un par de referencias cruzadas entre ISO 9001 e ISO 9000-3, complementándose y proviéndose de asistencia mutua.

La ISO 9000-3 es aplicable en situaciones contractuales de productos de software cuando se presenta:

- Que el cumplimiento del contrato específicamente se requiere un esfuerzo de diseño, y los requerimientos del producto son expresados principalmente para cumplimiento de plazos predeterminados o en su defecto éstos serán establecidos.
- La confianza del producto está depositada en función de la demostración adecuada de ciertas capacidades del consultor en desarrollar, proveer y suministrar mantenimiento en el desarrollo de sistemas.

La Norma ISO 9000-3 se destaca principalmente por la definición concisa de los términos utilizados al desarrollar software. A continuación se definen dichos términos tal como los menciona ISO.

### **Software**

Creación intelectual incluyendo programas, procedimientos, reglas y cualquier documentación asociada perteneciente a la operación del sistema de procesamiento de datos.

[ISO 2382-1:1984]

### **Producto de software**

Conjunto completo de programas de computadora, procedimientos, documentación asociada e información diseñada para su liberación a un usuario.

### **Partida de software**

Cualquier parte identificable de un producto de software en el paso intermedio o en el paso final del desarrollo.

### **Desarrollo**

Todas las actividades a realizar para crear un producto de software.

### **Fase**

Segmento definido de trabajo.

### **Verificación de software**

El proceso de la evaluación de productos de una fase dada para asegurar su correcta aplicación y consistencia con respecto a los productos y estándares definidos como entradas a esa fase.

### **Validación de software**

El proceso de evaluación de software que asegura el cumplimiento con requerimientos específicos.

Las principales diferencias entre la ISO 9001 y la ISO 9000-3 se establecen en los puntos siguientes:

La ISO 9000-3 en la sección cinco punto uno hace referencia a las actividades del ciclo de vida. Se establece durante el desarrollo de un proyecto de software, éste debe ser organizado de acuerdo a un modelo de un ciclo de vida. Las actividades relacionadas con la calidad deben ser planeadas e implantadas con respecto a la naturaleza del modelo del ciclo de vida utilizado. Además vale la pena mencionar que en esta parte ISO 9000 es flexible ya que menciona que la descripción, guías, requerimientos o estructura se encuentran en función del modelo del ciclo de vida utilizado, es decir, no es restrictivo a un sólo modelo.

La ISO 9000-3 en su sección cinco punto diez hace referencia explícita sobre el mantenimiento de software, el cual si es inmediatamente posterior a la liberación del sistema; debe estar previamente acordado en el contrato que sostuvieron el comprador y el desarrollador. Las actividades que surjan en el mantenimiento las clasifica en tres puntos principales que son: resolución del problema, modificación de interfaces y la ejecución de mejoras. Los puntos a considerar en el mantenimiento del sistema deben ser específicamente acordados en el contrato y se mencionan como ejemplos a considerar los siguientes puntos:

- Programas de computadoras.
- Estructuras e información de sistemas.
- Especificaciones.
- Generación de documentos para el comprador o usuario.
- Generación de documentos de mantenimiento para uso del futuro proveedor de mantenimiento.

La ISO 9000-3 en su sección seis punto cuatro menciona que a través del uso de la tecnología de cómputo es posible la evaluación de las métricas del sistema. Esto con objeto de conocer qué tan efectivo se llevó a cabo el proceso de desarrollo en la reducción de la probabilidad de introducir fallas. Además de evaluar qué tan bien se llevó a cabo el proceso de desarrollo en cuanto a los objetivos de calidad basados en la calendarización del sistema. Vale la pena mencionar que ISO en dicho apartado menciona que no existen métricas de calidad de software a nivel universal y que queda a disposición del desarrollador utilizar la que él considere necesarias.



## **SISTEMAS DE INFORMACIÓN**

*Tanto que hay por hacer; y tan poco  
hecho.*

**Cecil Rhodes.**



## **SISTEMAS DE INFORMACIÓN**

### **Introducción**

En los últimos años se ha incrementado notablemente el empleo de la tecnología en las empresas u organizaciones. La tecnología facilita el desempeño de diferentes procesos dentro de una organización y además contribuye a mantener una estrategia de competitividad. Esto se debe principalmente a la rapidez de los cambios con los que ahora las empresas deben responder sobre los entornos en los que operan; así las organizaciones buscan alternativas de solución a sus problemas y con ello va en aumento la creación de sistemas de información que ayuden a resolver éstos mediante la toma de decisiones.

Primeramente es conveniente diferenciar dos conceptos que frecuentemente causan confusión entre las personas, el de tecnología de información y el de sistemas de información. El referimos a uno u otro tiene su diferencia y confundirlos puede resultar riesgoso. Por una parte la tecnología de información se concibe como la conjunción de las computadoras, software, telecomunicaciones y microprocesadores en sus distintos dispositivos o aparatos de transmisión y comunicación. La importancia radica en la conectividad que mantiene entre individuos y sus respectivas organizaciones, ya que permiten el intercambio de información que a la postre propina la toma de decisiones de una empresa. En tanto que los sistemas de información son el concepto usado para denotar el estudio del uso de la tecnología en contextos sociales. Por ejemplo, si se estudia el flujo de información dentro de una organización, se considera a la tecnología de información como el medio y la información involucrada como el fin.

Los avances continuos en la tecnología de computadoras y comunicaciones tienen un efecto profundo sobre la forma en que las personas realizan su trabajo. El uso cada vez más extenso de los sistemas de información está cambiando la naturaleza propia de la sociedad que hace uso de ellos. El desarrollo de sistemas de información ha venido jugando un papel importante en la evolución de la economía, a su vez los creadores de esos sistemas han influido y continuarán haciéndolo, en muchos de los aspectos fundamentales de cambio.

Los sistemas de información y el constante incremento del uso de las computadoras ocupan un sitio especial en las empresas donde facilitan la operación de sus actividades, tales como la reservación de boletos en aerolíneas, departamentos de archivos clínicos en hospitales, funciones de contabilidad y nómina, banca electrónica, sistemas de conmutación electrónica y un sin fin de aplicaciones grandes y pequeñas en diversas industrias.

Antiguamente en la era industrial lo más importante era el uso del capital, dinero y recursos tangibles, para generar nuevos productos. En el presente los recursos básicos son las ideas y el uso correcto, eficiente y eficaz de la información; así el empleo estratégico de la información, continuará creando virtualmente, nuevas oportunidades en todas las industrias. La habilidad para manejo de la información, junto con la innovación de nuevos productos, modernización de servicios y con un eficaz y mejor trato hacia los clientes y proveedores, serán uno de tantos factores que decidan cuáles empresas sobrevivirán para el siguiente siglo.



Figura 1. Elementos de un sistema de información

## 1. Perspectiva de los sistemas de información

En nuestros días los sistemas de información, a través de su papel central en la toma de decisiones de corporaciones, están llevando a cabo cambios fundamentales en cuatro aspectos:

- ◆ Las personas trabajan de manera más inteligente.
- ◆ Un cambio global en el concepto de industria.
- ◆ Las ideas como la información están tomando mayor importancia que el dinero.
- ◆ Las personas que trabajan con la información dominan la fuerza de trabajo.

El continuo incremento en la confiabilidad de los sistemas de información significará para aquellos que lo diseñan una responsabilidad cada vez mayor. Los sistemas de información deben ser capaces, utilizables, confiables y por encima de todo, servir como medios para alcanzar fines sin convertirse en un fin por sí mismos. Entre las metas principales a cumplir al diseñar un sistema de información basado en computadora debe *funcionar en forma apropiada, ser fácil de utilizar y adecuarse a la organización* para la que fue diseñado.

Los sistemas de información que benefician mayormente a las empresas son aquellos que se originan en los usuarios. De hecho se puede decir que se originan de una necesidad de la organización misma, comúnmente se utilizan para manejar funciones rutinarias o de monitoreo de la información para evitar problemas. El desarrollo de los sistemas es un esfuerzo conjunto, por una parte las contribuciones de los usuarios y por otro el papel de los analistas de software que entre sus actividades esenciales está el extraer las ideas necesarias para efectuar el análisis y diseño del sistema de información.

## **2. Conceptos de sistemas de información**

Para comprender ampliamente la importancia de los sistemas de información en una organización empezaremos con citar algunas definiciones de sistemas de información:

- ◆ Conjunto de elementos interrelacionados que utilizan datos y los procesa a fin de brindar información para la toma de decisiones.
- ◆ Aquellos elementos que proporcionan informes periódicos para la planeación, control y toma de decisiones.
- ◆ Complejo de elementos que interactúan entre sí para manipular, crear y consultar información proveniente de una base de datos.
- ◆ Conjunto de procedimientos organizados que a través de su desempeño proveen de suficiente información para la toma de decisiones, comunicaciones y el control propio de la organización.
- ◆ Conjunto de personas, procedimientos y recursos como hardware y software, que transforman datos en productos de información para usuarios finales. Es decir, transforman y diseminan la información dentro de la organización.

Los sistemas de información están compuestos de subsistemas que en conceptos de cómputo están relacionados con hardware, software, medios de almacenamiento de datos y bases de datos. A ese conjunto particular de subsistemas utilizados tal como equipo específico, programas, archivos y procedimientos se le denomina una aplicación de sistemas de información.

En la elaboración de los sistemas de información es posible identificar las siguientes características.

- ◆ Son desarrollados por especialistas de cómputo que laboran en empresas consultoras de sistemas.
- ◆ Involucran múltiples usuarios dentro de la organización.

- ◆ **Proveen de flexibilidad computacional, en otras palabras son creados según las necesidades particulares de cada organización.**
- ◆ **Son creados para solucionar problemas de un área en particular**

Debido a que los sistemas de información sirven de apoyo a los sistemas que conforman a una organización, lo primero a realizar por parte de los desarrolladores es el estudio del sistema organizacional, éste visto como un todo de tal forma que se pueda detallar precisamente el sistema de información. De lo anterior viene implícitamente la ejecución de políticas, normas y procedimientos para poder efectuar un aseguramiento de calidad en lo que respecta al análisis y diseño de software. Entre los principales puntos a detectar en una institución por parte del desarrollador del sistema de información están:

1. **Canales informales.** - Identificar las interacciones que existen entre personas y departamentos, que muchas veces no están descritos en los procedimientos de operación.
2. **Interdependencias.** - Identificar de que departamentos y componentes de la organización depende un elemento en particular.
3. **Personas y funciones clave.** - Identificar las personas y elementos más importantes en el sistema para que éste tenga éxito.
4. **Enlaces críticos de comunicación.** - Identificar cómo es el flujo de información e instrucciones entre los componentes de la institución.

La finalidad de los sistemas de información dentro de cualquier institución o empresa son: procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas que coadyuven en la toma de decisiones y benefician las operaciones de los negocios de cualquier organización.

### 3. Objetivos de los sistemas de información

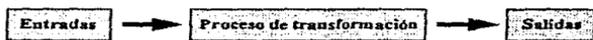
Los objetivos a cumplir por cualquier sistema de información se describen en la tabla siguiente.

Objetivo	Descripción
Especificar los elementos de diseño lógico.	Especificaciones detalladas de diseño que describen las características de un S.I. Entradas, salidas, archivos, base de datos y procedimientos.
Actividades de soporte para la empresa.	Los resultados del empleo del sistema serán de ayuda para mejorar el rendimiento de la empresa. El diseño debe ajustarse a la forma en que la compañía conduce sus actividades. La tecnología es secundaria en relación con los resultados obtenidos con el uso del sistema.
Satisfacer los requerimientos de los usuarios.	Satisfacer las necesidades de los usuarios en términos de: <ul style="list-style-type: none"><li>• Efectuar en forma correcta los procedimientos apropiados.</li><li>• Presentar en forma apropiada la información.</li><li>• Proporcionar resultados exactos.</li><li>• Utilizar los métodos de interacción apropiados.</li><li>• Proporcionar confiabilidad total.</li></ul>
Fácil de usar.	Ingeniería humana favorable. El diseño ergonómico debe ser físicamente cómodo y contribuir a la efectividad y eficiencia del usuarios.
Proporcionar las especificaciones de software.	Especificar los componentes y funciones con suficiente detalle para construir el software de aplicación.
Ajustar a los estándares de diseño.	El diseño y su especificación deben estar en concordancia con las reglas y prácticas establecidas por la organización.

#### 4. Fundamentos de los sistemas de información

La base de los sistemas de información se encuentra sustentada en los conceptos relacionados con sistemas, éstos permiten entender apropiadamente el desarrollo de sistemas de información, su tecnología, sus distintas aplicaciones y la administración de los mismos. Por consiguiente primero debemos establecer qué se entiende por sistema.

Un sistema se concibe como un grupo de componentes interrelacionados trabajando en conjunto encaminados hacia una meta común, aceptando entradas y produciendo salidas dentro de un proceso de transformación organizado.



Sistema

Figura 2. Esquema general de un sistema

Por lo tanto todo sistema está constituido por tres bloques principales que son: las entradas, el proceso a realizar y las salidas. A continuación se menciona en qué consiste cada uno.

**Entradas** - Este bloque contempla la captura y conjunción de los elementos que sirven de entrada al sistema. Por ejemplo, son los materiales, la energía, la información y los datos.

**Proceso** - Este bloque contempla el proceso de transformación que convierte las entradas en salidas.

**Salidas** - Este bloque involucra los elementos producidos por el proceso de transformación. Por ejemplo, los productos terminados, los servicios en general, el software producido.

### *Sistema de información*



Figura 3. Representación de un sistema referido a sistema de información con el uso de las computadoras

Frecuentemente un sistema puede tener dos componentes adicionales que son la **retroalimentación** y el control. El primero relacionado a los sistemas provee de información concierne al desempeño de la organización y el segundo monitorea y evalúa la retroalimentación, es decir, determina si el sistema cumple con las expectativas esperadas o necesita ajustarse para proporcionar las salidas adecuadas. Lo importante a relacionar en el contexto de los sistemas de información es precisamente el procesamiento de datos y operaciones de la organización. Los datos son atributos medibles de las distintas entidades de cualquier sistema, éstos pueden representarse en distintas formas como números, letras, texto en general, voz e imágenes.

Actualmente el esquema fundamental de los sistemas de información comprende los recursos de hardware (Pc y multimedia), software (programas y procedimientos) y las personas como especialistas de cómputo y usuarios finales; en conjunto representan las entradas, el procesamiento, las salidas, su almacenamiento y ejecutan el control

de actividades que convierten los recursos en productos de información para la toma de decisiones.

### **Modelo de sistemas de información**



Figura 4. Modelo de un sistema de información

### **Recursos utilizados por un sistema de información**

Los recursos de los sistemas de información se clasifican en cuatro grandes grupos que son: hardware, software, personal y los tipos de datos a procesar. A continuación se describe cada uno con su correspondiente contribución en las actividades del procesamiento de datos de los sistemas de información.



**Recursos de un sistema de información**

Figura 5. Recurso que utiliza un sistema de información

### Recursos de hardware

Dentro del hardware se contemplan todos los dispositivos electrónicos (dispositivos físicos y materiales) utilizados en el procesamiento de la información y los dispositivos electromecánicos (Por ejemplo bombas, sensores, motores) que proporcionan las funciones del mundo exterior. Específicamente como dispositivos electrónicos se tienen las computadoras personales, mainframes, workstations, dispositivos de almacenamiento como cintas magnéticas, discos duros, cartuchos y todos los dispositivos de redes de comunicación tales como concentradores, ruteadores, módem, cables de conexión, tarjetas de red.

### Recursos de software

El software incluye todo el conjunto de instrucciones para el procesamiento de la información. Esta definición contempla al conjunto de instrucciones de control de una computadora, mejor conocidos como programas, e introduce los procedimientos a efectuar por el personal correspondiente. El software típicamente se clasifica en dos grupos.

- ◆ **Software de sistema** - Son todos los programas que administran y soportan los distintos recursos de una computadora. Estos se subdividen en programas de administración de sistemas, programas de soporte de sistemas y programas de desarrollo de sistemas. Figura 6
  - ◆ Los programas para administración de sistemas incluyen a los sistemas operativos, sistemas administradores de base de datos, monitores de telecomunicaciones.
  - ◆ Los programas de soporte de sistemas incluyen a las utilerías de sistemas, monitores de seguridad.
  - ◆ Los programas de desarrollo de sistemas incluyen los lenguajes de programación, editores y debuggers, programas generadores, CASE.
- ◆ **Software de aplicación** - Son aquellos programas de aplicación que procesan directamente información de uso general o específico. Fig 6
  - ◆ Dentro de los programas de uso general se tiene a los procesadores de palabras, manejadores de bases de datos, hojas de cálculo, accesorios de escritorio.
  - ◆ Dentro de los programas de uso específico se tiene a los de análisis financieros, análisis económico, manufactura, CAD, de entretenimiento.
- ◆ **Procedimientos** - Son las instrucciones de operación para el personal que usará el sistema de información. Como ejemplos tenemos al llenado de formas o el uso de un paquete.

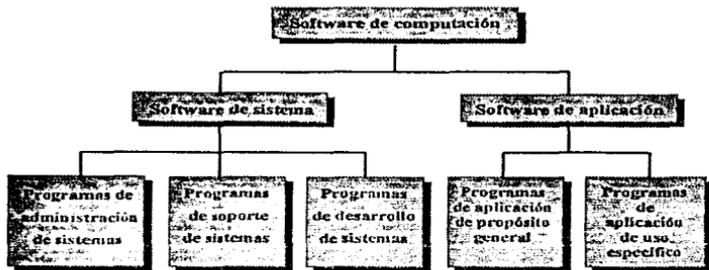


Figura 6. Diagrama de clasificación del software

### Recursos humanos (personal en general)

Todo sistema de información debe de ser utilizado y operado por personal de la organización para la que fue creado. Suele agruparse en dos tipos comúnmente.

- **Especialistas** - Compuesto por las personas que desarrollan y operan el sistema de información. Incluyen a los analistas de sistemas, programadores, operadores de computadoras.
- **Usuarios finales** - Compuesto por los usuarios o personas que utilizan los resultados obtenidos del sistema de información. Incluyen a los directivos de la organización.

### Recursos y manejo de información

La información y datos que serán procesados y analizados en el sistema de información suelen almacenarse en bases de datos, esto gracias al desarrollo de la computación, y pueden considerarse como información a datos tales como: simbología alfanumérica, gráficas, texto, imágenes, audio y video.

La información se define desde el punto de vista de sistemas como hechos y datos para ser utilizados en procedimientos, esto involucra su transformación para que tenga sentido, significado y sea útil a los usuarios finales. De hecho la información se ha convertido en un recurso de valor organizacional, ya que instituciones la consideran como uno de los activos de la organización. Para que la información generada del sistema sea útil debe poseer ciertos atributos o características, a éstas se les nombra dimensiones de tiempo, de contenido y de forma.

#### *Dimensión de tiempo*

Esta dimensión se basa en que toda información debe ser susceptible al tiempo. Además los sistemas de información deben ser capaces de proveer de información cada vez que se desee (demanda de informes), cada vez que ocurran condiciones específicas (informes especiales) y debe producirse en intervalos de tiempo (informes periódicos). Entre sus características a cumplir están:

Oportuna	La información debe proporcionarse cuando se necesite.
Al corriente	La información debe estar al día.
Frecuente	La información debe proporcionarse tan seguido como sea necesario.
Periódica	La información debe referirse tanto al pasado, como al presente y al futuro.

#### *Dimensión de contenido*

Esta dimensión es considerada como la más importantes de las tres. Entre sus características están:

Exacta	La información debe estar libre de errores, su realidad debe concordar con los datos procesados.
Apropiada	La información debe adecuarse a las necesidades reales del usuario, referirse a una situación específica.
Completa	Toda la información que sea necesaria debe de ser proporcionada.
Concisa	Sólo la información que se necesite debe de proporcionarse, es decir, debe ser no redundante (única).

<b>Estructurada</b>	La información puede estar estructurada ampliamente o estrechamente dependiendo del enfoque que se le dé.
<b>Funcional</b>	La información a través de reportes puede revelar el desempeño de las actividades, de los progresos logrados y de los recursos invertidos.

*Dimensión de forma*

Esta dimensión se basa en la forma en que debe presentarse la información (informes, pantallas, gráficas), ya que si es atractiva para el usuario ésta será fácil de comprender y de ser utilizada. Entre sus características están:

<b>Clara</b>	Debe presentarse en una forma que sea fácil de entender.
<b>Detallada</b>	Debe presentarse en detalle o en forma resumida.
<b>Ordenada</b>	Debe presentarse en una secuencia predeterminada.
<b>Presentada</b>	Debe mostrarse en texto, gráficas, formas etc.
<b>Dispuesta</b>	Debe presentarse en distintos medios de almacenamiento, como por ejemplo impresa en papel, en video etc.

En general los sistemas de información involucran primeramente al personal de la compañía, las computadoras, los programas, los procedimientos a ejecutar, las redes de comunicación y el personal que mantendrá el sistema. Las personas que interactúan al realizar un sistema son: gerentes, ingenieros industriales, ingenieros en computación, ingenieros electrónicos, administradores, personal operativo, etc..

También los sistemas de información se identifican como un conjunto de componentes o variables organizados, interdependientes, integrados y que interactúan con los usuarios de una organización. Al diseñar un sistema de información deben establecerse claramente las guías a observar a través de objetivos y metas comunes a cumplir en la organización. Éstas permitirán proclamar la misión a seguir de la compañía, las actividades a realizar y los trabajos a

desempeñar para alcanzar las metas. El ambiente que rodea a un sistema determina en cierta forma el desempeño de éste, sin embargo el ambiente es externo al sistema debido a que abarca todo lo que está fuera del control del sistema. En cambio los recursos son internos al sistema por lo que están bajo su control, además se encuentran disponibles para cualquier actividad, son un medio para alcanzar las metas trazadas.

Durante la fase de elaboración de un sistema debe contemplarse su administración en base a las actividades propias de la planeación y control. La planeación comprende el establecimiento de las metas, el uso de recursos y el desarrollo de un programa de actividades; el control es indispensable para una correcta ejecución de la planeación.

A continuación se muestran los diez puntos básicos que fundamentan la teoría general de sistemas:

1. Determinar las relaciones entre los componentes.
2. Un sistema es visto como un todo, es decir, no es necesario dividirlo en partes constitutivas ya que se puede perder de vista al sistema completo.
3. Los sistemas, en cierta forma, son alcanzar una meta u objetivo; así como la interacción de sus componentes en un estado final.
4. Los sistemas tienen entradas y salidas.
5. Todos los sistemas transforman entradas en salidas, por lo general las salidas difieren de las entradas.
6. Los sistemas exhiben entropía. La entropía describe el estado de un sistema cerrado (ninguna entrada desde afuera del sistema), donde todos los elementos se mueven hacia la desorganización y la incapacidad para atraerse; y las entradas del proceso no están preparadas para producir salidas, así el procesamiento de información se vuelve crítico para la sobrevivencia del sistema.

7. Los sistemas regulan su interacción con los componentes con el fin de alcanzar los objetivos. La planeación, control y retroalimentación son funciones asociadas a esta regulación.
8. Los sistemas comunmente están constituidos por pequeños subsistemas. La conformación de grandes sistemas en subsistemas se considera como una característica de la teoría de sistemas.
9. Los sistemas complejos exhiben diferenciación, unidades especiales ejecutan tareas especiales.
10. Los sistemas exhiben equifinalidad; los estados finales pueden ser alcanzados por diferentes alternativas. en otras palabras, existen diferentes maneras para conseguir los objetivos de un sistema.

El enfoque de análisis y diseño que se le da a cualquier sistema se basa en los puntos anteriores; su importancia radica entendiendo el sistema en general y modificándolo, en cierta manera, al problema en particular. Ver tabla 1

Normalmente al definir el problema se realizan preguntas acerca de los objetivos y lo satisfactorio que serán las salidas del sistema. En la fase de diseño suelen hacerse preguntas tales como: Cuáles son las variables del sistema, cómo se relacionan las variables entre si y con el medio que afectarán, cuáles son los límites del sistema, en otras palabras, se define su extensión

Tabla 1

Teoría general de sistemas	Importancia para el diseño de sistemas de Información
Interacción de componentes de un sistema.	Durante el análisis se identifican entidades y sus interrelaciones.
Un sistema es un todo.	Se asegura en definir al sistema entero antes de examinarse en subsistemas.
Los sistemas buscan un fin.	Identificar cuál es el objetivo del sistema de información.
Los sistemas tienen entradas y salidas.	La tarea de diseño es especificar entradas y salidas claramente.
Los sistemas transforman entradas para producir salidas.	Se diseñan y se especifican tareas de procesamiento para producir las salidas deseadas.
Los sistemas exhiben entropía.	El manejo y procesamiento de la información es crítico para el éxito de una organización.
Los sistemas deben ser controlados.	Los sistemas de información ayudan a controlar cualquier proceso dentro de una organización.
Los sistemas forman una jerarquía.	El diseño de sistemas de información es una tarea jerárquica, es decir, están conformados en jerarquías de subsistemas.
Los sistemas exhiben diferenciación.	Los sistemas de información tienen partes especializadas.
Los sistemas exhiben equifinalidad.	Existen diferentes caminos para alcanzar los objetivos.

### 5. Puntos a destacar dentro de un sistema de Información

Hoy en día las instituciones o empresas reconocen a la información como un recurso corporativo. Precisamente dichas organizaciones establecen la fuente de información y la distribuyen interiormente de acuerdo a las responsabilidades grupales o individuales; la interpretación y la toma de decisiones estará basada en dicha información, ésta resulta un elemento clave en la búsqueda y solución de los problemas de la empresa.

En los últimos años el desarrollo, implantación y aplicación de los sistemas de información han llegado a convertirse en una estrategia esencial en las grandes empresas; sin embargo, para que éstos sean aprovechados exitosamente, deberán cumplir con los puntos siguientes:

- ◆ Conocer específicamente la visión del negocio.
- ◆ Contar con toda la información importante del aspecto a analizar.
- ◆ Clasificar y actualizar la información.
- ◆ Estar desarrollados en una herramienta que permita su correcta explotación y se adecue a las necesidades del usuario, así como a las expectativas de crecimiento de la información, en general a las características o arquitecturas de sistemas y hardware de la empresa.

## 6. Categorías de los sistemas de Información

Los sistemas de información de cualquier organización deben estar enfocados al apoyo de los procesos de la empresa. Existen distintos tipos de sistemas de información para satisfacer las necesidades corporativas de cada empresa, entre los niveles de clasificación considerados comúnmente de acuerdo a su importancia están:

- ◆ Sistemas de información ejecutivos (SIE)
- ◆ Sistemas de información para el soporte de decisiones (DSS)
- ◆ Sistemas de información para el proceso administrativo (MIS)
- ◆ Sistemas de información para el procesamiento de transacciones (TPS)

Estos niveles pueden ser vistos en una pirámide como se muestra en la siguiente figura 7.

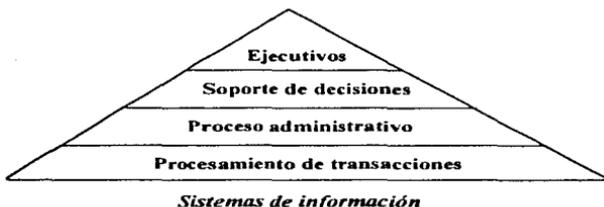


Figura 7. Tipos de sistemas de información

A continuación se mencionan cada uno de ellos con sus respectivas características.

#### **Sistemas de información para el procesamiento de transacciones (TPS)**

Actualmente la mayoría de los sistemas para el procesamiento de transacciones operan en línea, debido al creciente uso de tecnología de hardware y software. A este tipo de sistemas también se le conoce como *sistemas de procesamiento de transacciones en línea* (OLTP).

Este tipo de sistema tiene como finalidad mejorar las actividades rutinarias de una empresa y en general de aquellas de las que depende toda la organización, es decir, están orientados hacia operaciones de la organización. Una transacción es cualquier actividad que afecta a toda la organización. Como ejemplos de transacciones están: inventario, nomina, facturación, entrega de mercancías, pago de empleados, depósito de cheques, etc. Dentro de las actividades que incluye el procesamiento de transacciones se encuentran las siguientes:

- ◆ Cálculos mediante la manipulación de la información.
- ◆ Clasificación de la información.
- ◆ Ordenamiento de la información.
- ◆ Almacenamiento y recuperación de la información.
- ◆ Generación de resúmenes.

Las actividades mencionadas forman parte del nivel operacional de cualquier institución o empresa. De hecho se puede decir que es la relación entre los sistemas de información y los niveles de una organización. Ver la figura 8

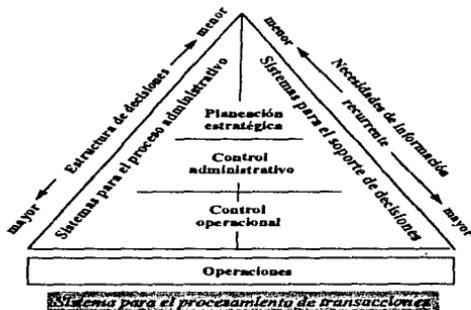


Figura 8. Relación entre sistemas de información y los niveles de una organización

Existen características similares entre las instituciones o empresas, éstas permiten establecer rutinas para el manejo de transacciones. Entre las principales características que se encontraron están:

1. Las aplicaciones operan sobre confiables y robustas plataformas.
2. Las aplicaciones garantizan que las acciones efectuadas son correctas y que ningún dato se perderá.
3. Las aplicaciones deben responder en forma rápida y consistente.
4. Las aplicaciones generan seguridad beneficiando la naturaleza misma de las funciones de la empresa.
5. Manejo de gran cantidad de información.
6. Gran volumen de transacciones.
7. Gran similitud de transacciones.
8. Los procedimientos para el procesamiento de transacciones están bien comprendidos y se pueden describir con detalle.
9. Existen muy pocas excepciones a los procedimientos normales.

Las rutinas describen que realizar en cada transacción, los pasos y procedimientos a seguir y lo que debe hacerse en caso de que se presente una excepción. A los procedimientos que procesan las transacciones se denominan "*procedimiento de operación estándar*".

### **Sistemas de información para el proceso administrativo (MIS)**

Este tipo de sistemas están enfocados a los directivos o gerentes; ayudándolos en la toma de decisiones para la solución de problemas de la empresa. Entre sus características podemos destacar las siguientes:

- ◆ Se recurre a la información almacenada en las bases de datos de la empresa, como consecuencia del procesamiento de transacciones.
- ◆ Identificar la información necesaria para formular decisiones en forma periódica (a la semana, al mes, al trimestre).
- ◆ La información se presenta en un formato diseñado con anterioridad. Ejemplo: reportes, elaboración de estados financieros, etc..

- ◆ Las decisiones apoyadas en estos sistemas suelen nombrarse como decisiones estructuradas; Son estructuradas ya que se conocen los factores o variables que influyen significativamente sobre el resultado de una decisión.
- ◆ Es posible anticipar los requerimientos de información.

### **Sistemas de información para el soporte de decisiones (DSS)**

Este tipo de sistemas ayudan a los directivos a tomar decisiones no muy estructuradas, es decir situaciones especiales. La decisión se considera no estructurada al no existir procedimientos claros para tomarla y no es posible identificar con anticipación los factores claves en la decisión. Hacen uso de modelos matemáticos y estadísticos, o de conocimiento específico a un problema particular. Entre sus características se encuentran:

- ◆ El factor clave es determinar e identificar la información necesaria.
- ◆ Típicamente recurren al manejo de operaciones del procesamiento de transacciones, pero usan la información para proyecciones y evaluación de escenarios.
- ◆ En ocasiones cuentan con herramientas de análisis (qué pasa si), para modelar los resultados del proyecto.
- ◆ Tienen gran flexibilidad ya que al contar con información puede conducir a otros requerimientos.
- ◆ El usuario al solicitar informes define su contenido y especifica la forma para producir la información.
- ◆ El criterio de los directivos es parte fundamental ya que los sistemas de soporte de decisiones son sólo herramientas de ayuda.

## **Sistemas de información ejecutivos (SIE)**

Estos sistemas proveen de toda la información necesaria a los ejecutivos de alto nivel para ayudarlos en la toma de decisiones. Estos sistemas permiten un acceso rápido y efectivo de la información compartida; Utilizan interfaces gráficas que sean de fácil uso para la interpretación de resultados. Además estos sistemas proveen sumarios de información interna y externa de la empresa. También se enlaza con otro tipo de aplicaciones tales como: correo electrónico, sistemas manejadores de bases de datos, hojas de cálculo, procesadores de palabras y sistemas para el control y administración de proyectos. Entre sus características a nivel usuario se encuentran:

- ◆ Visualizan la información en gráficas y tablas, proporcionan utilerías con opciones a cambiar el tipo de gráfica, el contenido de la misma, formatos utilizados y otras características de estilo e información.
- ◆ Proporcionan mecanismos automáticos para consolidar información dentro del SIE y para realizar cálculos con ella.
- ◆ Facilitan al usuario la consulta abierta de información a través de consultas "Queries" creadas por el propio usuario.
- ◆ Proveen de funciones preprogramadas para pronósticos y análisis del tipo (que pasa si), lo cual proporciona al usuario una flexibilidad en el análisis de sus datos.
- ◆ Cuentan con alarmas que ayudan al usuario a detectar valores fuera de los rangos comunes.
- ◆ Consolidan información provenientes de múltiples fuentes.

Como se puede apreciar un sistema de información ejecutivo puede decirse que esta conformado principalmente de dos partes: por un lado la interfaz con el usuario y por otro con las base de datos. Se dice que en la mayoría de las aplicaciones estos elementos se encuentran distribuidos en diferentes equipos e interactúan entre sí por medio de peticiones remotas; para el caso de la interfaz del usuario se le nombra *cliente* y al responsable de la manipulación de la información se le conoce como *servidor*.

Quando un sistema de información permite ver el estado actual y completo de la información que se maneja en toda la organización y más específicamente en cualquiera de sus áreas de trabajo se le nombra *Sistema integral de información*. Éste comúnmente se compone de distintos módulos, pudiendo ser un módulo por cada área de la institución.

## **7. Técnicas de sistemas de información**

Para solucionar problemas de una organización la elaboración de sistemas deberá ser concebida a través de un proceso de desarrollo, el cual deberá cubrir totalmente los requerimientos de la organización, y con el establecimiento de métodos y técnicas proporcionará el medio más apropiado para el uso eficiente de la tecnología de cómputo. Así varios autores reconocen al proceso de desarrollo como parte fundamental para el correcto aprovechamiento de los recursos de la institución, permite cumplir con los tiempos estimados del desarrollo del sistema y es una poderosa herramienta que define, administra y controla efectivamente al sistema.

Existen dos tipos de técnicas o enfoques para el desarrollo de sistemas de información basados en computadora, los cuales son:

1. Método del ciclo de vida para el desarrollo de sistemas.
2. Método del prototipo de sistemas.

### ***Método del ciclo de vida para el desarrollo de sistemas***

Este enfoque está constituido por un conjunto de actividades que los analistas, diseñadores y usuarios finales deben trabajar conjuntamente para desarrollar e implantar un sistema de información. Éste comúnmente consta de seis pasos generales, aunque varios autores suelen subdividirlo y quizá hacerlo más específico y detallado. Algunas veces cuando se decide utilizarlo dentro de una empresa se percibe que todas las actividades se encuentran relacionadas, por lo que para

muchos desarrolladores es difícil establecer los pasos en el orden correcto y es frecuente que omitan varios de ellos. Sin embargo, implícitamente ya siguieron la metodología.

El método del ciclo de vida para el desarrollo de sistemas esta conformado de las siguientes actividades:

1. Investigación preliminar del sistema
2. Análisis del sistema
3. Diseño del sistema
4. Programación del sistema
5. Verificación e implantación del sistema
6. Mantenimiento del sistema

#### Investigación preliminar del sistema

Esta sección se inicia con la solicitud para solucionar un problema de la institución o empresa. En esta etapa se elabora un estudio preliminar mejor conocido como el estudio de factibilidad, el cual tiene como objetivo investigar las necesidades del usuario, los objetivos a cumplir, las restricciones, vislumbrar los requerimientos, evaluar costos asociados, y entrever los beneficios que traerá a la institución; en si es detectar la viabilidad de la elaboración del sistema. Al realizar este estudio se evalúan las alternativas del sistema y se propone la más factible y conveniente para el desarrollo del sistema.

#### Análisis del sistema

En la fase de análisis del sistema se llevan a cabo varias actividades que se pueden tratar como una extensión de lo realizado en el estudio de factibilidad. Al final se pretende obtener los requerimientos para una solución del sistema. Comúnmente suelen conocerse como especificaciones o requerimientos funcionales, éstos consisten en determinar las necesidades de información requeridas para cada actividad del sistema (entradas, procesos a realizar, salidas). Idealmente se persigue establecer las propiedades del sistema de forma precisa, comprobable y formal.

### Diseño del sistema

En la fase del diseño del sistema se especifica el cómo el sistema debe acoplarse a los objetivos desarrollados en la etapa anterior, es decir aborda tanto el aspecto de diseño lógico y el diseño físico. Básicamente se resume en tres actividades principalmente que son: interfaz del usuario, diseño y manejo de datos y el diseño del proceso. En los párrafos siguientes se explicará en que consiste cada una.

- **Interfaz del usuario**

Esta actividad se concentra en buscar la interacción entre los usuarios finales y el sistema. Se pretende producir especificaciones en detalle tales como: pantallas de despliegue, reportes, formas o documentos.

- **Diseño y manejo de datos**

Esta actividad se enfoca en el diseño de la estructura lógica de las bases de datos y archivos a utilizar en el sistema. Se establecen las entidades, las relaciones, se especifican los elementos de datos, reglas de integridad.

- **Diseño del proceso**

Esta actividad se concentra en base al diseño de los recursos de software, en otras palabras los programas y procedimientos necesarios para el sistema.

### Programación del sistema

Esta fase consiste en la generación de programas y rutinas a partir del diseño, de tal forma que el sistema funcione adecuadamente de acuerdo con las especificaciones del diseño.

### Verificación e implantación del sistema

En esta fase se demuestra que el sistema satisface los requerimientos y funciona correctamente para todos los posibles datos de entrada. El objetivo es la eliminación de todas las condiciones y fallos inesperados del sistema. También involucra la capacitación de usuarios finales y de los especialistas que serán los encargados de operar el nuevo sistema.

### Mantenimiento del sistema

Esta fase involucra probar y evaluar el desempeño del sistema, así como realizar las modificaciones pertinentes. En resumen:

1. Corregir errores y modificar posibles deficiencias que no se descubrieron durante el desarrollo del sistema
2. Modificar el sistema para satisfacer los cambios de los requerimientos, adaptándolos a los cambios en el entorno y mejorando la eficiencia de operación y calidad general. Los recursos y actividades requeridos para asegurar que el sistema seguirá satisfaciendo las necesidades para el que fue creado se consideran como parte del mantenimiento.

### ***Método del prototipo de sistemas.***

Este método tiene como característica involucrar de forma más directa al usuario en la etapa de análisis y diseño; está constituido por software que acepta entradas, realiza procesos y produce información, ya sea impresa o presentada en pantalla. Suele utilizarse como primera versión del sistema por lo que permite evaluar el diseño y la información generada por el sistema, no olvidando que deben esperarse cambios a medida que el sistema es utilizado.

El método de prototipo puede referirse como un modelo piloto o de prueba; ya que su diseño evoluciona con el uso. La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que va revelando más información para la elaboración del nuevo diseño. Este proceso se repite las veces que sea necesario hasta poder identificar los requerimientos esenciales del diseño.

En general se concibe que el uso del método de prototipos tienen mayor utilidad bajo las siguientes condiciones:

- Se identifica sólo una parte de las características esenciales del sistema; las restantes son más difíciles de identificar a pesar del análisis de requerimientos.
- Con la evolución del sistema permitirá añadir una lista significativa de requerimientos que el sistema debe satisfacer.
- Los encargados de diseñar e implantar un sistema nunca han desarrollado uno con las características iniciales del sistema propuesto.
- Con la evolución de las versiones del sistema permite el refinamiento de las características.
- Los usuarios del sistema participan en el proceso de desarrollo.

Quando se utiliza el desarrollo de prototipos se observa que es un proceso iterativo, comienza con unas cuantas funciones y va creciendo o disminuyendo con el uso y la evolución del mismo. En resumen los pasos generales a seguir durante el desarrollo de prototipos son los siguientes:

1. Identificar los requerimientos de información que el usuario conoce junto con las características necesarias del sistema.
2. Desarrollar un prototipo que funcione.
3. Utilizar el prototipo anotando las necesidades de cambios y mejoras. Esto expande la lista de los requerimientos previamente conocidos.
4. Revisar el prototipo con base en la información obtenida a través de la experiencia del usuario y la evolución del sistema.
5. Repetir los pasos anteriores las veces que sea necesario, hasta obtener un sistema satisfactorio.

Cuadro resumen de las técnicas anteriores:

<b>Método del ciclo de vida de desarrollo de sistemas</b>	
<b>Descripción de actividades</b>	<b>Características de aplicación</b>
<ul style="list-style-type: none"> <li>• Investigación preliminar</li> <li>• Determinación de requerimientos</li> <li>• Diseño del sistema</li> <li>• Desarrollo de software</li> <li>• Pruebas de sistemas e implantación</li> </ul>	<ul style="list-style-type: none"> <li>• Requerimientos predecibles del sistema de información.</li> <li>• Manejable como proyecto.</li> <li>• Los datos se encuentran en archivos y bases de datos.</li> <li>• Gran volumen de transacciones y procesamiento.</li> <li>• Requiere de validación de los datos de entrada.</li> <li>• Abarca varios departamentos.</li> <li>• Desarrollo por equipos de proyectos.</li> </ul>

<b>Método de prototipo de sistemas</b>	
<b>Descripción de actividades</b>	<b>Características de aplicación</b>
<ul style="list-style-type: none"> <li>• Desarrollo iterativo o en continua evolución donde el usuario participa directamente en el proceso.</li> </ul>	<ul style="list-style-type: none"> <li>• Condiciones únicas de la aplicación donde los encargados del desarrollo tienen poca experiencia o información, o donde los costos y riesgos de cometer un error pueden ser altos.</li> <li>• Útil para evaluar la factibilidad del sistema, identificar requerimientos y evaluar el diseño de un sistema.</li> <li>• Los usuarios al participar en el proceso pueden señalar puntos más significativos para el sistema.</li> <li>• Un sistema se construye con rapidez, ya que se enfoca más en la velocidad de desarrollo que en la eficiencia en el procesamiento.</li> </ul>



## ORIENTACIÓN A OBJETOS

*El conocimiento nos conduce a lugares  
sin fronteras.*

**ESTA TESTS NO DEBE  
SALIR DE LA BIBLIOTECA**



## ORIENTACIÓN A OBJETOS

### Introducción

Una de las tendencias actuales en cuanto a software, y más específicamente en cuanto a sistemas de información, es la creación de sistemas corporativos que integren en uno solo todas las aplicaciones y consultas de información de las empresas. Bajo esta tendencia y con el propósito de satisfacer esta necesidad, los códigos tradicionales que conforman dichos sistemas crecen de manera estratosférica, lo cual crea problemas de comprensión en cuanto a código y estructura de programas, esto aunado a que los tiempos de programación y liberación de los sistemas aumentan considerablemente.

Por otra parte: es importante tener en cuenta que la alta calidad es esencial en el desarrollo del software, ya que una pobre calidad es un desperdicio de dinero, y si el desarrollo de un proyecto o una aplicación dura mucho tiempo, las corporaciones no pueden reaccionar ante la competencia de manera rápida.

Así pues; con el fin de obtener sistemas corporativos, en un plazo menor, con un menor costo, y que no sacrifiquen la confianza que se le pueda tener al software obtenido, se ha creado el concepto de la orientación a objetos. Esto puede implicar un cambio al diseño y a la estrategia tradicional de los sistemas, así como una nueva mentalidad por adoptar para los desarrolladores de sistemas tradicionales.

El análisis y diseño orientado a objetos; aunado con la programación orientada a objetos, tiende a combatir la complejidad y tendencia al *gigantismo* de los códigos tradicionales, esto a través de la reutilización del código en la conformación de

clases. Pero por otra parte implica un nuevo concepto en cuanto al análisis y al diseño de sistemas, los analistas deben conocer e identificar los objetos y manipular su comportamiento.

Las técnicas con orientación a objetos encuentran su mayor potencialidad al aplicarse a sistemas complejos, contrastando con las técnicas tradicionales. Esto puede aparentar un crecimiento en complejidad; sin embargo, no implica que el desarrollador deba conocer la estructura interna de los objetos, si bien los objetos tuvieron una programación previa de cierta complejidad, un desarrollador práctico deberá limitarse a conocer su comportamiento y su forma de uso.

Las técnicas de diseño y programación con orientación a objetos forman parte de un conjunto de nuevas tecnologías, que buscan tener las siguientes características en el software:

- Reutilización del código.
- Menor complejidad.
- Mayor capacidad de diseño en aplicaciones.
- Más flexibilidad en cuanto a manejo de código.
- Tener mayor rapidez en el desarrollo.
- Mayor facilidad en la modificación de código.
- Alta confiabilidad en los sistemas realizados.

En resumen; lo que se busca con una programación orientada a objetos, es lograr una aplicación del lenguaje con una nueva forma de ver la programación; así como lograr mayor potencia en cuanto al desarrollo (tiempo-programación contra desempeño del sistema), y también una flexibilidad en cuanto al código.

Tanto el diseño orientado a objetos; como la programación orientada a objetos, presentan una perspectiva diferente al diseño y programación estándar por procedimientos, ya que en lugar de pensar en un flujo de programa, se debe de pensar en la creación de objetos, componentes de objetos autocontenidos y que posean una funcionalidad interna definida y propia de ese objeto, así como de la funcionalidad que se presenta al usuario.

## 1. El por qué del término orientación a objetos

Debemos situar a los métodos con orientación a objetos, como el complemento para la formalización de la programación de lenguajes orientados a objetos, la cual por sí misma presenta una perspectiva de programación distinta a las demás. Esta perspectiva se puede interpretar como la orientación a la cual tiende el lenguaje por su diseño y naturaleza. De esta forma podemos encontrar otras orientaciones, como lo son la orientación a procedimientos; a la cual denominaremos en lo sucesivo como tradicional, y la orientación a datos.

Ahora bien; para la justificación del término objeto, debemos tomar en cuenta el concepto de abstracción, el cual es usado para referirse de manera genérica, tanto a datos como a operaciones y procedimientos. Bajo este término se representan a todas las cosas (objetos) en un plano de manejo común, en el cual puede interactuar un objeto dato, con algún objeto operación, mediante un objeto procedimiento. De esta manera puede "transportarse" a todos estos objetos a un mismo *nivel de abstracción*, el cual puede ser manejado por el lenguaje. De ahí el tomar el término de objeto como término genérico.

## 2. Metodología tradicional vs. Metodología con orientación a objetos

Uno de los principales contrastes que encontramos en las técnicas con orientación a objetos y las técnicas tradicionales, es que en estas últimas los desarrolladores

piensan en torno a los procesos y en su descomposición, mientras que en objetos se debe de pensar en los objetos y su comportamiento. Por otra parte, el análisis y diseño orientado a objetos presenta la ventaja sobre el análisis y diseño tradicionales, en cuanto a que estos últimos tienden a hacerse difíciles de controlar cuando el sistema comienza a crecer.

En algunas ocasiones las metodologías tradicionales, que son usadas para las etapas de análisis y desarrollo, difieren entre sí, y más aun en la etapa de programación; ya que cada etapa cuenta con diferentes modelos conceptuales. Por ejemplo, encontramos que los analistas usan modelos basados en entidad relación con funcionalidad definida; por su parte los diseñadores modelan a través de diagramas de flujo de datos o pseudocódigo, estructuras de tablas, etc., de igual manera los programadores modelan según el lenguaje que utilicen.

Existen algunos antecedentes en cuanto a la formalización del análisis estructurado, los cuales se apoyan fundamentalmente en los sub-programas y funciones como un mecanismo de abstracción para la construcción de sistemas complejos. Uno de los primeros métodos, con este enfoque, fue el de *descomposición funcional*. Este enfoque considera a un sistema como un conjunto de áreas con determinada función, las cuales pueden dividirse en procesos, expresando así módulos funcionales y su activación. Sin embargo, obliga a los programadores a centrarse en operaciones, prestando poca atención a la estructura de datos; el resultado es manejar más código y menos datos.

Otro antecedente es el enfoque de *suceso respuesta* el cual centra su atención en los sucesos externos para deducir los procesos del sistema, considerándolo como una caja negra para representar un proceso.

Por su parte; el método *orientado a objetos*, comienza por definir objetos, atributos, y posteriormente los ciclos de vida de cada objeto, y finalmente se definen las

interrelaciones que tienen los objetos. El resultado final de la aplicación de este método consiste en un mínimo código derivado de los datos.

En las técnicas de desarrollo anteriormente citadas, cada uno de los componentes de un sistema recibe una prioridad definida, dicha prioridad la resumimos a continuación,

**Prioridades en las técnicas de desarrollo de sistemas**

Prioridad	Descomposición Funcional	Suceso-respuesta	Orientado a objetos
1	Proceso	Control	Datos
2	Control	Proceso	Control
3	Datos	Datos	Proceso

Es evidente que un sistema orientado a objetos, la prioridad de componentes recae en el contenido de las entidades (datos) de los objetos. Además no se agrupan funciones por proceso, sino métodos, siempre y cuando éstos pertenezcan a una misma abstracción de datos. Lo que determina la secuencia de funcionamiento es el paso de mensajes entre objetos.

Otra de las ventajas que encontramos en las técnicas basadas en orientación a objetos, es en cuanto a la utilización de un mismo modelo conceptual para todo el equipo de desarrollo, es decir los analistas, diseñadores, programadores y hasta usuarios. De esta manera todos los involucrados en el sistema piensan en torno a los objetos que lo conforman, y contribuyen al establecimiento de clases, generalizaciones, encapsulamiento y eventos.

Esta nueva conceptualización hace que el análisis y el diseño no se delimiten de manera tajante y compartan procedimientos, lo que hace disminuir tiempos. Esto se puede apreciar cuando se involucran tecnologías CASE.

La orientación a objetos, para el desarrollador de sistemas tradicional; implica un nivel de abstracción que va más allá de los procedimientos y datos. Se propone que para descubrir la diferencia entre la programación basada en procedimientos y la programación basada en objetos puede usarse el siguiente método.

Leer las especificaciones del software a desarrollar. Si se subrayan las acciones (verbos) se estará persiguiendo un código por procedimientos, mientras que si se subrayan los nombres (sustantivos) se tendrá la tendencia a programar con orientación a objetos.

En la realidad; la noción de orientación a objetos implica una serie de conceptos y definiciones que conforman el paradigma de orientación a objetos y que serán tratados en este capítulo.

A continuación se muestra una tabla comparativa entre las metodologías de procedimientos y de orientación a objetos.

Metodología por procedimientos	Metodología con orientación a objetos
Consta de procedimientos y datos en el cuerpo del programa.	Consta de objetos que contienen tanto métodos como atributos.
Mediante el uso de rutinas aisladas se logran efectuar tareas determinadas.	A través de una clase, es posible ofrecer una serie de servicios a objetos de un tipo en particular.
De carácter descendente.	De carácter ascendente.
Descomposición funcional basada en subprocesos y procesos de niveles superiores.	Composición de clases basada en abstracción de datos.
En sus diagramas expresan juegos secuenciales de módulos interdependientes que comparten datos.	Definen, a través de los métodos de los objetos, un conjunto de módulos de comunicación independientes, pero con visibilidad limitada entre ellos.

### 3. Principios de la orientación a objetos

Una posible dificultad en la teoría de orientación a objetos, es en cuanto a la definición de sus términos, los cuales suelen ser muy extensos y algunos de ellos no están estandarizados. Esto puede representar un gran obstáculo para aquellos desarrolladores que intentan adoptar esta nueva forma de construir sistemas de cómputo.

Con el propósito de sentar un precedente firme de lo que es la orientación a objetos; presentamos a continuación una tabla de los conceptos básicos usados en este tipo de programación. Dicha tabla parte del concepto mínimo (objeto), pasando por los conceptos que se asocian a él, ascendiendo a conceptos de mayor complejidad.

◆ Objeto
<ul style="list-style-type: none"><li>• Composición de objetos</li><li>• Instancia de un objeto</li><li>• Estado de un objeto</li><li>• Ciclo de vida</li><li>• Persistencia</li><li>• Encapsulado</li></ul>
◆ Mensajes
<ul style="list-style-type: none"><li>• Métodos</li><li>• Eventos (tipos de)</li></ul>
◆ Clase
<ul style="list-style-type: none"><li>• Instancia de una clase</li><li>• Subclase</li><li>• Herencia</li><li>• Abstracción (nivel de)</li><li>• Polimorfismo</li><li>• Modularidad</li></ul>

## ◆ Objeto

En lo referente al cómputo y a la programación, y con la aparición de la interfaz de usuario de tipo gráfico (GUI), han surgido lo que en algunas bibliografías se denomina WIMP (Windows Icons, Mice and Pointers), que es acorde a la filosofía de objetos en cuanto a denominar como tales a estos elementos gráficos.

Dentro de un objeto encontramos procedimientos similares a los que están en los lenguajes convencionales, además de datos como: números, matrices, cadenas, funciones, etc.. Con base en ésto el programador ve a los objetos como módulos individuales que se interrelacionan unos con otros y conformarán en conjunto una determinada aplicación.

Algunos autores clasifican a los objetos en torno a su actividad refiriéndose a ellos como *objetos pasivos*, que actúan solamente bajo una petición, y *objetos activos o agentes*, que efectúan un seguimiento de los sucesos que ocurren en una aplicación y actúan de forma autónoma. Ejemplo de los primeros puede ser la alarma de un auto, que actúa bajo la petición de activarse, y ejemplo de los segundos puede ser un reloj despertador que suena todos los días a las 6:00 am.

Algunos autores pioneros en la orientación a objetos han establecido una diagramación para representar a los objetos, sin embargo día con día surgen nuevas formas de representación de objetos, que varía dependiendo de la metodología que se este aplicando en cuanto a modelado y al análisis.



Para fines demostrativos, un ejemplo de objeto puede ser el automóvil.

- **Composición de objetos**

Podemos encontrar objetos compuestos de objetos. Por ejemplo, en el caso de un automóvil, encontramos que por si solo es un objeto para transportarnos, sin embargo este objeto esta compuesto por otros objetos como lo son llantas, motor, carrocería, asientos, parabrisas, volante, palancas, faros, sistema de frenos, etc..

Un objeto puede contener información a través de variables, que pueden definirse mediante el término de *variables de instancia* o *variables modelo*, que contienen datos locales a ese objeto.



Un objeto puede estar compuesto de otros objetos.

- **Instancia de un objeto**

En la orientación a objetos el término de *instancia* se usa para expresar una subordinación en cuanto a los elementos usados; este caso, los objetos. Por ejemplo, un objeto puede ser automóvil, pero una instancia del objeto automóvil sería "el automóvil de Roberto". Dicho de otra forma, una instancia de objeto identifica y define a un objeto (en este caso automóvil) como un objeto en particular, (en este ejemplo, el automóvil de Roberto). Podemos afirmar entonces que "el automóvil de Roberto" es una instancia del objeto "automóvil".



En el caso de la programación, se puede definir un objeto "Pantalla" el cual servirá de base para definir un grupo de instancias de pantalla, estas podrían ser: "Pantalla de presentación", "Pantalla de menú", "Pantalla del modulo 1", "Pantalla del modulo 2", etc..

- **Estado de un objeto**

El *estado de un objeto* es un conjunto de enlaces entre objetos que se relacionan a un objeto en particular, esta relación es temporal y se da en un instante determinado. Esta relación puede estar definida por valores de los atributos.

Se dice que un objeto puede existir en varios estados; por ejemplo, el objeto "automóvil" puede estar ligado a otro objeto, por ejemplo "taller mecánico",

"corralón", "agencia automotriz" etc., De ahí que el objeto automóvil pueda considerarse "descompuesto", "con multa", "nuevo" o "seminuevo", etc. es decir bajo un estado en particular. Tales estados pueden estar asociados a un ciclo de vida del objeto en particular; por ejemplo, el objeto "automóvil" primeramente debió de estar en el objeto "amadora de autos" posteriormente pasar al objeto "agencia automotriz" para luego pasar al objeto "compañía López", y así consecutivamente, hasta llegar al objeto "deshuesadero" y finalmente al objeto "fundidora", para de ahí posteriormente pasar de nuevo a una "armadura de autos". Es claro que un cambio del estado de un objeto implica que este pase de una categoría a otra, éste cambio se dará en función a un *evento* en particular.

En todo este hipotético ciclo de estados de un objeto, el objeto fue ligado a otros objetos; por ejemplo, el objeto "automóvil" puede estar en el objeto "compañía López" pero a su vez pertenecer a la instancia "departamento de ingeniería de la compañía López sucursal Guadalajara", o estar implicado en un tipo de objeto en particular; por ejemplo, "autos del DF que no circulan los lunes".

- **Ciclo de vida de los objetos**

En la teoría con orientación a objetos, debemos tener en cuenta la secuencia de hechos que tiene un objeto:

Primero: Los objetos se crean cuando se necesitan.

Segundo: Los objetos intercambian mensajes, moviéndose éstos de uno a otro, procesando la información de objeto a objeto y de objeto a usuario.

Tercero: Finalmente, los objetos que ya no son necesarios se borran de la memoria.

- **Persistencia**

Entre las etapas de vida de un objeto, se establece que esta presente la creación, un intercambio de mensajes, y la destrucción del objeto. La persistencia refiere a la permanencia del objeto en la memoria de la computadora, después de la etapa de creación. Cuando un objeto ya no es necesario, es destruido y se recupera la memoria asignada, este proceso de recuperación de memoria es automático y es denominado *recolección de basura*. Cuando un objeto se crea para durar permanentemente, y está dispuesto para su uso en toda la ejecución del programa, se dice que es un objeto persistente.

- **Encapsulado**

Es el concepto, resultado, o acto de tener todo el conjunto de datos y métodos con que trabaja un objeto de manera empaquetada y oculta de tal forma que otros objetos no puedan ver su contenido y estructura.

- ◆ **Mensajes**

En la teoría de objetos los protagonistas de las acciones son los objetos. La forma en que los objetos tienen la "posibilidad de actuar" es mediante los *métodos* y los *mensajes*. Un mensaje es una solicitud de un objeto para con otro objeto; en la cual se le pide que actúe de cierta manera. Las acciones entre objetos sucederán cuando un objeto reciba un mensaje; de esta forma el objeto emisor no necesita saber la manera en que el objeto receptor llevará a cabo la solicitud. A través de un mensaje se puede especificar que el objeto receptor trabaje de una manera en especial. Un objeto puede responder a un conjunto de mensajes al cual se le denomina *protocolo del objeto*.

- **Métodos**

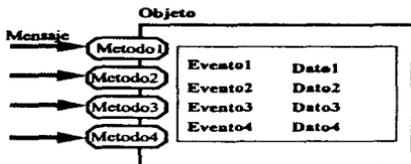
Los métodos son procedimientos u operaciones que residen dentro de un objeto, estos procedimientos pueden estar asociados a los eventos y determinan cómo actúa y reacciona el objeto cuando recibe un mensaje.

Los métodos se ejecutan en respuesta a mensajes previamente definidos en el código del objeto, y manipulan los valores de las variables locales a ese objeto (variables de instancia). De hecho, la única forma por la cual se deben cambiar las variables locales del objeto, debe ser mediante los métodos propios de ese objeto en particular.

En el ejemplo del objeto "automóvil" podemos activar la alarma, con esto el método "abrir puerta", estará asociado a un evento, éste podrá ser "señal audible de alarma". Entonces afirmaremos que el objeto "automóvil" que pertenece a la clase de medios de transporte "automóvil" realizará el evento "señal audible de alarma" en cuanto se dé el método de "abrir puerta".

En programación, un ejemplo de método puede ser el activar el código de salir de aplicación en cuanto se elige un botón de salida. Decimos entonces que la instancia "botón de salida" del objeto "botón" ejecutará el método "salir de la aplicación" en cuanto se efectúe el evento "clic" en el área del botón.

El método es la especificación de cómo llevar a cabo la operación, son las condiciones necesarias (filtros necesarios) para que el objeto pueda desencadenar un evento. De esta forma la estructura interior de los objetos permanece custodiada mediante los métodos, de tal manera que la estructura interior de los objetos permanece oculta a otros objetos, a los usuarios y a los programadores.



En este esquema, los métodos son el único camino tanto para modificar los datos, como para desencadenar los eventos asociados a el objeto. Los métodos a nivel programa son el código que implanta la operación.

### • Eventos

Los eventos son acciones que puede distinguir un objeto, y su ocurrencia produce el cambio de estado del mismo objeto.

Por ejemplo, en el caso del objeto "automóvil" tenemos que al activarse su alarma, estamos definiendo que al darse el método "abrir puerta" efectúe un evento para realizar cierta acción.

En teoría de objetos un evento establecerá que un objeto cambie de un estado (asociación de un objeto) a otro estado (asociación con otro objeto); este cambio puede ser por ejemplo, el cambio de una instancia del objeto a otra instancia, es decir, el objeto "automóvil de la compañía López" ha participado en el evento "choque entre periférico y viaducto", es por eso que pasará del objeto "estacionamiento de la compañía López" al objeto "taller Rodríguez".

### • Tipos de Eventos

Para distinguir diferentes tipos de eventos que pueden darse entre objetos, podemos generalizar los siguientes:

- Creación de objetos
- Terminación de objetos

- Incorporar un objeto como instancia de otro objeto
- Desincorporar un objeto como instancia de otro objeto
- Cambio de contexto de un objeto
- Cambio de atributo de un objeto

Es importante mencionar que el cambio de un evento puede estar relacionado con otros eventos. En las modificaciones de un sistema, con orientación a objetos, es común que estos cambios incidan de manera grave debido a la reutilización de código, ya que muchos objetos pueden estar dependiendo de ese código, por lo que pueden verse afectados si no se tiene un control adecuado.

#### ◆ Clase

Una clase es una descripción que engloba un conjunto de objetos que tienen características muy similares; una clase contendrá, tanto métodos como características comunes de ese grupo de objetos.

El definir una clase significa situar código reutilizable en un depósito común en lugar de tener que escribirlo una y otra vez para cada objeto.

La clase definirá características referentes a la apariencia y el comportamiento de un objeto. En el caso de *clases* de medios de transporte podemos distinguir a la clase "automóviles", conformado por un grupo de objetos como pueden ser: "automóviles de lujo", "automóviles austeros"; en los dos casos los *objetos* pertenecen a la clase de medios de transporte "automóviles" pero presentan características distintas en carrocería, motor y otras.

Para programación podemos definir como objeto un botón, ese botón puede ser, dependiendo de su clase, más grande, más chico, o de un color en especial.

Las clases presentan algunos conceptos relacionados como los siguientes:

- **Instancia de una clase**

Al igual que en las instancias de objetos, encontramos instancias de clase. Se dice que un objeto es instancia (o modelo) de una clase, cuando el objeto proviene de ella.

Por ejemplo, la subordinación que tienen los objetos para con las clases; podemos definir de entre las clases de medios de transporte una en particular, la clase "automóvil" (anteriormente denominado objeto), y que dentro de ésta tenemos los objetos "automóvil vagoneta", "automóvil compacto", "automóvil de lujo", etc. Es claro que todos los objetos son automóviles, sin embargo, pertenecen a una sola clase, la clase "automóvil", que se distingue de entre una agrupación de "clases" de medios de transporte. Con base en esto decimos que el objeto "automóvil compacto" es una instancia de la clase "automóvil".

Al igual que los objetos que contienen variables locales, las clases tienen variables de clase, que son valores que se comparten con todos los objetos que surgen de esa clase en específico.

- **Subclase**

De la misma forma en que una clase puede resumir un conjunto de características en un conjunto de objetos, también es posible resumir ciertas características de una subclase, el resultado de esto seguirá siendo una clase, pero estará conformada de resúmenes parciales de características comunes entre objetos.

La ventaja del uso de subclase es el poder describir las aplicaciones como un conjunto de módulos generales que simplifiquen la complejidad, elevando a los métodos y datos compartidos, de tal forma, que puedan ser accesibles a todos los objetos desprendidos de las subclases de la clase general

- **Herencia**

La herencia es el mecanismo que permite compartir, en forma automática, métodos y atributos entre clases, subclases y objetos. La forma de aprovechar la herencia, en la orientación a objetos, es definiendo solamente las diferencias con la clase principal. Si la clase tiene métodos inadecuados para la aplicación, entonces el programador podrá definir un nuevo código para ese objeto, o si es necesario, podrá definir una subclase en específico.

La herencia, en orientación a objetos, implica la definición de árboles de jerarquías de clases que permitan una especificación, también implican los conceptos de herencia múltiple y herencia simple.

- Herencia simple**

- La herencia simple implica heredar atributos y métodos solamente de una clase en específico.

- Herencia múltiple**

- La herencia múltiple implica la posibilidad de que una subclase pueda adquirir métodos y atributos de más de una clase, conformando así un comportamiento compuesto por más de una rama de una jerarquía de clases.

- **Abstracción**

Los conceptos ya mencionados fomentan que los desarrolladores piensen en términos abstractos. Ésto está presente en las superclases que buscan un comportamiento común entre objetos, el establecimiento de bibliotecas de clases que centralizan el código común de los objetos, el mecanismo que establece la herencia, etc. Todos estos conceptos marcan un determinado nivel de abstracción que contribuye a facilitar el trabajo de los desarrolladores.

- **Polimorfismo**

Como se mencionó anteriormente, los objetos reaccionan mediante mensajes. Cuando varios objetos tienen la capacidad de reaccionar de una manera en particular al recibir un mensaje en común, se afirma que está presente el polimorfismo. De esta manera es posible establecer un mensaje genérico para un grupo de objetos, y evitar tener que manejar un mensaje para cada uno de ellos.

- **Modularidad**

Una forma de trabajar el software; ya que ha alcanzado una complejidad exagerada, es mediante la modularidad. Por una parte encontramos a los métodos, que deben incluir cambios de estados en los objetos y, por otra, encontramos que bajo estos métodos deben de aislarse las operaciones de su causa y de su efecto. Bajo este punto de vista en cuanto a modularizar objetos y métodos, que hacen que las operaciones actúen de manera aislada, se tiene un mejor control, ya que ante un objeto de mucha complejidad, se pueden usar mediante solicitudes relativamente fáciles. Esto beneficiará en su uso y en su mantenimiento.

#### **4. El Modelado en la orientación a objetos**

Al igual que muchos procesos en ingeniería, una manera clara de tener una idea de cómo se conforma un sistema o una problemática, es a través del modelado del mismo. Entre más aproximado se encuentre el modelo del sistema, al sistema real, nuestras conclusiones y valoraciones serán más efectivas y congruentes para dar solución a la problemática o sistema reales en estudio. Dicho de otra forma, el modelo representará un aspecto de la realidad y nos ayudará para la comprensión de ésta

En la etapa de análisis orientado a objetos, se modela la problemática a resolver en términos de objetos y lo que les ocurre a éstos. En forma global, la siguiente figura nos muestra la forma de construir sistemas:



1. -El analista concibe un modelo del área de interés.
2. -El modelo concebido se convierte en un diseño.
3. -El diseño es transformado en código.

Es conveniente que los modelos sean comprensibles para los usuarios, fomentando su creatividad y la representatividad de los problemas a solucionar.

Para apoyar a la programación orientada a objetos, las etapas de análisis y diseño deben ser orientados en:

- La identificación de objetos y la definición de clases.
- La organización jerarquizada de clases.
- La reutilización de clases.
- La construcción de marcos estructurales de aplicación a partir de librerías de clases.

Además deberá tenerse en cuenta que la parte integral de los diseños orientados a objetos esta conformada de la reutilización, por la modularidad, la encapsulación y, la fuerte interrelación análisis-diseño y código.

Debemos considerar que no existe, a la fecha, un método de diseño orientado a objetos que defina la diagramación o sus etapas de análisis-diseño como un estándar ampliamente aceptado. En este trabajo lo que se pretende es mostrar de forma general algunos parámetros a considerar en la etapa de desarrollo, con el fin de centrar una base sólida para la aplicación de un marco organizacional para el equipo de trabajo de sistemas.

El análisis y diseño orientado a objetos deben estar pensados en forma independiente a los lenguajes de programación.

#### **5. Perspectivas a considerar en el desarrollo con orientación a objetos**

El objetivo por alcanzar de un buen diseño, es el de aprovechar al máximo las ventajas de un lenguaje orientado a objetos, además de producir menos código y más reutilizable. El resultado de todo esto determinará qué tan flexible será el sistema.

El diseño orientado tiene por objeto definir las clases, cada una de ellas está integrada por un conjunto de métodos y una lista de objetos con características comunes. El diseño quedará completo únicamente cuando quede definida cada clase, método, y mensaje. El proceso de diseño puede ser incremental y comúnmente suele partir de un conjunto de clases preexistentes.

Una metodología que soporte a la etapa de diseño, debe proporcionar reglas para definir y organizar clases, métodos y mensajes; así como, proporcionar estrategias de organización para librerías de clase.

Para dar inicio al diseño se definen los objetos, después se identifican los aspectos comunes entre objetos, y partiendo de eso se definen las clases que serán subclases de otras clases conformando un nivel de abstracción cada vez mayor.

El nivel de abstracción más grande de todos se denomina "marco estructural". Este elemento expresa un conjunto de clases que representa un diseño completo para una familia de aplicaciones interrelacionadas entre sí.

Se puede definir la etapa de análisis-diseño orientado a objetos, en cuatro pasos:

1. Identificación y definición de objetos y clases.
2. Organización de relaciones entre clases.
3. Extracción de estructuras en una jerarquía de clases.
4. Construcción de bibliotecas de clases y marcos estructurales de aplicación reutilizables.

Comúnmente las metodologías combinan el método descendente del análisis estructurado y las reglas básicas de diseño orientado a objetos, o bien, comienzan con un diagrama de flujo de datos para iniciar la identificación de objetos.

## **6. Características y beneficios de las técnicas de orientación a objetos**

### **• *Análisis natural***

Cambia nuestra forma de pensar en cuanto a la programación. Una técnica con orientación a objetos es más natural que la empleada en diseño estructurado. Esto es debido a que el mundo está formado por objetos, dichos objetos presentan propiedades, como pueden ser el color, tamaño, olor, y más aún, podemos encontrar que objetos con propiedades distintas puedan servir de forma similar, como ejemplo, un vaso o una taza pueden contener un líquido, o distintos martillos pueden clavar un clavo, o diferentes autos pueden trasladarnos a algún lugar.

### **• *Explicación sencilla***

En ocasiones es más fácil una explicación, a los usuarios finales, en términos de objetos, eventos y mecanismos de activación, al igual que con los diagramas de orientación a objetos, que con los diagramas de flujo de datos

• **Reutilización**

Los sistemas con orientación a objetos suelen construirse a partir de objetos ya existentes, lo que implica alta reutilización en el código, menor tiempo de desarrollo, mayor control en cuanto al código y, por tanto, más confiabilidad.

• **Composición y complejidad controlada**

Posibilidad de construcción de más objetos a partir de otros objetos, abatiendo la complejidad, ya que ésta se lleva de manera integrada a cada uno de ellos, pero aislada del resto de los objetos.

• **Bibliotecas para el desarrollo y posibilidad de crecimiento**

Flexibilidad de un desarrollo de bibliotecas de objetos personalizados, o bien, comprados y con la posibilidad de integración con herramientas CASE

• **Control a partir de clases**

Partiendo de las clases, usando el control de métodos y de objetos contenidos en ésta, es posible un control de sistemas que presenten un funcionamiento correcto.

• **Desarrollo con CASE**

Algunas de las técnicas con orientación a objetos tienen soporte en torno a CASE, como una tendencia en herramientas de desarrollo integradas.

• **Flexibilidad a cambios o modificaciones**

Flexibilidad a los cambios y modificaciones en los sistemas. Debido a que si los procedimientos o el mundo real cambia, los objetos del sistema que se asocian a estos cambios, serán los únicos que se tengan que sustituir, o bien, modificar, sin afectar al resto del sistema. También se aprovechará para realizar continuos cambios o mejoras.

• **Modularidad**

Fomenta la modularidad haciendo muy claras las fronteras entre objetos, define una comunicación y oculta los detalles entre ellos.

• ***Incremento de calidad***

Los diseños con orientación a objetos tienden a la calidad, puesto que se conforman con objetos que habrán sido verificados por su distribuidor o por anteriores proyectos.

• ***Integridad inherente***

Debido a la naturaleza de los objetos, de poder ser accedidos únicamente mediante los métodos establecidos, garantizan la integridad de los datos que sean almacenados en los objetos.

• ***Compatibilidad con programación visual***

Una técnica de orientación a objetos puede apoyarse en la programación visual, aprovechando su compatibilidad conceptual y logrando con esto una interactividad intuitiva de navegación para el usuario final.

• ***Migración de aplicaciones existentes a orientación a objetos.***

Algunas de las herramientas con orientación a objetos podrán soportar la migración de aplicaciones no orientadas a objetos, siempre que éstas se ajusten a un contenedor orientado a objetos, de tal forma que la comunicación con ellas sea a través de mensajes con orientación a objetos estandarizados.



**PROPUESTA DE POLÍTICAS, NORMAS Y  
PROCEDIMIENTOS PARA LA ELABORACIÓN  
DE SISTEMAS DE INFORMACIÓN CON  
ORIENTACIÓN A OBJETOS, BAJO LA  
NORMA ISO 9000**

*Proyecta lo difícil  
partiendo de donde aún es fácil.  
Realiza lo grande  
partiendo de donde aún es pequeño.  
Todo lo difícil comienza siempre fácil.  
Todo lo grande comienza siempre pequeño.*

*Lin-756*



## **PROPUESTA DE POLÍTICAS, NORMAS Y PROCEDIMIENTOS PARA LA ELABORACIÓN DE SISTEMAS DE INFORMACIÓN CON ORIENTACIÓN A OBJETOS, BAJO LA NORMA ISO 9000**

### **1. Objetivo**

Establecer las políticas, normas y procedimientos que permitan homogeneizar el desarrollo de sistemas de información dentro de una institución o empresa.

### **2. Introducción**

El software es el objeto de análisis y estudio de una de las ramas más jóvenes de ingeniería. De hecho, podemos decir que la concepción misma del software, es decir, su naturaleza de construcción y desarrollo es lo que lo hace particularmente distinto. Entre las características fundamentales del software podemos mencionar que su naturaleza es abstracta debido a que su herramienta principal es el manejo de la información. Además el software se desarrolla, no se fabrica como cualquier producto en un sentido clásico; se crea mediante la transformación del poder intelectual y cerebral de los especialistas en el conocimiento.

Consecuentemente, siendo la información uno de los recursos más valiosos de cualquier institución u organización, es de gran importancia contar con sistemas que permitan su uso eficiente, en armonía con todas las áreas. Para garantizar que esto ocurra dentro de una institución o empresa es indispensable contar con *políticas, normas y procedimientos que permitan homogeneizar la elaboración de sistemas de información.*

### 3. Políticas para la elaboración de sistemas

La *política de calidad* se define como las directrices u objetivos generales que tiene una institución o empresa concernientes a la calidad, las cuales son emitidas por la dirección. De hecho, a través de la administración de la calidad se definen las actividades a desarrollar para cumplir con esa política de calidad. La dirección es la encargada de difundirla y principalmente de asegurar de que sea entendida, implantada y revisada periódicamente en su organización. [ISO 8402/3.1], [ISO 9001/4.1.3], [ISO 9004-1/4.2] Apéndice Norma ISO

#### 3.1. Objetivo

Establecer un marco de referencia general para garantizar que los sistemas de información que apoyan las tareas sustantivas de una institución o empresa, sean concebidos y desarrollados de una manera tal, que permitan su articulación para una adecuada interacción entre las áreas.

[ISO 9000-1/4.9.2], [ISO 9001/4.1.1] Apéndice Norma ISO

#### 3.2. Políticas

3.2.1. Toda elaboración de sistemas deberá estar orientada a satisfacer las necesidades de manejo de información para las funciones sustantivas de una institución o empresa; es importante concebir el diseño de dichos sistemas de manera que permitan su integración y consolidación en una *base de datos institucional* y *un banco institucional de sistemas*, en un futuro próximo.

[ISO 9000-1/4.1 inciso b], [ISO 9000-1/4.5] Apéndice Norma ISO

3.2.2. Toda elaboración de sistemas, tanto interna como externa, debe cumplir con las normas establecidas por el *comité de informática* de una institución o empresa. El cumplimiento de las normas es un requisito indispensable para considerar un sistema apto para su liberación definitiva.

[ISO 9000-1/4.1 inciso b], [ISO 9000-1/4.5] Apéndice Norma ISO

- 3.2.3.** Toda elaboración de sistemas, tanto interna como externa de carácter institucional, deberá estar avalada por un dictamen técnico de la dirección de informática, órgano que en unión con el *comité de informática* de la institución o empresa debe normar el uso y aprovechamiento de los recursos informáticos de acuerdo al reglamento interno de ésta. El dictamen técnico no será necesario para aquellos sistemas internos de carácter técnico especializado o específico de las *áreas* de la institución o empresa, sólo deberán apegarse a los estándares establecidos por el *comité de informática*.  
[ISO 9000-1/4.1 inciso b], [ISO 9000-1/4.5] Apéndice Norma ISO
- 3.2.4.** La elaboración de sistemas institucionales debe apegarse a los estándares en cuanto al uso de software. Cuando esto no sea posible, el área usuaria deberá solicitar un dictamen técnico a la dirección de informática de la institución o empresa, justificando plenamente el uso de las herramientas propuestas para el desarrollo.  
[ISO 9000-1/4.1 inciso b], [ISO 9000-1/4.5] Apéndice Norma ISO
- 3.2.5.** La contratación externa para la elaboración de sistemas deberá sujetarse a la normatividad de adquisiciones vigente, los costos que se generen por dicha contratación deberán ser cubiertos por el área solicitante.  
[ISO 9001/4.3] , [ISO 9001/4.6.2] Apéndice Norma ISO
- 3.2.6.** Antes de la aprobación de cualquier contrato, la dirección de informática deberá asegurarse que: a) Los requisitos definidos en el contrato se expresen siempre de manera adecuada e invariablemente en forma textual; b) En relación con las características del sistema, se hayan resuelto todas aquellas diferencias de opinión entre las áreas usuarias y de informática y el desarrollador, constando su firma de conformidad en el

contrato; y c) El proveedor sea capaz de cumplir los requisitos del contrato, tomando como referencia su *curriculum vitae* presentado.

[ISO 9001/4.3] Apéndice Norma ISO

- 3.2.7. Todos los sistemas y sus componentes desarrollados por personal de la institución o empresa son propiedad de éste, por lo que la institución o empresa tendrá los derechos de autor para la utilización de dichos desarrollos en las diferentes áreas que así lo requieran. En el supuesto de que el *comité editorial* apruebe su publicación, cada área de informática deberá informar la terminación del sistema al área de la institución o empresa que tramita los derechos de autor, con objeto de que se lleven a cabo los trámites requeridos. Así también, él o los autores deberán firmar la cesión de derechos a la institución o empresa, en el entendido de que la institución o empresa otorgará los créditos respectivos en la publicación.

[ISO 9001/4.1.2 ] Apéndice Norma ISO

- 3.2.8. Durante el análisis, desarrollo e implantación de cualquier sistema, el área solicitante deberá participar con su área de informática respectiva y con la dirección de informática y la empresa externa.

[ISO 9000-1/4.5] Apéndice Norma ISO

- 3.2.9. Es responsabilidad del área de informática y de la empresa externa que desarrolló un sistema, el proporcionar capacitación y asistencia técnica al personal operativo del área usuaria para el uso y mantenimiento del sistema. Será obligación del área solicitante asegurar que estos procesos cubran todas sus necesidades y requerimientos sustantivos.

[ISO 9001/4.18] Apéndice Norma ISO

- 3.2.10. El *comité de informática* a través de la dirección de informática establecerá de manera formal su política de calidad en cuanto a las normas y procedimientos por utilizar, con objeto de que funcione eficazmente el sistema de aseguramiento de calidad.  
[ISO 9001/4.1.3] Apéndice Norma ISO
- 3.2.11. El *comité de informática*, conjuntamente con la dirección de informática, será el encargado de establecer la organización interna más adecuada para las diferentes áreas de informática de la institución o empresa. Entre los puntos por considerar destacan:  
a) Establecer un organigrama, b) Delegar autoridad y c) Compartir las responsabilidades. Conforme a las necesidades de las áreas de informática de la institución o empresa, distribuir adecuadamente los recursos materiales para las áreas respectivas y establecer una política de calidad que se base en principios, con el fin de crear relaciones para que las personas trabajen en conjunto de manera efectiva. Los recursos humanos serán proporcionados por las áreas respectivas de la institución o empresa.  
[ISO 9001/4.1.2] Apéndice Norma ISO
- 3.2.12. En la elaboración y diseño de sistemas de información internos la dirección de informática será la encargada de establecer y mantener un sistema de calidad documentado para asegurar productos conforme a los requerimientos especificados por ella misma, además de alcanzar consistentemente los objetivos de calidad de la institución o empresa. Entre los documentos que se generarán por los desarrolladores, están los manuales de procedimientos, técnicos, de instalación, operativos y de usuario.  
[ISO 9001/4.5] Apéndice Norma ISO

- 3.2.13. El *comité de informática* será el encargado de establecer políticas de administración, calidad y control de calidad; así como, la justificación y consistencia de éstas. Periódicamente tiene la obligación de revisar las políticas establecidas y evaluar los resultados logrados. También verificará a través de la dirección de informática, la cooperación y comunicación entre las distintas áreas y evaluará a las empresas relacionadas (subcontratistas, distribuidores, etc.).

[ISO 9000-1/4.9.2], [ISO 9001/4.3] Apéndice Norma ISO

- 3.2.14. El *comité de informática*, a través de la dirección de informática, será el encargado de establecer y describir las clases de trabajo para el desarrollo de sistemas de información internos. Entre las clases de trabajo por considerar están: división del trabajo, identificación de fuentes de autoridad y establecimiento de relaciones. En lo que concierne a las formas de trabajo, se encuentran la funcional, por proyectos y la matricial.

[ISO 9001/4.1.2 ] Apéndice Norma ISO

#### 4. Normas para la elaboración de sistemas

##### 4.1 Objetivo

Definir la metodología a la que debe someterse todo el personal involucrado en la elaboración de sistemas, con objeto de obtener productos de alta calidad que resulten de fácil mantenimiento para cualquier miembro del equipo de trabajo.

##### 4.2. Normas generales

- 4.2.1. Los desarrollos de sistemas, tanto internos como externos, deberán respetar los lineamientos y estándares definidos en el *Manual de Procedimientos para el Desarrollo de Sistemas con Orientación a Objetos* (MPDSOO).  
[ISO 9001/4.15.1] Apéndice Norma ISO
- 4.2.2. El área de informática y la empresa externa deberán entregar al área solicitante: los programas fuentes y ejecutables, documentación técnica, manual de instalación y manual del usuario de acuerdo al MPDSOO.  
[ISO 9001/4.15.2] Apéndice Norma ISO
- 4.2.3. Para los desarrollos internos y externos, cualquier área de informática de la institución o empresa deberá entregar a la dirección de informática el original del sistema con su respectiva documentación y todos aquellos elementos que hagan posible su incorporación al *banco institucional de sistemas*, conservando una copia.  
[ISO 9001/4.15.1], [ISO 9001/4.15.6], [ISO 9001/4.16] Apéndice Norma ISO

- 4.2.4. Para aquellos sistemas que se desarrollan en un estándar para la institución, será necesario que los sistemas cuenten con un módulo de importación/exportación a través de un archivo de datos. [ISO 9001/4.15.2], [ISO 9001/4.15.5] Apéndice Norma ISO 9001
- 4.2.5. Todas las fases del desarrollo de un sistema deben estar documentadas de acuerdo al MPDS. [ISO 9001/4.5], [ISO 9001/4.16] Apéndice Norma ISO 9001
- 4.2.6. Si la institución o empresa se encuentra en un proceso de reestructuración en cuanto a su organización, en esta circunstancia se deberá iniciar la elaboración del sistema en este último caso, es conveniente la implementación de un sistema de gestión de la calidad, al menos dos meses después de que se haya implementado el nuevo esquema de organización. Cuando se contemple el uso del sistema, los límites de la reestructuración organizacional como se establezcan, se establecerán los canales de comunicación y coordinación respectiva de sus proyectos. [ISO 9001/4.4.3], [ISO 9001/4.9] Apéndice Norma ISO 9001
- 4.2.7. Por ninguna causa se deberá comenzar el desarrollo del sistema en general, sin antes tener un análisis y diseño. Para el caso en que el sistema se haya dividido en módulos, será válido el desarrollo de cada uno de ellos si se cuenta con un análisis y diseño concluidas, además de un análisis de carácter general del sistema. [ISO 9001/4.9] Apéndice Norma ISO 9001
- 4.2.8. Durante las fases de análisis y diseño de un sistema deberá tener presente el realismo de los requisitos, tanto de clases como de objetos,

semejanzas comunes, de esta forma se constituirán clases que sucesivamente afectarán los niveles de abstracción del sistema.  
[ISO 9001/4.4.1], [ISO 9001/4.9] Apéndice Norma ISO

- 4.2.9.** La metodología de orientación de objetos empleada para cualquier sistema de información invariablemente deberá proporcionar reglas para identificar, definir y organizar las clases, métodos, y mensajes que acompañen a cada objeto, así mismo debe proporcionar estrategias para organizar librerías de clases y las directrices específicas para la construcción de aplicaciones.  
[ISO 9001/4.9] Apéndice Norma ISO
- 4.2.10.** El equipo desarrollador del sistema tendrá como objetivo, en un diseño con orientación a objetos, el almacenar métodos y datos de clases en el último nivel de abstracción, esto es, lo más alto posible con el fin de aprovechar al máximo la reutilización de código.  
[ISO 9001/4.2.2], [ISO 9001/4.9] Apéndice Norma ISO
- 4.2.11.** Es conveniente que el equipo de desarrollo involucrado reserve, tanto tiempo, como personal, para familiarizarse con las librerías de clases preexistentes, con el propósito de que su uso y potencialidad sean aprovechados en la elaboración del sistema.  
[ISO 9001/4.1.2.2], [ISO 9001/4.4.2] Apéndice Norma ISO
- 4.2.12.** Durante la elección del software de desarrollo se debe contemplar que éste tenga la capacidad de soportar de manera robusta toda la aplicación, incluyendo la utilización de las clases y superclases, y en consecuencia contribuir a la reutilización del código.  
[ISO 9001/4.1.2.2] Apéndice Norma ISO

#### 4.3. Normas para el análisis de sistemas

- 4.3.1. Los desarrollos de sistemas deberán contar con un *estudio de factibilidad tecnológica y económica* que permita identificar y describir las necesidades del usuario con objeto de justificar la elaboración del sistema.  
[ISO 9001/4.4.1], [ISO 9001/4.4.2], [ISO 9001/4.4.3] Apéndice Norma ISO
- 4.3.2. Apoyados en el *estudio de factibilidad tecnológica y económica*, las áreas de informática y usuarias deberán solicitar un *dictamen técnico* a la dirección de informática.  
[ISO 9001/4.4.3], [ISO 9001/4.4.4], [ISO 9001/4.4.6] Apéndice Norma ISO
- 4.3.3. La fase de análisis de sistemas deberá apegarse a la metodología de orientación a objetos de James Rumbaugh, aprobada por el comité de informática (Apéndice A).  
[ISO 9001/4.4.1], [ISO 9001/4.4.2] Apéndice Norma ISO
- 4.3.4. Durante la fase de análisis el equipo de desarrollo del sistema obtendrá una serie de abstracciones del problema, así una abstracción es el resultado de una estructuración inteligente de la descripción del problema en elementos independientes e intuitivamente correctos.  
[ISO 9001/4.4.1], [ISO 9001/ 4.4.4] Apéndice Norma ISO
- 4.3.5. Al modelar un sistema usando metodologías con orientación a objetos se examinan e investigan los requerimientos del sistema. El sistema se interpreta con base en la perspectiva y visión de objetos y clases encontrados en el mismo, así como en las distintas alternativas de solución propuestas; su énfasis radica en la construcción de modelos apegados a la realidad.  
[ISO 9001/4.4.1], [ISO 9001/4.4.4] Apéndice Norma ISO

- 4.3.6.** En la elaboración de sistemas con orientación a objetos, uno de los principios fundamentales a observar es la elaboración de una vista lógica del problema, así como efectuar una descripción completa, específica y correcta del sistema. Enfocándose en la identificación de objetos, definición de clases y visualización de interacciones básicas entre objetos y clases.

[ISO 9001/4.4.1] Apéndice Norma ISO

- 4.3.7.** En la elaboración de sistemas con orientación a objetos es conveniente que el equipo de desarrollo visualice, en primera instancia, los atributos y propiedades relevantes de nuestro sistema, partiendo de identidades tangibles o de información relacionada a un concepto común; para la integración de objetos y clases se puede considerar por ejemplo:

Gente	Personas que desempeñan una función determinada.
Lugares	Conjunto de áreas utilizadas por personas o cosas.
Cosas	Objetos físicos o grupo de objetos que son tangibles.
Organizaciones	Colección organizada de personas, recursos, facilidades y capacidades teniendo una misión definida, su existencia es completamente independiente de los individuos.
Conceptos	Principios o ideas no tangibles utilizadas para organizar y mantener una guía de actividades.

[ISO 9001/4.4.4] Apéndice Norma ISO

- 4.3.8.** Es conveniente que el equipo de desarrollo del sistema considere las directrices siguientes para la definición de clases y métodos:
- Modelar con clases las entidades que ocurren de forma natural en el sistema.
  - Diseñar métodos de finalidad única, así como evitar métodos extensos.
  - Diseñar un nuevo método al encontrar una oportunidad de ampliar uno existente.
  - Guardar como variables de instancia los datos necesitados por más de un método, o por una subclase.
- [ISO 9001/4.9] Apéndice Norma ISO

- 4.3.9.** El equipo de desarrollo del sistema examinará, para la creación de nuevas clases, si éstas se generan a través de descripciones conceptuales de otras clases u objetos del sistema; de hecho la clasificación de objetos se inicia a partir de las descripciones efectuadas al sistema durante la fase de análisis, frecuentemente se dice que entre mejor sea la descripción mejor será la clasificación.
- [ISO 9001/4.9] Apéndice Norma ISO

- 4.3.10.** Se deberán establecer los grupos de trabajo encargados para las actividades de diseño de encuestas, entrevistas, recopilación de datos, etc.
- [ISO 9001/4.4.1], [ISO 9001/4.4.2], [ISO 9001/4.4.3] Apéndice Norma ISO

#### **4.4. Normas para el diseño de sistemas**

- 4.4.1.** La fase de diseño de sistemas deberá apegarse a la metodología de orientación a objetos de James Rumbaugh, aprobada por el comité de informática (Apéndice A).
- [ISO 9001/4.4.1], [ISO 9001/4.4.2] Apéndice Norma ISO

- 4.4.2.** De existir *manuales de procedimientos* vigentes en la institución o empresa, todos los grupos de trabajo involucrados en el diseño de sistemas deberán tener conocimiento del contenido de ellos, a fin de reflejarlos en el sistema cuando éstos lo afecten.  
[ISO 9001/4.4.1], [ISO 9001/4.5.2] Apéndice Norma ISO
- 4.4.3.** Para los casos en los cuales se efectúe un cambio en el diseño de un sistema, dicho cambio deberá ser documentado previa revisión y justificación, así como aprobación de los responsables para posterior notificación al encargado del control de la documentación, con el fin de que todas las áreas se enteren del cambio efectuado.  
[ISO 9001/4.4.9] Apéndice Norma ISO
- 4.4.4.** En la elaboración de sistemas con orientación a objetos es conveniente que el área de desarrollo comience con el estudio de bibliotecas de clases preexistentes de la institución o empresa. Para estos casos dichas bibliotecas deberán de ponerse a disposición de los programadores y diseñadores, mediante el jefe de proyecto, anexando una explicación global de las principales características de las bibliotecas. Además se deben considerar los entornos de desarrollo de sistemas anteriores, así como las bibliotecas de clases suministradas por los lenguajes de programación.  
[ISO 9001/4.4.2], [ISO 9001/4.4.3], [ISO 9001/4.4.4] Apéndice Norma ISO
- 4.4.5.** Durante la definición de clases en el modelado con orientación a objetos, el equipo de desarrollo del sistema deberá considerar todos aquellos objetos o subclases que tengan una o más propiedades en común. Frecuentemente se dice que tales propiedades o características son necesarias y suficientes para definir una clase. Se recomienda crear una nueva clase si:
- Dicha clase represente una abstracción significativa del problema.

- Sea probable que los servicios que proporciona sean utilizados por varias clases más.
  - Su conducta sea inherentemente compleja.  
[ISO 9001/4.2.2], [ISO 9001/4.4.5], [ISO 9001/4.14.3] Apéndice Norma ISO
- 4.4.6.** En las etapas previas a programación, el equipo de desarrollo no debe de perder de vista:
- La identificación de objetos.
  - La organización de jerarquías de clases.
  - La reutilización de clases.
  - La construcción de marcos estructurales (plantillas) a partir de las librerías de clases.  
[ISO 9001/4.4.2], [ISO 9001/4.4.3] Apéndice Norma ISO
- 4.4.7.** Durante las primeras etapas de elaboración del sistema, el equipo de desarrollo debe asegurar el aprovechamiento máximo del lenguaje de programación orientado a objetos a utilizar, considerando para ello las clases preexistentes (librerías de clases).  
[ISO 9001/4.4.1], [ISO 9001/4.4.4] Apéndice Norma ISO
- 4.4.8.** Debe recordarse que en el desarrollo de sistemas de información con orientación a objetos, el énfasis del diseño recae en los datos contenidos de las entidades u objetos, sin perder de vista la agrupación de métodos comunes con el fin de generalizarlos en las clases más adecuadas.  
[ISO 9001/4.4.5], [ISO 9001/4.9] Apéndice Norma ISO
- 4.4.9.** Para los casos en los cuales el proyecto se divida en módulos, un módulo no podrá considerarse en la etapa de diseño respectiva como *terminada* si antes no se ha definido expresamente cada clase, método y mensaje que formen parte de este módulo.  
[ISO 9001/4.4.5], [ISO 9001/4.4.6] Apéndice Norma ISO

- 4.4.10.** En los casos en que algún miembro del equipo de desarrollo sugiera incluir bibliotecas de clases específicas para un módulo, y por razones de seguridad (encriptamientos) desee permitir únicamente a algún miembro del equipo de desarrollo el detalle del código, procedimiento u objeto, deberá presentar un escrito ante el comité de informática, en el cual se argumente la razones por las cuales sugiere mantener dicha reserva.  
[ISO 9001/4.4.1], [ISO 9001/4.4.3] Apéndice Norma ISO
- 4.4.11.** En el refinamiento de niveles de abstracción que sean de utilidad para el sistema, debe mantener un nivel de sencillez que logre una fácil comprensión de la especialización, tanto a nivel de clases, como a nivel de subclases, de forma tal que se logre una profundidad más que un ancho en los arboles de herencia  
[ISO 9001/4.4.1], [ISO 9001/4.9] Apéndice Norma ISO
- 4.4.12.** En los arboles de herencia, la generalización de las jerarquías de clases debe reflejar un nivel de calidad aceptable en cuanto a su manejo, para esto se contemplan los siguientes tres puntos:
- A. Manejo de protocolos estándar.** Comprende tanto los nombres como el manejo de mensajes y métodos de forma común con el fin de formar una superclase, siempre y cuando el conjunto de clases comparta dicho protocolo.
  - B. Construcción de clases abstractas.** Define la tendencia de construcción de bibliotecas de clases con profundidad y con forma estrecha.
  - C. Identificación de marcos estructurales** Permiten a los programadores representar un nivel más alto de abstracción. Se identifican examinando grandes clases y reconsiderando su organización, dicha organización establecerá un conjunto

de jerarquías de clases y cada una de ellas permitirá construir categorías concretas de aplicación.

[ISO 9001/4.4.1], [ISO 9001/4.9] Apéndice Norma ISO

**4.4.13.** Para los casos en que se detecte algún error en algún método perteneciente a alguna clase que contenga una profundidad (especialización) trascendente para el resto de los módulos y proyectos, se debe notificar de manera inmediata al jefe de proyecto con el fin de que tome las medidas pertinentes.  
[ISO 9001/4.4.1], [ISO 9001/4.4.5], [ISO 9001/4.4.6] Apéndice Norma ISO

**4.4.14.** En el caso de la construcción de superclases con respecto a la herencia, se debe procurar una elaboración justificada y a la vez no especializada.  
[ISO 9001/4.9] Apéndice Norma ISO

**4.4.15.** Para que un conjunto de clases pueda ser considerado como parte de la biblioteca de objetos, éste deberá ser completo en cuanto a código útil, seguimiento documentado y con garantía de reusabilidad.  
[ISO 9001/4.4.1], [ISO 9001/4.4.6] Apéndice Norma ISO

#### **4.5. Normas para la programación y documentación de sistemas**

**4.5.1.** Todos los programas que integren cualquier sistema deberán estar documentados conforme al *Manual de Procedimientos para el Desarrollo de Sistemas* con Orientación a Objetos (MPDSOO), punto 5.  
[ISO 9001/4.5], [ISO 9001/4.8], [ISO 9001/4.16] Apéndice Norma ISO

**4.5.2.** El área usuaria deberá aprobar el *manual del usuario* previo a la liberación de un sistema. La dirección de informática deberá revisar que el *manual técnico* se apegue a las especificaciones del MPDSOO; en los casos que así se considere necesario la dirección de informática evaluará dichos manuales.  
[ISO 9001/4.9], [ISO 9001/4.10.1] Apéndice Norma ISO

**4.5.3.** El encargado del control de la documentación de un sistema, será subordinado directo del líder de proyecto en que esté involucrado el sistema. Este encargado dictaminará el control de la documentación con base en claves de control (en la esquina inferior izquierda), las cuales serán de conocimiento general para el equipo de desarrollo del sistema. Los documentos serán inventariados en una lista maestra de control de documentos, en la cual se tenga constancia del estado actual de cada documento y de quién tiene posesión del mismo. Ejemplos:

(  
CO=Confidencial  
NF=No fotocopiable  
CE \_\_\_\_ = Control específico del \_\_\_\_  
UG= Uso general  
CF \_\_/\_\_/\_\_ = Cambios frecuentes de fecha de publicación.  
etc. )

[ISO 9001/4.5], [ISO 9001/4.5.2], [ISO 9001/4.5.3] Apéndice Norma ISO

**4.5.4.** El encargado del control de la documentación tendrá especial cuidado en la documentación que presente cambios frecuentes, ya que será su obligación el velar que en todas las áreas se cambie la documentación obsoleta del sistema por documentación actualizada.  
[ISO 9001/4.5] Apéndice Norma ISO

- 4.5.5. Todos aquellos códigos que sean objeto de programación, ya sean módulos, programas, pantallas, etc., deberán contener información de quién efectuó la programación y en qué fecha; de ser posible en el mismo software, mediante comentarios y adicionalmente en la documentación por escrito [Apéndice B].  
[ISO 9001/4.8], [ISO 9001/4.16] Apéndice Norma ISO
- 4.5.6. Después de concluida la programación de una parte del sistema, se deberá registrar en un documento que dicha parte del sistema ha sido concluida, especificar el o los nombre(s) del o los programador(es), así como el tiempo de programación en horas; esto con el fin de establecer un control de calidad del trabajo de los programadores.  
[ISO 9001/4.8], [ISO 9001/4.9] Apéndice Norma ISO
- 4.6. Normas para la realización de pruebas
- 4.6.1. Antes de liberar un nuevo sistema, éste deberá ser sometido a pruebas de aceptación definidas por el área usuaria, utilizando para ello datos reales. En el caso de nuevas versiones, será necesario realizar corridas en paralelo para verificar su correcto funcionamiento con respecto a la versión anterior.  
[ISO 9001/4.4.7], [ISO 9001/4.11] Apéndice Norma ISO
- 4.6.2. Las pruebas de aceptación deberán ser clasificadas en: *preliminares*, para los casos en que se pruebe el módulo o el programa de manera individual, y *totales*, cuando se encuentren ensamblados todos los componentes del sistema. Para cada una de estas pruebas, se llevará un control de los resultados obtenidos.  
[ISO 9001/4.11] Apéndice Norma ISO
- 4.6.3. Las corridas de prueba que se realicen con el fin de acreditar un sistema como aceptado, deberán efectuarse con una cantidad de

datos superior al 50% de la cantidad de datos que el sistema correrá de manera cotidiana, y con el equipo de cómputo en el que se pretende operar sistemáticamente. Para el caso de sistemas que operen en red, también se deberán efectuar pruebas con usuarios concurrentes.

[ISO 9001/4.4.8], [ISO 9001/4.14.3] Apéndice Norma ISO

- 4.6.4.** Los tipos de datos con los cuales se efectúen las pruebas deberán estar apegados a la realidad, a fin de tomar en cuenta el rango de valores que soportará el sistema y posteriormente realizar una gráfica de rendimiento de cantidad de datos contra el tiempo de procesamiento. En el caso de sistemas para trabajo en red, deberán establecerse elementos que permitan observar objetivamente el desempeño del sistema. Si los resultados de rendimiento del sistema no son aceptables para fines prácticos, se consignará el módulo para su re-trabajo en programación.

[ISO 9001/4.13], [ISO 9001/4.14.2], [ISO 9001/4.14.3] Apéndice Norma ISO

#### **4.7. Normas para la implantación de sistemas y capacitación**

- 4.7.1.** La capacitación al personal técnico-operativo formará parte fundamental de la liberación de un sistema. Dicha capacitación deberá cubrir todas las necesidades y requerimientos que el área usuaria especifique de común acuerdo con el área de informática o empresa externa.

[ISO 9001/4.18] Apéndice Norma ISO

- 4.7.2.** El proceso de capacitación deberá ser posterior a la aprobación de los manuales: a) técnico, b) de instalación, c) de operación y d) del usuario, que constituirán la guía con la que se lleve a cabo dicho proceso.

[ISO 9001/4.5], [ISO 9001/4.9] Apéndice Norma ISO

- 4.7.3. Los manuales de operación deberán especificar los métodos de manejo que permitan cuidar la integridad, tanto física como lógica de los elementos que conforman el sistema, ya sean datos, información, software, hardware y documentación.  
[ISO 9001/4.15.1], [ISO 9001/4.15.2] Apéndice Norma ISO

#### 4.8. Normas para el mantenimiento de sistemas

- 4.8.1. El área usuaria deberá solicitar el mantenimiento de un sistema a su área de informática o empresa externa, siempre y cuando se identifiquen y justifiquen plenamente los ajustes y cambios necesarios que permitan mejorar el desempeño y cobertura del sistema en cuestión. Para los casos que así se considere necesario, deberá solicitarse un *dictamen técnico* a la dirección de informática.  
[ISO 9001/4.14.2] Apéndice Norma ISO.
- 4.8.2. Aquellos códigos del sistema que no trabajen de manera óptima con respecto a las necesidades o rendimiento que se pretenda satisfacer, serán dispuestos a un proceso de re-trabajo; en primera instancia a quien realizó la programación, y en último caso a un nuevo equipo de trabajo para programación, esto considerando un estilo de programación diferente que sea más adecuado a la necesidad a satisfacer. La situación anteriormente descrita debe registrarse en la documentación correspondiente.  
[ISO 9001/4.13], [ISO 9001/4.14.2] Apéndice Norma ISO.

## **5. Manual de procedimientos para el desarrollo de sistemas con orientación a objetos (MPDSOO)**

### **5.1. Introducción**

El presente *Manual de Procedimientos para el Desarrollo de Sistemas con Orientación a Objetos* (MPDSOO) está dirigido a las personas directamente relacionadas con el desarrollo de sistemas; como tal, representa una guía para orientar y normar los trabajos y actividades involucradas en el análisis, diseño y desarrollo de los sistemas.

Existen varias metodologías y tecnologías que apoyan el desarrollo de sistemas, la mayoría de éstas identificadas con los distintos métodos de análisis y diseño. Este manual de procedimientos está organizado siguiendo los lineamientos de dichos métodos; establece una metodología constituida por una serie de actividades orientadas a regular las acciones de los analistas.

Los procedimientos para el desarrollo de sistemas que se presentan en este manual se encuentran apoyados en las políticas y normas anteriormente descritas.

### **5.2. Objetivo del MPDSOO**

Describir los procedimientos para cada una de las etapas que comprenden el desarrollo de sistemas, observando el cumplimiento de los elementos que constituyen un plan de calidad para el proceso de elaboración de sistemas con orientación a objetos.

### **5.3. Plan de calidad para el desarrollo de sistemas**

Todo plan de calidad para el desarrollo de sistemas debe contemplar y describir esencialmente los puntos siguientes:

- A. Establecer los objetivos, propósitos y alcances del plan (qué, cuándo, dónde, quién, para quién).  
[ISO 9001/1], [ISO 9001/4.1.1] Apéndice Norma ISO
- B. Pormenorizar otros documentos relacionados con el plan.  
[ISO 9001/4.3], [ISO 9001/4.16] Apéndice Norma ISO
- C. Describir la estructura organizacional del equipo de desarrollo, tareas que realizarán y las responsabilidades específicas con la calidad del sistema.  
[ISO 9001/4.1.2], [ISO 9001/4.4.3] Apéndice Norma ISO
- D. Definir los criterios de aceptación para las entradas y salidas del sistema, así como para cada fase.  
[ISO 9001/4.4.4], [ISO 9001/4.4.5] Apéndice Norma ISO
- E. Puntualizar las pruebas a efectuar al sistema y su realización.  
[ISO 9001/4.10.1], [ISO 9001/4.11] Apéndice Norma ISO
- F. Definir las herramientas y técnicas específicas que se usarán para apoyar las actividades del control de calidad.  
[ISO 9001/4.1.2.2], [ISO 9001/4.4.1] Apéndice Norma ISO
- G. Explicar los métodos y facilidades que servirán para proteger los medios físicos del programa de computadora.  
[ISO 9001/4.15.6] Apéndice Norma ISO
- H. Detallar los estándares, prácticas y convenciones que se utilizarán en el sistema.  
[ISO 9001/4.4.4], [ISO 9001/4.9] Apéndice Norma ISO
- I. Puntualizar los métodos y facilidades para el mantenimiento del sistema.  
[ISO 9001/4.8], [ISO 9001/4.15.5], [ISO 9001/4.16] Apéndice Norma ISO

#### 5.4. Etapas para el desarrollo de sistemas

Con el fin de contar con un marco conceptual uniforme, se considera el ciclo de vida de un sistema constituido por seis etapas:

- A) Análisis
- B) Diseño
- C) Programación documentada
- D) Pruebas
- E) Implantación y capacitación
- F) Mantenimiento

En las siguientes secciones se define el objetivo de cada etapa, se describe en qué consiste y se listan las principales actividades involucradas.

Por último, en la *tabla de formas o formatos* para la documentación de sistemas se resumen los productos que deben obtenerse en cada una de las etapas de elaboración, se identifican los responsables de cada una de ellas: *área usuaria (U)*, *área de informática (I)* y, en caso de existir, *empresa responsable del desarrollo (E)*. Es de gran importancia integrar estos productos en un expediente que permita conocer todas las etapas del ciclo de vida de un sistema, asegurando así su continuidad a través de la independencia del grupo de trabajo que lo desarrolló.

[ISO 9001/4.4.1], [ISO 9001/4.4.2] Apéndice Norma ISO

#### 5.5. Modelado con orientación a objetos.

Una manera clara de tener idea de cómo se conforma un sistema o una problemática es a través del modelado del mismo. Entre más aproximado se encuentre este modelo al sistema real, nuestras conclusiones y acciones serán más efectivas y congruentes para solucionar la problemática.

En primer lugar se contempla un *modelado estático*, el cual se dedica de manera individual a los aspectos de objetos, así como de sus datos y la estructura de los mismos. Por otra parte se efectúa un *modelado dinámico* que se dedica a representar los aspectos de control de un sistema, su principal objetivo es el modelar el control que se tendrá en el sistema en cuanto a aspectos temporales. Por último se presenta un *modelado funcional* cuyo propósito es abstraer los aspectos de función en cuanto a las transformaciones que se impliquen en el tratamiento de la información.

Cada modelo puede comprenderse por sí mismo, sin que esto implique que son independientes. Cada uno de ellos presentará referencias y consideraciones para con los otros, procurando lo más posible el no duplicar las características modeladas; lo cual implica que dichas interconexiones sean limitadas y explícitas. Cada modelo evolucionará en el ciclo de vida del sistema, ya que el análisis contempla al sistema como solución en un dominio específico, el diseño añade consideraciones al sistema para la solución, y la programación considerar los aspectos de solución y de aplicación.

[ISO 9001/4.4.1] Apéndice Norma ISO

## **6. Análisis de sistemas**

### **6.1. Objetivo**

Llevar a cabo los procesos de comprensión y clasificación de hechos, diagnóstico de problemas y estructuración de la información para recomendar soluciones a la problemática en estudio.

### **6.2. Descripción**

La fase de análisis contempla un conjunto lógico y bien definido de actividades enfocadas a determinar lo que el sistema debe realizar, sin importar la forma en que lleve a cabo su función, es decir, se abordan los aspectos lógicos y no los físicos.

El propósito del análisis orientado a objetos es la obtención de un modelo completo, preciso, conciso, comprensible y correcto del sistema. Para ello es necesario examinar los requisitos, analizar sus implicaciones y volver a plantearlos rigurosamente. Esta fase indica lo que debe hacerse, sin limitar la forma en que se llevará a cabo, se especifican claramente los requisitos del sistema a elaborar, es una base para el acuerdo entre el solicitante del software y el desarrollador del sistema, y se concibe el marco de trabajo para el posterior diseño y desarrollo. En la definición del sistema se indaga un conjunto de requerimientos tales como la definición del ámbito del problema, conocer el contexto de la aplicación, realizar suposiciones e investigar necesidades de rendimiento.

Durante esta fase se obtendrá una descripción de qué va hacer el sistema, y no cómo lo va a realizar; por lo que se busca obtener un modelo del mundo real a través de la identificación de objetos, clases, atributos, responsabilidades y colaboraciones. Es frecuente que la mayoría de éstos se aparten del vocabulario descriptivo del sistema, en este sentido el análisis es la fase donde los desarrolladores en conjunto con el usuario dialogan y obtienen este vocabulario.

[ISO 9001/4.4.2], [ISO 9001/4.4.3] Apéndice Norma ISO

### 6.3. Actividades del análisis

#### 6.3.1. Estudio de factibilidad

Cuando se recibe una petición formal para solucionar una problemática a través de un sistema, se inicia una investigación preliminar en la cual se lleva a cabo un "estudio de factibilidad". Este estudio permite evaluar los aspectos técnicos, económicos y operacionales que enmarcan a la problemática en estudio, dando como resultado alternativas de solución, tales como la creación de un sistema o la automatización de procedimientos a través de paquetes de uso general.

ISO 9001/4.1.2.2] Apéndice Norma ISO

##### *Factibilidad técnica*

El analista de sistemas evalúa los méritos técnicos del concepto del sistema, mientras que al mismo tiempo conjunta información adicional sobre el rendimiento, fiabilidad, facilidad de mantenimiento y posibilidad de producción. Es esencial que el proceso de análisis y de definición se realice en paralelo con el análisis técnico. De esta forma se pueden juzgar las especificaciones concretas según se van determinando. Entre las preguntas a plantearse estarían: ¿qué tecnologías se requieren para conseguir la funcionalidad y el rendimiento del sistema?, ¿qué métodos, algoritmos o procesos se requieren y cuál es el riesgo de su desarrollo?, ¿cómo afectarán al costo estos elementos de tecnología?, etc.

[ISO 9001/4.1.2.2]. [ISO 9001/4.4.4.] Apéndice Norma ISO

##### *Factibilidad económica*

Este estudio se lleva a cabo en segundo lugar. Los recursos básicos que deben considerarse son: el tiempo de personal de análisis del sistema, el costo de la realización integral de un estudio de sistemas, el costo del tiempo del empleado para la

institución, el costo estimado del equipo y el costo estimado del software que se comprará o desarrollará.

[ISO 9001/4.3], [ISO 9001/4.4.3] Apéndice Norma ISO

***Factibilidad operativa***

Esta depende de los recursos humanos que participan durante la operación del sistema. Esto se refiere a pronosticar de alguna manera lo que puede acontecer cuando el sistema esté ya instalado, de que éste llegara a funcionar o a usarse. Si los usuarios del sistema existente no tienen queja aparente del sistema, están muy acostumbrados a usar ese sistema, y no acuden a los directivos para tratar de hacerle mejoras, es probable que se resistan bastante a un cambio, por lo que puede originarse el problema de que el sistema no llegue a ser operativo. [ISO 9001/4.1.2.2] Apéndice Norma ISO

**6.3.2. Establecimiento del *plan para el desarrollo de un sistema***

La disposición correcta de las partes esenciales de un plan proporcionarán el medio para orientar las acciones a considerar en la elaboración de cualquier sistema.

Una acción puede realizarse a través de un plan previamente elaborado o sin él, la diferencia está en los resultados obtenidos, los cuales habrán costado más o menos esfuerzo, tiempo, dinero, recursos humanos, etc., según se haya procedido siguiendo un plan o sin seguirlo.

**Plan de desarrollo de sistemas**

Las partes esenciales de un plan para el desarrollo de un sistema son:

- A. Título del plan.
- B. Definición del sistema, incluyendo su propósito y alcances del mismo.  
[ISO 9001/4.1.1] [ISO 9001/1] Apéndice Norma ISO
- C. Establecimiento y definición de objetivos haciendo referencia tanto a los del área de desarrollo como a los del cliente.  
[ISO 9001/4.1.1] Apéndice Norma ISO
- D. Relación de los principales obstáculos o problemas que debe superar el plan para alcanza el resultado.  
[ISO 9001/4.1.2] Apéndice Norma ISO
- E. Descripción de la organización del sistema, orientandose a la estructura del equipo de desarrollo del sistema, a los calendarios comprendidos, responsabilidades, asignación y división del trabajo que concierne a los módulos del sistema.  
[ISO 9001/4.1.2], [ISO 9001/4.4.2] Apéndice Norma ISO
- F. Especificación de las actividades a seguir para rastrear los requerimientos del cliente, así como la descripción de los recursos utilizados tanto materiales como humanos en caso de que existan. Se muestran también los métodos y herramientas que se pretenden utilizar en el desarrollo del sistema.  
[ISO 9001/4.1.2.2] Apéndice Norma ISO
- G. Describir el desarrollo del sistema a través de fases , esto es, la transformación de la especificación de los requerimientos del cliente, la identificación de las entradas del sistema, las salidas obtenidas, y verificación de las salidas obtenidas.  
[ISO 9001/4.4.1], [ISO 9001/4.4.4], [ISO 9001/4.4.5] Apéndice Norma ISO

Para que tenga sentido hablar de control sobre la acción o de coordinación entre varias acciones simultáneas es necesario darle su valor real al trabajo de planeación, o sea, tener planes completos y no sólo una parte.

[ISO 9001/4.4.1], [ISO 9001/4.4.2], [ISO 9001/4.4.3] Apéndice Norma ISO

#### **6.3.3. Calendario de trabajo**

Una vez aprobado el plan para el desarrollo del sistema, se estructura un calendario de trabajo con el propósito de identificar los tiempos requeridos para cada una de las etapas del desarrollo, así como a los responsables de éstas.

[ISO 9001/4.4.1], [ISO 9001/4.4.3] Apéndice Norma ISO

#### **6.3.4. Relación de documentos**

Todos los documentos recabados durante el análisis, tales como formatos utilizados por el área usuaria, manuales de organización y procedimientos, deberán integrarse en el expediente del sistema y relacionarse para facilitar su referencia y localización. Para tal efecto, se recomienda llevar a cabo técnicas de levantamiento de información (encuestas previas, entrevistas, etc.), para tener una visión amplia del manejo de datos que se alimentarán y serán procesados por el sistema para obtener los resultados deseados.

[ISO 9001/4.4.3], [ISO 9001/4.5] Apéndice Norma ISO

#### **6.3.5. Modelado en orientación a objetos**

- **Modelado estático**

Se analizan los requisitos del sistema, con base en la construcción de un modelo de objetos. Éste muestra la estructura estática de datos correspondientes al sistema, la

**organiza en segmentos manejables describiendo clases de objetos del mundo real y las interrelaciones entre sí. Durante el modelado de objetos se realizan:**

[ISO 9001/4.4.4], [ISO 9001/4.9] Apéndice Norma ISO

*Identificación de clases y objetos*

Se identifican las clases de objetos concernientes al dominio de la aplicación. Se consideran las entidades físicas así como los conceptos, se enumeran los candidatos a clases y objetos que se encuentran en la descripción escrita del sistema. En un principio es importante no ser demasiado selectivo al escoger las clases y objetos, es decir, se escriben todas las clases y objetos que se identifiquen. Posteriormente, se eliminan clases y objetos inadecuados, para ello es importante recordar si una clase tiene poco o nada que ver con el sistema.

[ISO 9001/4.2.3], [ISO 9001/4.4.4] Apéndice Norma ISO

*Diccionario de clases*

Se construye un diccionario de clases cuyo propósito es documentar y detallar cada clase en cuanto a sus atributos, métodos, alcances y las entidades con las cuales se asocia. Este diccionario complementa algunas características, suposiciones, restricciones y comentarios acerca de sus miembros o de su utilización.

[ISO 9001/4.4.3], [ISO 9001/4.8], [ISO 9001/4.9] Apéndice Norma ISO

*Identificación de asociaciones entre objetos*

Se identifican las asociaciones entre clases, una asociación es toda dependencia entre dos o más clases, así como una referencia de una clase a otra. Éstas muestran dependencias entre clases con el mismo nivel de abstracción que las clases en sí. Para la identificación de asociaciones suele corresponderse con verbos de estado o con locuciones

verbales. Nuevamente se extraen todos los candidatos y posteriormente se seleccionan las más adecuadas mediante un refinamiento.

[ISO 9001/4.4.1], [ISO 9001/4.4.7] Apéndice Norma ISO

*Identificación de atributos*

Se identifican los atributos de los objetos, éstos son propiedades o características de objetos individuales. A diferencia de las clases y de las asociaciones, es menos probable que los atributos se describan por completo en la definición del problema. Los analistas recurren al conocimiento del dominio del sistema y del mundo real para encontrarlos.

[ISO 9001/4.4.4] Apéndice Norma ISO

*Refinamiento mediante herencia*

Se organizan las clases utilizando la herencia a fin de compartir una estructura común. La herencia puede ser de dos tipos ascendente y descendente. En la herencia ascendente se generalizan aspectos comunes de clases, como por ejemplo atributos, asociaciones u operaciones, generando una superclase y en cuanto a la herencia descendente se refinan las clases existentes para dar origen a una subclase especializada.

[ISO 9001/4.4.7], [ISO 9001/4.9] Apéndice Norma ISO

- **Modelado dinámico**

Éste muestra la forma en que el comportamiento del sistema y de objetos que lo integran va variando con el tiempo. Se inicia indagando los sucesos que son estímulos y respuestas visibles del sistema, se continúa con el establecimiento de los escenarios típicos y se extraen los sucesos de los escenarios respectivos. Se recomienda identificar primero los sucesos y asignar entonces cada suceso a su objeto destino. El fin del

modelado dinámico es capturar la lógica de control de la aplicación, es decir, se enfoca en el flujo y control de información. Durante el modelado dinámico se realiza: [ISO 9001/4.4.1], [ISO 9001/4.9] Apéndice Norma ISO

#### *Establecimiento de escenarios*

Los escenarios muestran las interacciones principales del sistema, los formatos de visualización externa y los intercambios de información. Un escenario está conformado por una secuencia de sucesos, los sucesos se producen siempre que se intercambia información entre un objeto del sistema y un agente externo; como por ejemplo un usuario, un sensor o alguna otra tarea. Durante cada escenario se identifica qué objetos participan en él, las responsabilidades de cada objeto y cómo esos objetos colaboran con otros objetos en términos de sus operaciones. En primer lugar se describen escenarios para casos normales, esto es, interacciones sin ninguna entrada extraña y sin situaciones de error. Después se consideran los casos especiales tales como valores máximos y mínimos, y valores repetidos.

[ISO 9001/4.4.5], [ISO 9001/4.9] Apéndice Norma ISO

#### *Identificación de sucesos*

Se examinan los escenarios para identificar todos los sucesos que le ocurran al sistema. Entre éstos se cuentan todas las señales, entradas, decisiones, interrupciones, transacciones y acciones procedentes o destinadas al usuario o a otros dispositivos. También hay que asignar cada tipo de suceso a las clases de objetos que lo envían y lo reciben. El suceso es de salida para el remitente y de entrada para el destinatario. Es importante recordar que todo escenario debe mostrarse como una traza de sucesos, esto es, una lista ordenada de entre distintos objetos asignados a las columnas de una tabla.

[ISO 9001/4.4.1], [ISO 9001/4.4.4] Apéndice Norma ISO

### **Diagrama de estados**

Este diagrama debe mostrar los sucesos enviados y recibidos por el objeto u objetos que contenga el sistema. Así, todo escenario o seguimiento de sucesos ayudará a la construcción del diagrama de estados.

Éste comienza por los diagramas de seguimiento de sucesos que afectan a una clase que se esté modelando, en primera instancia se consideran solamente los sucesos que afecten a un único objeto y posteriormente a los demás. Su disposición se muestra en los arcos regulados por los sucesos de entrada y de salida que se encuentran a lo largo de una columna de seguimiento (diagrama de seguimiento de sucesos). El intervalo entre dos sucesos cualesquiera es lo que forma un estado. Existe la posibilidad de que aparezcan bucles dentro del diagrama, éstos se identifican cuando se repite indefinidamente una secuencia de sucesos, si en un bucle su primer estado y el último son idénticos, es importante que por lo menos uno de sus estados tenga múltiples transacciones que salgan de él, de otra forma el bucle no terminará nunca.

[ISO 9001/4.2.1], [ISO 9001/4.8] Apéndice Norma ISO

- **Modelo funcional**

En este se muestra la forma en que se procesan los valores del sistema, no se toman en cuenta las secuencias, ni la estructura de los objetos. El modelo muestra qué valores dependen de qué otros valores, y las funciones que los relacionan. Hace uso del diagrama de flujo de datos para mostrar dependencias funcionales, las funciones pueden representarse por medio del lenguaje natural o de expresiones matemáticas. La relación que mantiene el diagrama de flujo de datos con la orientación a objetos, es la correspondencia de los procesos con las actividades o acciones de los diagramas de estados de las clases, mientras que los flujos se relacionan con

los objetos o con valores de los atributos de un diagrama de objetos. Otros puntos a considerar son:

[ISO 9001/4.2.1], [ISO 9001/4.9] Apéndice Norma ISO

*Identificación de valores de entrada y de salida*

Se examina la definición del sistema para buscar y enumerar todos los valores de entrada y de salida, estos representarán los parámetros de los sucesos que se intercambian entre el sistema y el exterior.

[ISO 9001/4.4.4] Apéndice Norma ISO

*Diagrama de flujo de datos*

El análisis permite especificar lo que se requiere que haga el sistema o la aplicación, identificando las funciones de los principales componentes que lo constituyen. Para ello, una vez aprobada la factibilidad del sistema, se elaborará un modelo informativo del área en la que funcionará el sistema, mediante un *diagrama de flujo de datos* (DFD); éste consiste en representar gráficamente los elementos básicos de los procesos, la información que fluye entre ellos, el sitio donde se almacenan los datos y las fuentes, y destinos de éstos. También permitirán mostrar cada valor de salida a partir de los de entrada; pueden construirse en capas, la superior puede constar de un solo proceso, o quizá de uno para obtener entradas, otro para calcular valores y un tercero para generar las salidas.

[ISO 9001/4.4.4] Apéndice Norma ISO

*Descripción de funciones*

Una vez refinado el diagrama de flujo de datos se escribe una descripción de cada función, tal descripción puede estar en lenguaje natural, expresiones matemáticas o algún otro diagrama adecuado. Se debe centrar en lo que hace y no el cómo va implantarse. Existen dos tipos de descripciones, la

declarativa que especifica las relaciones entre los valores de entrada y los de salida, y las relaciones entre valores de salida; la procedimental especifica la función dando un algoritmo para calcularla, su propósito es establecer lo que hace la función. [ISO 9001/4.4.1], [ISO 9001/4.4.4] Apéndice Norma ISO

*Diccionario de términos*

Todas las definiciones de los elementos en el sistema —flujo de datos, procesos y almacenes de datos— deberán describirse en forma detallada en un *diccionario de términos*.

El diccionario de términos es una referencia de los *datos acerca de los datos*, recopilados por el equipo que desarrollará un sistema, datos que le servirán como guía para ayudarse en el análisis y diseño. Por lo general, las referencias al documento se hacen empleando el nombre del objeto de interés; de ahí que se le dé el nombre de diccionario de términos. También contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre y sinónimo del dato (alias), descripción, los datos elementales que se relacionan con el término, el rango permitido del dato, la longitud disponible en caracteres, contenidos y organización.

[ISO 9001/4.4.3], [ISO 9001/4.8] Apéndice Norma ISO

6.3.6. Acta de acuerdos

Será necesario establecer compromisos concretos y por escrito entre el personal de sistemas y los usuarios, estipulando claramente los requerimientos completos del usuario; estos compromisos garantizarán que el sistema en cuestión cubrirá totalmente los requerimientos expuestos por los usuarios. Para ello se utilizarán *actas de acuerdos* donde se registren claramente, para cada etapa del desarrollo del sistema, cada uno de los acuerdos y compromisos adquiridos por ambas partes.

[ISO 9001/4.3], [ISO 9001/4.4.3] Apéndice Norma ISO

## **7. Diseño de sistemas**

### **7.1. Objetivo**

**Definir las especificaciones necesarias para desarrollar e implantar el sistema de acuerdo con los resultados del análisis.**

### **7.2. Descripción**

A través de la definición de las características funcionales del sistema, se logrará una adecuada selección de las herramientas de software, así como el hardware que se utilizarán para su construcción.

Para poder llevar a cabo el desarrollo de un sistema, es necesario diseñar las estructuras de las bases de datos y tablas o archivos necesarios para la integración de la información que utilizará, así como los procedimientos de entradas, salidas y procesos que permitirán satisfacer los requerimientos identificados en la etapa de análisis, con ello se logrará la optimización en el uso de los recursos de cómputo involucrados y se garantizará la facilidad de operación, la integridad de la información y la confiabilidad del sistema.

Durante esta fase se implanta una estrategia de alto nivel, se crea una arquitectura para poder desarrollar la implantación del sistema, es decir, se toman decisiones acerca de la forma en que se elaborará el sistema. Se construye la solución, primeramente viendo al sistema desde un nivel global y después empleando otros niveles de abstracción. Frecuentemente se incluye decisiones acerca de la organización en subsistemas, decisiones conceptuales y de políticas comunes para ser utilizadas por los elementos del sistema que posteriormente serán las que constituyan un marco de trabajo para el propio diseño.

La arquitectura de un sistema orientado a objetos engloba su estructura de objetos y clases, es importante mostrar el conjunto de clases dentro de

categorías de clases (a nivel lógico), y al conjunto de módulos dentro de subsistemas (a nivel físico). La obtención de una arquitectura denota una captura significativamente importante de todas las categorías y subsistemas. Los objetos y clases descubiertos en la fase anterior permiten al equipo de desarrollo escoger distintas formas de implantar la solución del problema, las operaciones identificadas se expresarán en forma de algoritmos, descomponiendo las operaciones complejas en operaciones internas sencillas. Las clases, atributos y asociaciones deben constituirse en forma de estructura de datos específicas.

El diseño con metodologías de orientación a objetos se vuelve un proceso iterativo, comúnmente al estar terminando un nivel de abstracción, se debe reflexionar para añadir detalles y completar el diseño y así llevarlo a algún nivel más fino de detalles. Quizá sea preciso agregar nuevas operaciones y atributos a las clases existentes, inclusive es posible que haya que identificar las nuevas clases, se revisan las relaciones entre objetos y posiblemente se realicen cambios en la jerarquía de clases.

[ISO 9001/4.1.2.2], [ISO 9001/4.4.1], [ISO 9001/4.4.2], [ISO 9001/4.4.3], [ISO 9001/4.4.4], [ISO 9001/4.7] Apéndice Norma ISO

### 7.3 Actividades del diseño

#### 7.3.1. Selección e identificación de las herramientas de desarrollo y del hardware

Lo deseable es que primero se identifiquen las aplicaciones por resolver, posteriormente se seleccione el software que las pueda soportar y por último seleccionar el hardware que soporte a dicho software. Por lo general, el desarrollo de sistemas para una institución o empresa partirá de su infraestructura de equipos de cómputo ya establecida (se espera que sea de la última tecnología), por lo que, en la definición de las características

funcionales del sistema se determinará la forma de aprovechar al máximo dicha infraestructura.

En paralelo, se identificarán las herramientas de software adecuadas para construir el sistema, apegándose a los estándares establecidos por la dirección de informática.

[ISO 9001/4.1.2.2], [ISO 9001/4.4.1], [ISO 9001/4.4.4] Apéndice Norma ISO

### 7.3.2. Diagrama entidad-relación

El propósito de este diagrama es dar una descripción de las relaciones entre entidades (personas, lugares, eventos y objetos) de un sistema y el conjunto de información que utilizan dichas entidades, con lo cual puede conceptualizarse la arquitectura o modelo de datos y el proceso de programación.

[ISO 9001/4.5], [ISO 9001/4.9] Apéndice Norma ISO

### 7.3.3. Diseño de las bases de datos y archivos

Una vez descritas las relaciones entre las entidades del sistema, se procederá a la estructuración de las bases de datos, catálogos y tablas o archivos que conformarán el sistema. Dentro de esta estructuración se generará el *diccionario de datos*, el cual contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo: nombre, descripción, alias, contenidos, organización y su probable interacción con otras bases de datos o archivos de la institución.

Dada la importancia que tienen las bases de datos como núcleos de información sustantiva para el apoyo a la toma de decisiones, resulta conveniente que dentro de las áreas de informática exista una persona (o grupo de personas) cuya función y

responsabilidad sea diseñar, mantener y administrar las bases de datos para un correcto funcionamiento de los sistemas. Esta función se conoce como *administrador de bases de datos*.

Cuando la problemática de las áreas de la institución o empresa sea común, se justificará la necesidad de que la estructuración de las bases de datos, catálogos y tablas sea orientada hacia la integración y consolidación de una *base de datos institucional*; en este sentido, será competencia de la dirección de informática el instrumentar un grupo de trabajo que se aboque a esta tarea. Cuando la problemática sea exclusiva de un área, la base de datos será particular.

[ISO 9001/4.4.2], [ISO 9001/4.4.4], [ISO 9001/4.4.5], [ISO 9001/4.9]  
Apéndice Norma ISO

#### 7.3.4. Definición de procesos

Debe hacerse una descripción detallada de los procesos que el sistema realizará, tales como: tomar datos de los almacenes, realizar operaciones y transformaciones a los datos, depositar nuevos datos en almacenes, validar datos, emitir resultados en pantallas o en impresoras, entre otros.

[ISO 9001/4.4.1], [ISO 9001/4.4.6] Apéndice Norma ISO

#### 7.3.5. Diseño de entradas y salidas

El diseño de las entradas, salidas e interfaces con el usuario deberá estar definido de acuerdo con estándares que permitan homogeneidad en la presentación de datos e informes a través de pantallas o reportes, sencillez y claridad en su presentación, así como facilidad de uso por parte de los usuarios.

[ISO 9001/4.4.1], [ISO 9001/4.4.4], [ISO 9001/4.5] Apéndice Norma ISO

**7.3.6. Diagrama estructural del sistema**

A través de la construcción del diagrama estructural del sistema, se describirá la jerarquía de los módulos que lo integren.  
[ISO 9001/4.5], [ISO 9001/4.9] Apéndice Norma ISO

**7.3.7. Combinación de los modelos**

Después de la fase de análisis se tienen conceptualizados los modelos estático, dinámico y funcional. El primero de ellos es el entorno principal alrededor del cual se construye el diseño. Es conveniente refinar las operaciones del modelo estático a través de la transformación de las acciones y actividades del modelo dinámico y a partir de los procesos del modelo funcional en operaciones asociadas a las clases del modelo estático. La realización de estas transformaciones hace corresponder la estructura lógica del modelo de análisis en una organización física del sistema.

[ISO 9001/4.4.3] Apéndice Norma ISO

**7.3.8. Importancia de la selección de algoritmos**

Toda operación especificada debe ser implantada mediante un algoritmo, éste muestra el cómo se hace la operación. Un algoritmo puede hacer llamadas a operaciones más sencillas, y así sucesivamente, hasta llegar a las operaciones del nivel más bajo, donde serán ejecutadas directamente sin más refinamiento. Un algoritmo implica la selección de las estructuras de datos a las cuales se aplican, es decir, se organiza la información de forma más eficiente.

[ISO 9001/4.4.5] Apéndice Norma ISO

### **7.3.9. Desarrollo de la herencia entre clases**

Una vez que se progresa en el análisis y diseño del sistema, es posible ajustar las definiciones de las clases y de los métodos por medio de la herencia. Es común que una misma operación se defina a través de varias clases, y es fácil que sea heredada de un antepasado común, las operaciones aplicadas a diferentes clases son parecidas pero no idénticas; así mediante sencillas modificaciones de las operaciones es factible abarcarlas empleando una sola operación heredada. Con el fin de aumentar la herencia dentro de un sistema es posible considerar los puntos siguientes.

- ◆ En algunas operaciones o métodos se pueden tener menos argumentos que en otros y se pueden añadir o ignorar los que falten.
- ◆ En algunas operaciones o métodos se pueden tener menos argumentos porque son casos especiales de otros más generales. Se deben establecer los métodos especiales llamando al método general con unos valores adecuados de los parámetros.
- ◆ Cuando haya atributos similares en clases distintas, pero con nombres diferentes, hay que dar el mismo nombre a los atributos y trasladarlos a una clase antecesora común. Así los métodos que accedan a los atributos coincidirán mejor.

La herencia es un proceso iterativo y de refinamiento que se vuelve a examinar en el diseño, buscando detalles comunes entre clases. Si un conjunto de métodos y atributos está repetido en dos clases, es posible que éstas sean en realidad variaciones especializadas de una misma cosa cuando se observan desde un nivel de abstracción más elevado.

Al reconocer un comportamiento común, es posible crear una superclase común que establezca las características compartidas, dejando solamente las especializadas en las subclases. Comúnmente la superclase resultante es abstracta, lo que significa que no tiene instancias directas de ella y que el comportamiento que define pertenece a las instancias de sus subclases.

[ISO 9001/4.4.7], [ISO 9001/4.4.8] Apéndice Norma ISO

## **8. Programación y documentación de sistemas**

### **8.1 Objetivo**

Generar los módulos, programas y rutinas a partir del diseño, para que el sistema funcione correctamente de acuerdo con las especificaciones del diseño; así como, generar e integrar toda la documentación requerida para contar con todas las especificaciones completas del sistema

### **8.2 Descripción**

El proceso de programación consiste en traducir el diseño lógico de un sistema a código fuente; para ello, se utilizará un lenguaje de alto nivel, o bien, un generador de código.

Durante la programación, mientras el sistema se convierte en una realidad, es de gran importancia documentar todos los procesos involucrados, con la finalidad de facilitar la etapa de mantenimiento, la cual es la más prolongada del ciclo de vida de un sistema; sin embargo, éste es uno de los procesos más olvidados y menos trabajados por la mayoría de los desarrolladores de sistemas.

[ISO 9001/4.4.5], [ISO 9001/4.5] Apéndice Norma ISO

### **8.3. Actividades de la programación y documentación de sistemas**

#### **8.3.1. Integración de bibliotecas de funciones**

Deberán identificarse las funciones comunes a varios módulos del sistema, tales como: dibujo de pantallas, validación de datos y configuración de dispositivos; éstas deberán programarse, probarse e integrarse en bibliotecas que permitan a los programadores utilizarlas en el sistema que se está desarrollando o inclusive en otro.

[ISO 9001/4.8], [ISO 9001/4.11], [ISO 9001/4.16] Apéndice Norma ISO

**8.3.2. Programación de módulos**

Los programas que constituyen el sistema deberán elaborarse en forma modular y con técnicas de programación adecuadas a la aplicación a resolver, incluyendo comentarios que faciliten su comprensión por cualquier persona del área de sistemas [Apéndice B].

[ISO 9001/4.4.1], [ISO 9001/4.8], [ISO 9001/4.9], [ISO 9001/4.16]  
Apéndice Norma ISO

**8.3.3. Manual técnico**

Deberá incluir una descripción detallada de los módulos y programas que constituyen el sistema, las funciones que se utilizan y los parámetros que requieren, así como de las estructuras de las bases de datos y archivos que lo conforman.

También deberán incorporarse los procedimientos de compilación y ligado de los diferentes componentes, así como los requerimientos de hardware y la forma de instalar el sistema.

[ISO 9001/4.5], [ISO 9001/4.9] Apéndice Norma ISO

**8.3.4. Manual del usuario**

Este manual es una descripción de la forma en que el usuario deberá interactuar con el sistema para llevar a cabo su trabajo.

Es necesario incluir todas las pantallas que se utilizan, así como ejemplos de todos los reportes que puede generar el sistema.

[ISO 9001/4.5], [ISO 9001/4.9] Apéndice Norma ISO

## **9. Pruebas**

### **9.1. Objetivo**

Asegurar el cumplimiento de los requerimientos del usuario mediante las pruebas efectuadas al sistema, garantizando al mismo tiempo que el software desarrollado se encuentre libre de errores y que se lleven a cabo en los términos acordados entre el desarrollador y el usuario.

### **9.2. Descripción**

Antes de utilizar un sistema, deberán hacerse las pruebas necesarias para poder asegurar que éste funcione correctamente en las condiciones de trabajo para las que fue diseñado. Para este efecto deben considerarse condiciones para con: los datos de prueba, el desempeño general y el cumplimiento de especificaciones iniciales del sistema. Una vez efectuadas las pruebas el desarrollador podrá realizar la liberación formal del sistema y el producto obtenido estará listo para su implantación.

[ISO 9001/4.10.1], [ISO 9001/4.11] Apéndice Norma ISO

#### **Plan de pruebas de sistemas**

Este plan establece el conjunto de pruebas, especificaciones y procedimientos a realizar al sistema desarrollado. Es común que su conformidad pueda plantearse como un acuerdo entre el desarrollador y el cliente. Este plan de pruebas describe esencialmente:

A. El establecimiento de un conjunto de pruebas que verificarán el sistema elaborado.

[ISO 9001/4.4.2] Apéndice Norma ISO

B. El tipo de pruebas efectuadas al sistema tales como las funcionales, de desempeño y de tensión.

[ISO 9001/4.11] Apéndice Norma ISO

C. Pruebas que permitan evaluar el rendimiento del sistema con base a la utilización de datos reales y comparación con sistemas anteriores.  
[ISO 9001/4.4.7], [ISO 9001/4.4.8] Apéndice Norma ISO

D. Considerar la opinión de los clientes como parte importante previo a la liberación del sistema.  
[ISO 9001/4.10.1] Apéndice Norma ISO

### 9.3. Actividades y tipos de pruebas

#### 9.3.1. Pruebas con datos reales

Antes de utilizar de manera definitiva cualquier sistema es necesario realizar pruebas utilizando datos reales; el personal del área usuaria debe colaborar estrechamente con el personal de sistemas en la definición y estructuración de los conjuntos de datos que se utilizarán para probar el sistema. Los resultados de estas pruebas deben ser verificados y aprobados por el área usuaria, debiendo dejar constancia por escrito de que los resultados fueron satisfactorios.

[ISO 9001/4.4.7], [ISO 9001/4.4.8] Apéndice Norma ISO

#### 9.3.2. Corridas en paralelo

Cuando un sistema va a reemplazar a uno que se encuentra operando, es necesario que se defina un periodo durante el cual funcionarán los dos sistemas, comparándose los resultados producidos por ambos en busca de anomalías en el nuevo sistema. Al final de dicho periodo, el usuario deberá hacer constar por escrito su satisfacción con el nuevo sistema, para poder eliminar el anterior y considerar al nuevo sistema terminado.

[ISO 9001/4.4.7], [ISO 9001/4.4.8] Apéndice Norma ISO

**9.3.3. Pruebas funcionales**

Estas pruebas especifican condiciones operativas, valores de entrada y resultados esperados comunes. Las pruebas de función comprueban el comportamiento dentro, sobre y más allá de las fronteras funcionales. Este tipo de pruebas se basa en las especificaciones y el cumplimiento de los requisitos.

[ISO 9001/4.4.7], [ISO 9001/4.4.8], [ISO 9001/4.11] Apéndice Norma ISO

**9.3.4. Pruebas de desempeño**

Estas pruebas se proyectan para verificar el tiempo de respuesta bajo cargas variables, porcentaje del tiempo de ejecución empleado en varios segmentos del programa, el rendimiento, la utilización de memorias primaria y secundaria, y las tasas de tráfico en los canales de datos y los enlaces de comunicación. Éstas deben ser diseñadas para demostrar que el sistema satisface sus requisitos.

[ISO 9001/4.4.7], [ISO 9001/4.4.8], [ISO 9001/4.11] Apéndice Norma ISO

**9.3.5. Pruebas de tensión**

Se diseñan para sobrecargar un sistema de varias maneras. Entre los ejemplos de las pruebas de tensión se intenta asignar más del número máximo de terminales permitidas, procesar más del número permitido de identificadores, desconectar un enlace de comunicaciones, etc..

[ISO 9001/4.4.7], [ISO 9001/4.4.8] Apéndice Norma ISO

**9.3.6. Pruebas de unidad**

Consisten en la evaluación individual de los componentes de software. Las pruebas de unidad deben realizarse previo a la

implantación, dichas pruebas se inician en los componentes fundamentales del sistema probados por unidad.  
[ISO 9001/4.4.7], [ISO 9001/4.4.8], [ISO 9001/4.10.1], [ISO 9001/4.11]  
Apéndice Norma ISO

**9.3.7. Pruebas de integración**

Se realizan para verificar y observar el desempeño en conjunto de todos los componentes del sistema. Éstas pueden efectuarse durante la evolución del sistema o bien realizarse únicamente al final.

[ISO 9001/4.4.7], [ISO 9001/4.4.8], [ISO 9001/4.10.1], [ISO 9001/4.11]  
Apéndice Norma ISO

**9.3.8. Pruebas de sistema**

Se aplican al sistema propuesto como producto final, se les considera de manera formal y previas a la liberación del sistema. Sirven para evaluar el producto de software obtenido y si éste satisface las expectativas, y requerimientos del cliente.

[ISO 9001/4.4.7], [ISO 9001/4.4.8], [ISO 9001/4.12] Apéndice Norma ISO

**9.3.9. Pruebas independientes**

En estas pruebas el sistema es evaluado por desarrolladores externos que cuenten con experiencia en la elaboración de sistemas. Cabe aclarar que éstos no forman parte del equipo que desarrolló el sistema, su tarea consiste en buscar errores o *bugs* que otros no suelen encontrar. Dichas pruebas producen un costo agregado al sistema elaborado.

[ISO 9001/4.4.6]] Apéndice Norma ISO

### **9.3.10. Pruebas beta**

Se presentan como una opción a la problemática que surge debido a que las personas que realizan las pruebas piensan inherentemente como desarrolladores y no como usuarios finales. De hecho, en estas pruebas el sistema se libera informalmente a un conjunto de usuarios previamente seleccionados, ellos están dispuestos a colaborar para mejorar el sistema a pesar de que no se encuentra liberado formalmente.

[ISO 90014.4.6], [ISO 90014.10.1], [ISO 90014.11] Apéndice Norma ISO

## **10. Implantación y capacitación**

### **10.1.Objetivo**

**Asegurar que la implantación del sistema se lleve a cabo en los términos acordados entre el desarrollador y el usuario, y que se culmine con un proceso de capacitación que permita obtener los beneficios esperados del sistema.**

### **10.2.Descripción**

Con la implantación de un sistema, culminan los esfuerzos realizados durante el análisis, el diseño y la programación. También debe ponerse especial atención en capacitar al personal que utilizará el sistema, pues de no hacerse así se utilizará incorrectamente y se seguirán realizando las actividades para las que fue desarrollado, como si éste no existiera, o no se aprovecharán todos los beneficios que proporciona.

[ISO 9001/4.18] Apéndice Norma ISO

### **10.3.Actividades para la implantación y capacitación**

#### **10.3.1. Configuración del desempeño**

El desarrollador del sistema verificará que la infraestructura involucrada para la instalación y desempeño del sistema esté puesta a punto. Es conveniente que el personal de la institución o empresa colabore activamente.

[ISO 9001/4.9], [ISO 9001/4.19] Apéndice Norma ISO

#### **10.3.2. Registro de la implantación**

La implantación del sistema deberá registrarse en las bitácoras de la plataforma que soporta el sistema de información, asentando

una breve descripción de las tareas realizadas para la puesta en marcha del sistema.

[ISO 9001/4.5], [ISO 9001/4.8] Apéndice Norma ISO

#### **10.3.3. Cursos de capacitación**

El desarrollador del sistema establecerá en conjunto con el cliente las fechas, duración y horario para efectuar los cursos de capacitación en relación al sistema.

[ISO 9001/4.18] Apéndice Norma ISO

#### **10.3.4. Niveles de capacitación**

Para el uso del sistema será válido efectuar distintos niveles de capacitación, tales como: instalación, operación, captura, administración básica o avanzada, dependiendo del puesto y las tareas que el personal desempeñará.

[ISO 9001/4.18] Apéndice Norma ISO

#### **10.3.5. Integración de sistemas al *banco institucional de sistemas***

Considerando el gran esfuerzo que representa el desarrollo de un sistema, es importante concentrar todos los desarrollos en un *banco institucional de sistemas*, para difundir las características de todos los desarrollos y ponerlos a disposición de otras áreas que puedan utilizarlos, evitando así la duplicidad de esfuerzos y optimizar los recursos.

Al contar con la aprobación del área usuaria y del área de informática para liberar un sistema, el área de informática debe entregar el original de la documentación del sistema con todos sus componentes a la dirección de informática para su integración al *banco institucional de sistemas* y conservará una copia de lo mismo que entregó.

[ISO 9001/4.5], [ISO 9001/4.10.1], [ISO 9001/4.10.5], [ISO 9001/4.16]  
Apéndice Norma ISO

### 10.3.6. Capacitación

Para que el personal del área usuaria pueda interactuar con el nuevo sistema, es necesaria una etapa de entrenamiento, con características pedagógicas adecuadas, que permita asegurar la correcta operación, así como la explotación de todos los beneficios ofrecidos por el sistema.

También es importante capacitar al personal del área de sistemas para que pueda hacerse cargo de las actividades de mantenimiento necesarias para asegurar que el sistema opere correctamente y pueda adaptarse a nuevos requerimientos.

[ISO 9001/4.18] Apéndice Norma ISO

## **11. Mantenimiento de sistemas**

### **11.1. Objetivo**

Garantizar la permanencia en operación de un sistema, mejorándolo, adaptándolo a nuevos requerimientos o corrigiendo problemas que sean detectados durante su operación.

### **11.2. Descripción**

El mantenimiento de un sistema involucra todas las etapas de su desarrollo. Cuando el objetivo es mejorarlo o adaptarlo, el mantenimiento reinicia los trabajos del desarrollo en la etapa de análisis. Cuando se trata de corregir un problema puede reiniciarse en el análisis, el diseño o la programación, por lo que esta parte del ciclo de vida de un sistema queda sustentada en las secciones anteriores de este manual.

Antes de modificar un sistema debe analizarse cuidadosamente si dicha modificación está justificada; de ser así debe procederse con la misma metodología utilizada durante el desarrollo, para llevar a cabo nuevamente las fases que sean necesarias del análisis, diseño, programación, pruebas e implantación, poniendo especial atención en dejar una documentación completa y clara de los cambios efectuados, ya que de no hacerlo puede resolverse temporalmente un problema, pero también se contribuye a la rápida degradación del sistema.

[ISO 9001/4.5], [ISO 9001/4.8], [ISO 9001/4.9], [ISO 9001/4.15.5] [ISO 9001/4.16] Apéndice Norma ISO

## **Plan para el mantenimiento de sistemas**

Este plan debe describir esencialmente lo siguiente.

### **A. Establecimiento del alcance del mantenimiento.**

[ISO 9001/4.3], [ISO 9001/4.19] Apéndice Norma ISO

### **B. Identificación y descripción del estado inicial del sistema.**

[ISO 9001/4.13] Apéndice Norma ISO

### **C. Descripción de actividades de mantenimiento, tales como:**

- Uso de los mismos procedimientos del desarrollo.
- Solución de problemas, en base a la identificación del tipo de mantenimiento.
- Modificación de interfaz, en base a la identificación del tipo de mantenimiento.
- Mejoramiento del desempeño, en base a la identificación del tipo de mantenimiento.

[ISO 9001/4.4.1], [ISO 9001/4.9] Apéndice Norma ISO

### **D. Descripción del mantenimiento de registros y reportes,**

- Lista de reportes del sistema que mostrará entre otras cosas:
  - ◆ Organización o área responsable.
  - ◆ Describir la prioridad de cambios a efectuar.
  - ◆ Puntualizar los resultados después de efectuar las acciones.
  - ◆ Explicar datos generados en base a errores y cambios efectuados.
- La liberación de reportes mostrará entre otras cosas:
  - ◆ Establecimiento de las reglas de mantenimiento, considerando lo modificado al sistema.
  - ◆ Descripción de los tipos de alcance.

- ◆ Estipulación de la regresión de los métodos de prueba del sistema.
- ◆ Estipulación para el mantenimiento por parte del desarrollador y del cliente.

[ISO 9001/4.4.9], [ISO 9001/4.8], [ISO 9001/4.16] Apéndice Norma ISO

#### E. Definición del mantenimiento que dará soporte a la organización.

- Representantes de ambas partes tanto del desarrollador como del cliente.
- Identificación de facilidades y recursos.

[ISO 9001/4.3], [ISO 9001/4.19] Apéndice Norma ISO

### 11.3. Actividades para el mantenimiento de sistemas

#### 11.3.1. Identificación del tipo de mantenimiento

El mantenimiento más que modificar un programa para *corregir errores*, consiste en crear o cambiar estructuras de datos y procesos lógicos de programas ya existentes, utilizando para ello la documentación existente. Podemos describir y clasificar el mantenimiento en cuatro actividades que se llevan a cabo después de entregar un sistema.

##### *Mantenimiento correctivo*

Durante el uso de cualquier sistema es común descubrir nuevos errores, debido a que no es razonable asumir que la pruebas del software haya descubierto todos los errores latentes del sistema; de tal forma que se informará al equipo de desarrollo de éstos. A este proceso que incluye el diagnóstico y la corrección de uno o más errores se le conoce como mantenimiento correctivo. Suelen encontrarse problemas como:

- ◆ *Fallas de procesamiento.* Terminación anormal del sistema, pérdida de datos de entrada, salidas incorrectas del sistema, etc.
- ◆ *Fallas de rendimiento.* Procesamiento inadecuado o tiempo de respuesta lento.
- ◆ *Fallas de implantación.* Violaciones e inconsistencias en el diseño, manejo inadecuado de estándares, etc.

[ISO 9001/4.14.2] Apéndice Norma ISO

*Mantenimiento adaptativo*

Éste se produce por el rápido cambio inherente a cualquier aspecto de la industria del cómputo. Frecuentemente escuchamos que aparecen nuevas generaciones de hardware, nuevas versiones de sistemas operativos y surgen nuevas versiones de software de aplicación. Se sabe que algunas veces al realizar un cambio en una parte de un sistema, se tienen que realizar cambios a otras partes y son precisamente estos cambios secundarios que dan el nombre de adaptativo. En síntesis, es una actividad que modifica el software para que interactuación adecuadamente con su entorno cambiante. Este tipo de mantenimiento se presentan cuando hay:

- ◆ Cambios en el medio ambiente de los datos, es decir, se realizan cambios en la estructura de la base de datos.
- ◆ Cambios al procesamiento del medio ambiente, se efectúa al migrar un sistema a una nueva plataforma de hardware o aun nuevo sistema operativo.

[ISO 9001/4.4.9], [ISO 9001/4.15.5] Apéndice Norma ISO

*Mantenimiento perfecto*

Éste se realiza debido a recomendaciones de usuarios sobre nuevas posibilidades, modificaciones sobre funciones ya

existentes y sobre mejoras en general. Se ve como una actividad para mejorar una futura facilidad de mantenimiento o fiabilidad, y proporciona una base para futuras mejoras. Se caracteriza por:

- ◆ Cambios al diseño o al código para mejorar el rendimiento del sistema, es decir, se mejora el acceso al sistema y el tiempo de respuesta.
- ◆ Cambios o adición de nuevas características al sistema.

[ISO 9001/4.4.9], [ISO 9001/4.15.5] Apéndice Norma ISO

#### **Mantenimiento preventivo**

Este se realiza sobre el sistema en un esfuerzo para prevenir el origen de un posible error o que pueda ocurrir una falla en una función dada.

[ISO 9001/4.14.3] Apéndice Norma ISO

### **11.3.2. Solicitud de modificación**

Cualquier modificación a un sistema debe solicitarse por escrito, describiendo detalladamente en qué consiste y cuáles son las razones que la justifican.

[ISO 9001/4.4.9], [ISO 9001/4.14.2] Apéndice Norma ISO

### **11.3.3. Análisis de la solicitud de modificación**

Antes de iniciar los trabajos necesarios para la modificación de un sistema, debe analizarse cuidadosamente la justificación de la solicitud presentada. Si la modificación no se considera necesaria, debe elaborarse un documento en donde se aclare el porqué y se propongan caminos alternativos para resolver la problemática planteada en la solicitud.

[ISO 9001/4.4.9], [ISO 9001/4.14.2] Apéndice Norma ISO

**11.3.4. Aprobación de la solicitud y elaboración de un programa de trabajo**

Una vez aprobada la modificación de un sistema, tanto por el área usuaria como por la dirección de informática, debe hacerse constar por escrito los alcances de dicha modificación. También es importante estructurar un programa de trabajo en donde se detallen las actividades necesarias para llevar a cabo la modificación, así como su duración y los responsables, dicho programa de trabajo deberá ser aprobado y firmado por las áreas involucradas.

[ISO 9001/4.4.1] . [ISO 9001/4.4.9], [ISO 9001/4.14.2] Apéndice Norma ISO

12. Tabla de formas o formatos para la documentación de sistemas

Etapas del desarrollo	Responsables			Forma	Productos
	U	I	E		
Análisis de sistemas	X				Solicitud de desarrollo
		X		DA01	Estudio de factibilidad
	X	X	X	DA02	Calendario de trabajo
		X	X	DA03	Relación de documentos
		X	X	DADMF04	Diagramas de flujo de datos
		X	X	DADMF05	Diccionario de términos
		X	X	DADME06	Diagrama de clases
		X	X	DADME07	Diccionario de clases
		X	X	DADMD08	Diagrama de seguimiento de sucesos
		X	X	DADMD09	Diagrama de estados
Diseño de sistemas	X	X	X	DA10	Acta de acuerdos
		X	X	DD01	Selección o identificación de software y hardware
		X	X	DD02	Diagrama entidad-relación
		X	X		Estructuras de bases de datos y archivos
		X	X	DD03A	Estructura de tablas o archivos
		X	X	DD03B	Registro de llaves secundarias
		X	X	DD03C	Registro de archivos de índices
		X	X	DD04	Diccionario de datos
		X	X	DD05	Descripción de procesos
		X	X	DD06	Entradas y salidas
	X	X	DD07	Diagrama estructural del sistema	
Programación documentada		X	X	DP01	Descripción de programas y rutinas
		X	X		Manual del usuario
		X	X		Manual técnico
Pruebas del sistema	X	X	X	DI01	Resultados de pruebas
Implantación de sistemas y capacitación		X		DI02	Integración al banco institucional de sistemas
Mantenimiento de sistemas	X			DM01	Solicitud de mantenimiento
		X	X	DM02	Resultado de solicitud
	X	X	X	DA02	Calendario de trabajo

U = Usuario    I = Área de Informática    E = Empresa responsable del desarrollo

### 13. Formas o formatos para la documentación de sistemas

A continuación se anexan las formas que permitirán documentar las diferentes etapas del *ciclo de vida de un sistema*.

Para cada forma se establecen las indicaciones para su llenado, con el mismo número de forma, la forma en blanco se encuentra a continuación de la(s) de indicaciones.

Para facilitar su uso, en una(s) copia(s) de cada una de todas las formas se explica el llenado de todos y cada uno de los elementos que deberán registrarse.

## FORMAS Y FORMATOS

*Los principios importantes pueden ser flexibles.*

*Lincoln*





## ESTUDIO DE FACTIBILIDAD

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_  
PROYECTO \_\_\_\_\_ ABREV \_\_\_\_\_ HOJA 1 DE 2

Nombre del área usuaria

Nombre del proyecto del sistema

Número de identificación de la forma

Identificación de la clave presupuestal del área usuaria

Identificación de la forma o formato

Nombre de la persona que elaboró el Formato

Fecha de elaboración

Abreviación del nombre del proyecto del sistema

Clave de referencia del proyecto en desarrollo.  
Se construye con el número de la forma más las siglas de abreviación más número consecutivo.

**Ejemplo: DA01-SIP-01**

Nombre de la persona que autorizó la forma

Fecha de actualización

Identificación del número de hoja con respecto al total de hojas

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ESTUDIO DE FACTIBILIDAD

ÁREA: \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA 2 DE 2

*En esta forma, el o los analistas presentarán los resultados del estudio de factibilidad. Con el propósito de "estandarizar" las evaluaciones de los expertos técnicos, económicos y operacionales propios de la problemática del estudio, se sugiere el siguiente contenido mínimo:*

1. *Definición de objetivos.*
2. *Determinación de recursos.*
- 2.1. *Factibilidad técnica.*

*Recursos tecnológicos necesarios, existentes y por adquirir.*

*Capacidad adecuada y suficiente del equipo para soportar el nuevo sistema, así como su crecimiento*

*Capacidad del sistema para responder al número y ubicación de los usuarios.*

*Garantías técnicas de exactitud, confiabilidad, facilidad de acceso y seguridad de datos.*

- 2.2. *Factibilidad económica.*

*Costos del desarrollo completo del sistema.*

*Costos del software y del hardware.*

*Situación financiera del área y fuentes de recursos para cubrir el proyecto.*

- 2.3. *Factibilidad operacional.*

*Apoyos para llevar a cabo el desarrollo del sistema.*

*Aceptación por parte de usuarios.*

*Resultados esperados al entrar en operación el nuevo sistema (usuarios directos e indirectos)*

3. *Ventajas y desventajas de la implantación del sistema.*
4. *Conclusiones.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ESTUDIO DE FACTIBILIDAD

AREA \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

Large empty rectangular box for the main content of the feasibility study.

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_





## CALENDARIO DE TRABAJO

AREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

FECHA INICIO \_\_\_\_\_ FECHA TERMINO \_\_\_\_\_ TIEMPO TOTAL PROGRAMADO \_\_\_\_\_ TIEMPO TOTAL REAL \_\_\_\_\_

DESCRIPCION DE LA ACTIVIDAD	RESPON	TIEMPO	TIEMPO EN: DIAS ( )							SEMANAS ( )														
			1	2	3	4	5	6	7	8	1	1	1	1	1	1	1	2	2	2	2	2	2	2
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						
		ESTRADO																						
		REAL																						

OBSERVACIONES

ELABORÓ \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## RELACION DE DOCUMENTOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE \_\_\_\_\_

NOMBRE	FUENTE QUE ORIGINA EL DOCUMENTO	DESCRIPCIÓN DEL DOCUMENTO	FECHA EN QUE SE RECOPILO
<p style="text-align: center;">↑</p> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Nombre del documento, forma o formato</i> </div>	<p style="text-align: center;">↑</p> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Indicar la fuente que origina el documento (área, organismo, institución, etc.)</i> </div>	<p style="text-align: center;">↑</p> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Indicar el tipo de información que recaba o genera el documento, describiendo brevemente su contenido</i> </div>	<p style="text-align: center;">↑</p> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; width: fit-content; margin: 0 auto;"> <i>Indicar la fecha en que el documento fue recopilado para la tarea de análisis</i> </div>

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## RELACIÓN DE DOCUMENTOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE: \_\_\_\_\_

NOMBRE	FUENTE QUE ORIGINA EL DOCUMENTO	DESCRIPCIÓN DEL DOCUMENTO	FECHA EN QUE SE RECOPIÓ

ELABORO: \_\_\_\_\_

AUTORIZO: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE FLUJO DE DATOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_  
 PROYECTO \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA 1 DE 5

*Se considera a un sistema como un conjunto de componentes interrelacionados, con una frontera que los distingue de su ambiente, cuyas tres funciones principales son: el almacenamiento, la transmisión y la transformación de información. Un diagrama de flujo de datos, debe representar dichas funciones en forma abstracta, utilizando un conjunto mínimo de símbolos.*

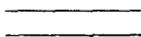
### SÍMBOLOS

*Los diagramas de flujo de datos deberán utilizar los siguientes símbolos:*

- Arcos terminados en flecha para representar flujo de datos;
- Líneas paralelas para representar almacenamiento de datos;
- Círculos para representar transformaciones (procesos) y
- Cuadrados para representar el origen o el destino de la información.



*Flujo de datos*



*Almacenamiento*



*Transformación*



*Origen o destino*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE FLUJO DE DATOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO. \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA 2 DE 5

*Un diagrama de flujo de datos no debe mostrar el flujo de control, es decir, solamente debe mostrar las posibles alternativas de entradas y salidas. No las decisiones, ni la secuencia temporal con que ocurren los procesos.*

*Cada componente de un diagrama de flujo de datos debe tener un nombre único, que permita distinguirlo del resto de los elementos, así como referenciarlo con la descripción de relaciones funcionales entre módulos y con el diccionario de términos; en la medida de lo posible, este nombre debe seleccionarse de manera que resulte significativo, con el objeto de facilitar la lectura del diagrama.*

*Un sistema complejo no puede ser descrito por completo mediante un solo diagrama de flujo de datos, sino que debe dividirse en una estructura jerárquica, en la que se parta de un diagrama de contexto y se vaya profundizando en diferentes diagramas vinculados a éste.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

FORMA/DADMF04



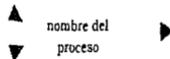
## DIAGRAMA DE FLUJO DE DATOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: 3 DE 5

*El modelo funcional se encarga de especificar el significado de las operaciones y de las restricciones. Un diagrama de flujo de datos, muestra las relaciones entre los valores calculados por un sistema, incluyendo los valores introducidos, los obtenidos y los almacenes internos.*

**Proceso:**



**Flujo de datos entre procesos:**



*Los procesos se dibujan en forma de elipses que contienen una descripción de la transformación, cada proceso tiene un número fijo de flechas de entrada y salida de datos, cada una de las cuales lleva un valor de tipo dato, las entradas y salidas pueden comentarse con el fin de mostrar su tarea en el cálculo.*

**Objeto Entidad (como fuente o receptor de datos)**



*Una Entidad es un objeto activo que controla el flujo de datos produciendo o consumiendo valores, las acciones de las Entidades van más allá del alcance del DFD.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE FLUJO DE DATOS

ÁREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA: 4 DE 5

*Un almacén de datos es un objeto pasivo dentro de esta clase de diagramas, su papel es simbolizar el lugar (o la estructura) en la cual se almacenan los datos para su posterior utilización. No generan operaciones por sí mismos, sino que se limitan a responder solicitudes de almacenamiento y acceso a datos.*

Almacén de datos u objeto archivo:

.....  
nombre del  
almacén de datos  
.....

Almacén de datos u objeto archivo:

.....  
nombre del  
almacén de datos  
.....

*El acceso a los datos de los almacenes es representado mediante flechas que pueden ir de procesos a los almacenes. Las flechas de salida indicarán que alguna información se ha extraído del almacén. Las flechas de entrada al almacén indicarán que deposita algún dato en el mismo.*

Acceso a un valor de una almacén de datos:

Almacén de datos  
  
▼ d1  
▶ nombre del  
proceso

Actualización de un valor de un almacén de datos

Almacén de datos  
  
▲ d1  
▶ proceso

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE FLUJO DE DATOS

AREA: \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: 5 DE: 5

*Para simplificar el uso de flechas de entrada y salida de un almacén de datos, es valido el poner una flecha don dos cabezas que represente tanto al acceso como la actualización de los datos.*

Acceso y actualización de un valor en un almacén de datos

Almacén de datos



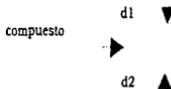
proceso

*El simbolo de flujo de datos es una flecha. Este simbolo conecta la salida de un proceso con la entrada de otro proceso. Representa un valor de datos intermedio dentro de un cálculo que no es modificado por el flujo de datos. En ocasiones un conjunto de datos se descompone con el fin de ir a procesos diferentes, y de igual forma se pueden integrar para ir a procesos comunes. para estos casos se tienen las siguientes convenciones.*

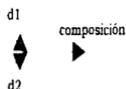
Duplicación de un valor de datos



Descomposición de un valor de datos



Composición de un valor de datos



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE FLUJO DE DATOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

Blank area for the Data Flow Diagram (DFD).

ELABORO: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZO: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DICCIONARIO DE TÉRMINOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

TÉRMINO	DESCRIPCIÓN

*Nombre del término tal como se utiliza en el sistema o procedimiento actual*

*Descripción del término, utilizar frases cortas y directas con el propósito de identificar claramente cada uno de los elementos del sistema*

*Esta forma registrará todas las definiciones de los elementos del sistema en sus flujos de datos, procesos y almacenamiento de datos  
Esta forma complementa la información descrita en la forma DADMF04  
"Diagramas de Flujo de Datos"*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

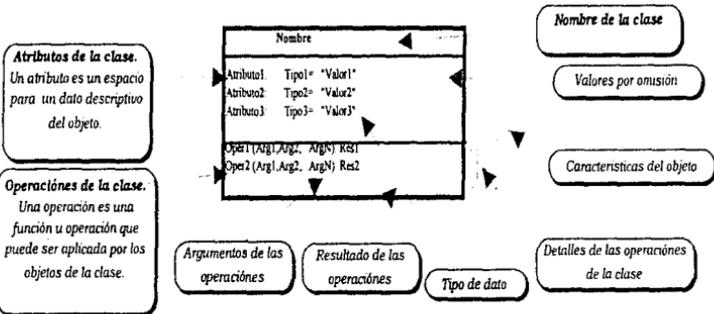




## DIAGRAMA DE CLASES

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE. REFERENCIA \_\_\_\_\_  
 PROYECTO: \_\_\_\_\_ ABBREV.: \_\_\_\_\_ HOJA. 1 DE 6

*Los diagramas de clases (o de clases de objeto) son esquemas plantillas para describir instancias de posibles datos. Un diagrama de clases puede dar lugar a un sinnúmero de diagramas de objetos (también llamados de instancia) Describen el caso general del modelado del sistema, si bien puede mezclarse con diagramas de objetos, esto no es recomendable.*



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE CLASES

AREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABRUV: \_\_\_\_\_ HOJA 2 DE 6

En los diagramas de clases se aplican los denominados "enlaces". Un enlace muestra una asociación entre dos o más objetos la cual se establece de manera bicondicional entre los objetos implicados, es decir, válida en todos los sentidos que se expresen, aunque se exprese de la misma forma.

Clase1 Atrib1	nombre de la asociación	Clase2 Atrib2	<b>Asociación</b>
------------------	----------------------------	------------------	-------------------

Como cada clase representa a un conjunto de instancias objetos, la manera de especificar la conación de estas relaciones es mediante los siguientes símbolos:

1+	Clase	1-2,4	Clase	●	Clase	○	Clase
	Una o más	Especificadas numéricamente		Muchas (cero o más)		Opcional (cero o una)	Exactamente una

Clase1 Atrib1	nombre de la asociación	Clase2 Atrib2	{ordenada}	●	Clase
	calificadiv				<b>Orden</b>
	rol-1	rol-2			

**Asociación calificada**

Existen una serie de anotaciones que pueden anexarse a un enlace establecido

**Cualificador:** Fija un atributo especial que establece la asociación a algunas instancias que cumplan con ciertas características

**Rol:** Es un nombre que identifica de forma única un extremo de la asociación, este nombre permite visualizar el papel que juega la instancia (o instancias) de una clase con respecto a otra clase y en algunos casos a ella misma.

El orden es una clasificación inherente de objetos involucrados en la asociación, mediante la cual se especifica la secuencia en que se presentan las clases.

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

FORMA/DADME06

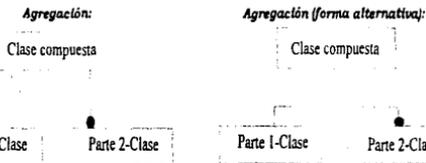


## DIAGRAMA DE CLASES

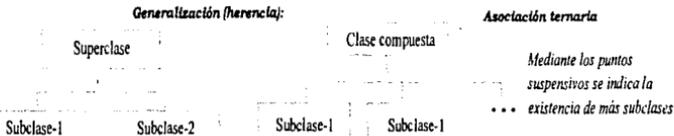
ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: 3 DE: 6

*A través de la generalización se expresa cómo los objetos pueden componer a otros objetos. La manera de expresarse es mediante un rombo que se indica en el extremo de la parte compuesta. Es válido emplear los cuantificadores. Se tienen formas alternativas y detalladas para expresar la relación entre objeto compuesto y objetos componentes.*



*Los conceptos de abstracción de generalización y herencia se expresan mediante un triángulo que une a las subclases implicadas. Mediante este símbolo, se expresa cómo una subclase comparte similitudes y establece sus diferencias, de forma tal que las clases de arriba son más generales, y las de abajo son más especializadas y heredan características de la clase superior (superclase).*



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

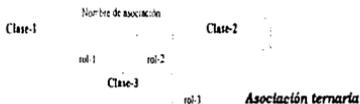


## DIAGRAMA DE CLASES

ÁREA \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

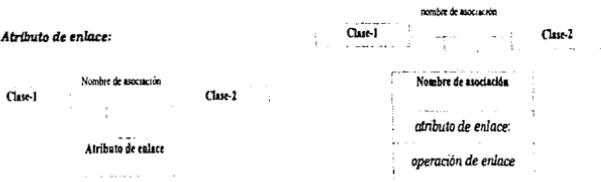
PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: 4 DE: 6

*De la misma forma que se tienen enlaces binarios entre clases de objetos, es posible tener enlaces ternarios, y (n)arios entre clases de objetos, la forma de expresar estos enlaces es mediante un rombo en la parte central de la línea que conforma el enlace. Una asociación ternaria no puede descomponerse en relaciones binarias sin perder información*



*Así como una clase posee atributos que conforman características sobresalientes de la clase, un enlace puede presentar características propias. La notación usada para este caso, es un rectángulo con dos regiones unido al enlace mediante un arco, en el cual se describen los atributos de ese enlace. Esta clase de atributos pueden presentarse en enlaces ternarios y (n)arios.*

**Atributo de enlace:**



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

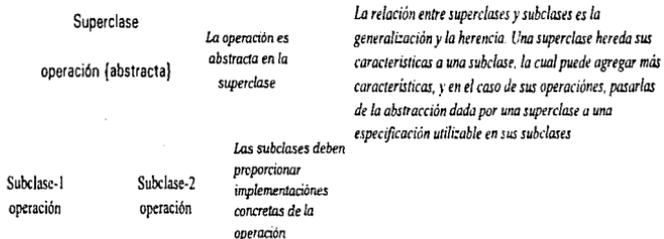


## DIAGRAMA DE CLASES

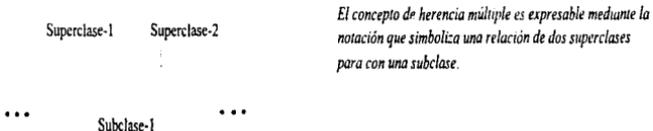
ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA 5 DE 6

### Generalización (herencia):



### Herencia múltiple



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

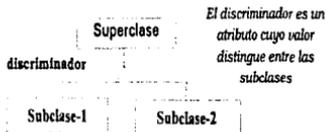


## DIAGRAMA DE CLASES

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA 6 DE 6

*Para establecer el mecanismo de herencia se usa un triángulo como el mostrado a continuación, pero además puede usarse el discriminador, que es un atributo que nos indica que propiedad está siendo abstraída, sólo es posible abstraer una propiedad a la vez.*



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

**DIAGRAMA DE CLASES**

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

Blank area for the Class Diagram.

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DICCIONARIO DE CLASES

ÁREA \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_  
PROYEC: Nombre de la clase Atributos definidos en la clase \_\_\_\_\_ HOJA: 1 DE: 1

NOMBRE:	ATRIBUTOS	DESCRIPCIÓN DE LA CLASE
SUBCLASE DE:	<i>Nombre de la clase de la cual se derivó</i>	
HERENCIA:	<i>Tipo de herencia (ninguna, simple, compuesta, etc.)</i>	
NOMBRE Y TIPO DE METODO	DESCRIPCIÓN DEL METODO	
<i>Tipo de método (público, privado, protegido, etc.)</i>	<i>Descripción o referencia del método de clase, utilizado</i>	<i>Descripción de la clase, utilizar frases cortas y directas con el propósito de identificar claramente cada uno de los elementos del sistema</i>
<i>Tiene como objetivo apoyar y complementar la notación gráfica formal usada para el modelado de objetos, clases y sus relaciones entre sí. En su contenido se debe expresar en forma explícita aquellos detalles que por extensión no puedan documentarse en la notación gráfica. Se debe procurar no ser redundantes en los conceptos y también no ser demasiado breves.</i>		

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

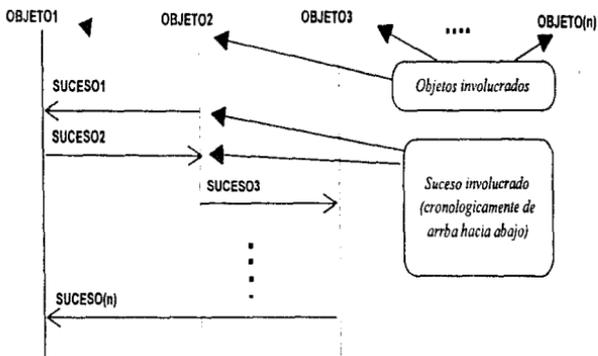




## DIAGRAMA DE SEGUIMIENTO DE SUCESOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_



*Los diagramas de seguimiento de sucesos, tienen como objetivo completar la documentación que describe la secuencia de operaciones que se producen en respuesta a estímulos externos. Esta secuencia de sucesos se expresa mediante el diagrama de seguimiento. Este diagrama muestra a cada objeto como una línea vertical, cada suceso como una flecha horizontal que va desde el objeto emisor al objeto receptor. La secuencia se genera de arriba hacia abajo, conformando la descripción completa de los escenarios.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE SEGUIMIENTO DE SUCESOS

A.R.: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

--	--

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA DE ESTADOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: 1 DE 3

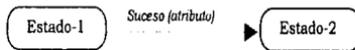
Los diagramas usados en el modelo dinámico representan la parte concerniente al control y secuencia de tareas que presentan los objetos ante estímulos externos, sin tomar en cuenta lo que hacen las operaciones. Las unidades principales de estos diagramas son los estados, que representan los valores de los objetos en un tiempo determinado, y los sucesos que son los estímulos externos que provocan el cambio de un estado a otro.

### Un suceso produce transición entre estados



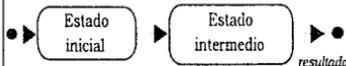
Un suceso es una transmisión de información con dirección única entre un objeto y otro. Los diagramas expresan cambios de estado de un solo objeto y se usan cuando dicho comportamiento sea importante.

### Suceso con atributo



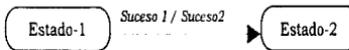
Cuando un suceso aporta información de un objeto a otro, dicha información se especifica mediante el atributo, que son valores de datos del objeto.

### Estados Inicial y final



Cuando existen estados inicial y final se especifica con un punto negro, y con un punto negro encerrado en un círculo respectivamente. Estos diagramas pueden ser subrutinas de otro diagrama y simbolizan la creación (inicio) y destrucción (final) de un objeto.

### Acción sobre una transición



Una acción es una operación instantánea que va asociada a un suceso, se expresa con "/" que separa el suceso principal del suceso secundario o acción. Se usa cuando se quiere especificar acciones a detalle mediante un suceso inicial, o final, y posiblemente intermedios.

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

FORMA/DADMD09

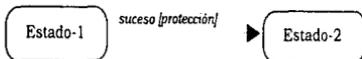


## DIAGRAMA DE ESTADOS

ÁREA \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA 2 DE 3

### Transición protegida



*Si bien es posible definir una transición de estados dado un suceso, también es posible establecer, mediante condiciones de protección, un flujo más específico, el cual permitirá la transición si y sólo si es verdadera dicha condición.*

### Acciones y actividad en un estado

**Nombre del estado**  
 entrada / acción de entrada hacer:  
 actividad-A  
 suceso-1 / acción-1  
 ....  
 salida / acción-salida

*Un suceso puede dar lugar a que se lleve a cabo una acción sin producir un cambio de estado. El nombre del suceso se escribe dentro del cuadro de estado y va seguido de ":" y el nombre de la acción. Las palabras reservadas son: entrada, salida y hacer. En este caso al producirse un suceso, las acciones de entrada y salida no se ejecutan puesto que no se sale del estado, sólo se efectúa la actividad precedida de la palabra hacer.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

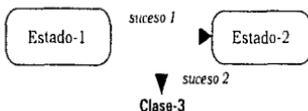


## DIAGRAMA DE ESTADOS

ÁREA: \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

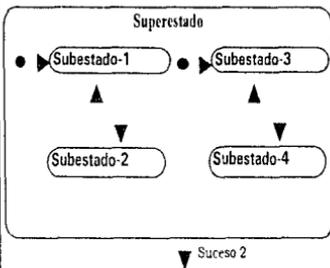
PROYECTO: \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA: 3 DE 3

### Envío de un suceso a otro objeto:



*Mediante esta notación se expresa el envío de un suceso de un objeto a otro. La línea discontinua que va desde una transición a un objeto, indica que se envía un suceso al objeto cuando se dispara la transición.*

### Subdiagramas concurrentes



*También es posible representar la situación de que un objeto pueda llevar a cabo dos o más actividades concurrentemente, clasificándose como subestados de superestados, los cuales solo entrarán en actividad cuando el superestado principal esté activo.*

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

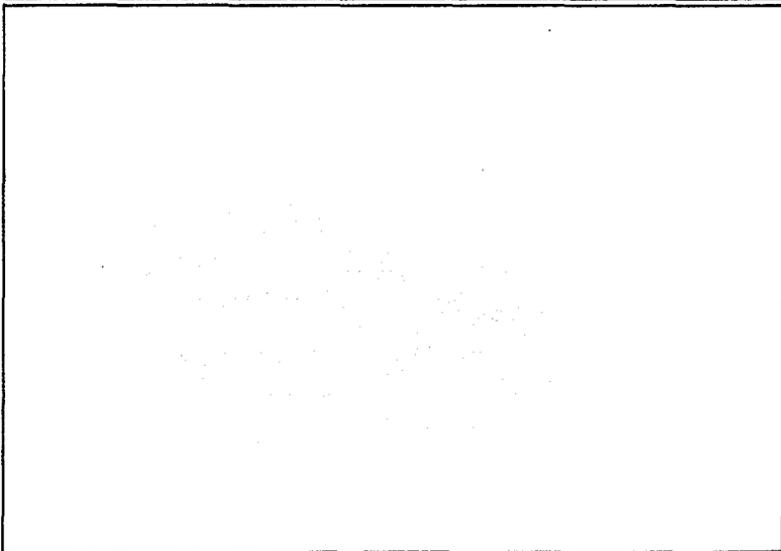
AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

**DIAGRAMA DE ESTADOS**

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE: \_\_\_\_\_



ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ACTA DE ACUERDOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO. \_\_\_\_\_ ABREV \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

### ASISTENTES:

*Nombres y puestos de los asistentes a la reunión de acuerdos para el desarrollo del sistema*

### PUNTOS TRATADOS:

*Enumerar los puntos tratados en la reunión de acuerdos*

### CONCLUSIONES Y ACUERDOS:

*Enumerar las conclusiones y acuerdos tomados*

### COMPROMISOS:

ACTIVIDAD

RESPONSABLE

FECHAS

*Describir los compromisos establecidos, así como los responsables y fechas de respuestas*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ACTA DE ACUERDOS

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

**ASISTENTES:**


**PUNTOS TRATADOS:**


**CONCLUSIONES Y ACUERDOS:**


**COMPROMISOS:**

ACTIVIDAD	RESPONSABLE	FECHAS

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## SELECCIÓN E IDENTIFICACIÓN DE SOFTWARE Y HARDWARE

ÁREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

SOFTWARE/HARDWARE

USO

*En esta columna deberá hacerse una lista de los elementos tanto de software como de hardware que se requieren para el adecuado funcionamiento del sistema, incluyendo la descripción técnica de sus características.*

*Al lado de cada elemento de software o hardware, deberá indicarse una breve descripción de la función que éste tiene dentro del contexto del proyecto.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

**SELECCIÓN E IDENTIFICACIÓN DE SOFTWARE Y HARDWARE**

ÁREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE. REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

SOFTWARE/HARDWARE

USO

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA ENTIDAD-RELACIÓN

AREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA: 1 DE 3

*El propósito de este diagrama es definir un modelo en el que se tengan identificadas las entidades que intervienen en el sistema, así como las relaciones o vínculos entre ellas. El diagrama entidad-relación constituye una abstracción lógica que servirá como base para especificar posteriormente las tablas de datos del sistema y sus campos.*

### SÍMBOLOS:

*Los diagramas entidad-relación deberán utilizar los siguientes símbolos:*

*Rectángulos para representar entidad*

NOMBRE\_DE\_ENTIDAD

*Elipses para representar atributos*

*Rombos para representar relaciones a vínculos*

NOMBRE\_DE\_LA\_RELACION

NOMBRE\_ATRIBUTO

*Cada atributo se asocia a una entidad o relación, mediante un segmento de línea*

NOMBRE\_ATRIBUTO

NOMBRE\_ATRIBUTO

NOMBRE\_ATRIBUTO

NOMBRE\_ATRIBUTO

NOMBRE\_DE\_ENTIDAD

NOMBRE\_DE\_LA\_RELACION

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



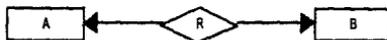
## DIAGRAMA ENTIDAD-RELACIÓN

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

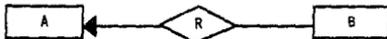
PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA 2 DE 3

*La cardinalidad de las relaciones entre entidades se representa mediante líneas dirigidas. A continuación se muestran los tres casos típicos de cardinalidad entre las entidades A y B vinculadas por la relación R.*

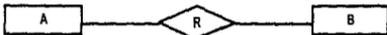
- Uno a uno:



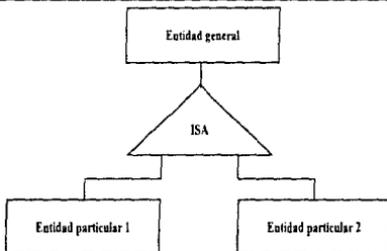
- Uno a muchos:



- Muchos a muchos:



*Las relaciones de generalización ISA ("es un") se representa mediante un triángulo, que apunta hacia la entidad más general y del que se desprenden las entidades más particulares. A continuación se ilustra una relación de este tipo.*



ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA ENTIDAD-RELACIÓN

AREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA 3 DE 3

### OBSERVACIONES:

- Los nombres de las entidades y relaciones deberán ser únicos y preferentemente dar una idea de lo que representan en el mundo real. En el caso de relaciones a las que no se les encuentre un nombre apropiado, deberá utilizarse uno que se componga de prefijos de las entidades a las que vincula.
- Deberá evitarse el repetir una misma entidad en diferentes partes de un diagrama, ya que esto puede disminuir la claridad y dificultar su interpretación.
- Cada atributo deberá asociarse a una y sólo una entidad o relación. Esto no excluye la posibilidad de que diferentes entidades o relaciones tengan atributos del mismo tipo.
- Se recomienda que se utilicen únicamente relaciones binarias (que vinculan a sólo dos entidades), aunque es posible representar relaciones de mayor orden.

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DIAGRAMA ENTIDAD-RELACIÓN

ÁREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ESTRUCTURA DE TABLAS

ÁREA \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

NOMBRE DEL CAMPO	TIPO DE DATO	LONGITUD DEL CAMPO	PRECISIÓN	RANGO DE VALORES	LLAVE PRIMARIA	VALOR DEFAULT
<i>Nombre del campo</i>	<i>Tipo de dato (carácter, numérico, etc.)</i>	<i>Número de dígitos a la derecha del punto decimal</i>	<i>Indicar si tiene un valor por omisión</i>	<i>Marcar si acepta valores nulos, así como el rango de valores</i>	<i>Marcar si forma parte de la llave primaria</i>	
<i>Para valores numéricos, el número de dígitos (incluyendo punto decimal), para otros campos el número de caracteres</i>						
<i>Las tablas y su estructura deberán derivarse del diagrama E-R que modela al sistema. Por lo general, cada entidad se convierte en una tabla y los atributos asociados forman parte de los campos que la componen. Sin embargo, en ocasiones es necesario incluir en algunas tablas otros campos que establezcan la relación con otra u otras, o bien, crear tablas que representen los vínculos.</i>						
<i>Adicionalmente deberá seguirse un proceso de normalización que permita garantizar un nivel apropiado de integridad y evitar la redundancia de información en la medida en que los requerimientos específicos de desempeño del sistema lo permitan.</i>						

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_









## REGISTRO DE ARCHIVOS DE ÍNDICES

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

NOMBRE DE LA TABLA			
NOMBRE DEL ARCHIVO DEL ÍNDICE	UNIQUE	ASCENDENTE O DESCENDENTE	DESCRIPCIÓN
↑ <i>Nombre del archivo en que se almacena el índice</i>	↑	↑	↑ <i>Nombre de la tabla a la que pertenece el índice</i>
	↑ <i>Indicar si el índice se genera con el atributo unique</i>		
		↑ <i>Marcar con una "A" si el índice es ascendente o con una "D" si es descendente</i>	↑ <i>El nombre del campo o la expresión mediante la que se genera el índice</i>

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_





## DICCIONARIO DE DATOS

ÁREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_  
PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

Descripción del formato del medio magnético en que se localiza el diccionario de datos

Etiqueta de identificación del medio magnético en que se localiza el diccionario de datos

TIPO DE MEDIO MAGNÉTICO \_\_\_\_\_  
ETIQUETA \_\_\_\_\_

NOMBRE DEL ARCHIVO

DESCRIPCIÓN

OBSERVACIONES

Deberá elaborarse un diccionario de datos que contenga al menos la información descrita en la estructura de las bases de datos y archivos.  
Esto implica que al menos deberán crearse las tablas de datos para:

- Tablas con los campos:  
(nombre de la tabla, nombre del campo, tipo de dato, longitud del campo, precisión, rango de valores, llave primaria, valor por omisión)
- Llaves secundarias, con los campos  
(nombre de la tabla, llave secundaria, tabla a la que se relaciona, cardinalidad)
- Índices con los campos  
(Nombre de la tabla, nombre del índice, único (s/n), ascendente o descendente, descripción)

Existen manejadores de bases de datos que incluyen un diccionario de datos con la información anterior y otra referente principalmente al manejo de la seguridad y los aspectos relacionados con la administración de la base de datos. En este caso, deben utilizarse y extraer de ellos la información requerida en este formato.

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DICCIONARIO DE DATOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

TIPO DE MEDIO MAGNETICO \_\_\_\_\_

ETIQUETA: \_\_\_\_\_

NOMBRE DEL ARCHIVO	DESCRIPCIÓN	OBSERVACIONES

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DESCRIPCIÓN DE PROCESOS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

NOMBRE DEL PROCESO:

INSUMOS DE ENTRADA	PRODUCTO	DESCRIPCIÓN DEL PROCESO	SUBPROCESOS UTILIZADOS	REFERENCIA DONDE SE DESCRIBE CADA SUBPROCESO
<p>Insumos o datos de entrada al proceso</p>	<p>Nombre del proceso que se describe</p>	<p>La salida o resultados que se obtienen del proceso</p>	<p>Si el proceso utiliza subprocesos, deberán listarse</p>	<p>Referencia donde se describe el subproceso, por ejemplo <b>DD05-SIP-05</b></p>
		<p>Descripción del algoritmo que se realiza dentro del proceso. En casos en que sea posible, deberá utilizarse pseudocódigo, de forma que se exprese claramente la funcionalidad del proceso</p>		
		<p>Cuando se trate de procesos de entrada/salida se hará referencia al o a los módulos involucrados (ver <b>DD06</b>)</p>		

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DESCRIPCIÓN DE PROCESOS

ÁREA: \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

NOMBRE DEL PROCESO:

INSUMOS DE ENTRADA	PRODUCTO	DESCRIPCIÓN DEL PROCESO	SUBPROCESOS UTILIZADOS	REFERENCIA DONDE SE DESCRIBE CADA SUBPROCESO

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



Nombre del módulo (pantalla o formato de captura, reporte,

## ENTRADAS Y SALIDAS

AREA \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

NOMBRE DEL MÓDULO		ENTRADA	SALIDA
TABLA PRINCIPAL ASOCIADA AL MÓDULO:			
OTRAS TABLAS RELACIONADAS:			
CAMPOS Y VARIABLES	OPERACIONES ASOCIADAS	FORMATO	VALORES VÁLIDOS
<p>Nombre de los campos y variables que aparecen en el módulo. Utilizar la notación <b>tabla.campo</b></p>	<p>Normalmente una pantalla o reporte se asocia a una tabla de la que depende principalmente. Si se tienen varias tablas principales, listarlas en el mismo espacio separadas por comas</p>	<p>Especificar claramente el formato de los datos</p>	<p>Si se trata de un módulo de entrada o salida de datos</p>
<p>Tablas asociadas con el módulo, por ejemplo: catálogos.</p>	<p>La operación asociada puede ser: - Edición (despliegue y modificación) - Despliegue - Cálculo Para el caso de variables que despliegan valores que se calculan (ejem. subtotales) especificar la forma como se calculan</p>		<p>Indicar el rango de valores válidos o reglas de validación asociadas a los datos</p>

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## ENTRADAS Y SALIDAS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

NOMBRE DEL MÓDULO: \_\_\_\_\_ ENTRADA SALIDA

TABLA PRINCIPAL ASOCIADA AL MÓDULO:

OTRAS TABLAS RELACIONADAS:

CAMPOS Y VARIABLES	OPERACIONES ASOCIADAS	FORMATO	VALORES VÁLIDOS

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

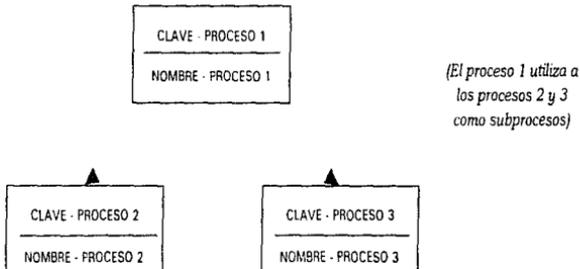


## DIAGRAMA ESTRUCTURAL DEL SISTEMA

ÁRFA \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE: \_\_\_\_\_

*El diagrama estructural deberá construirse utilizando rectángulos interconectados por flechas, como se muestra en la siguiente figura:*



*El diagrama deberá contener la estructura completa de procesos y subprocessos, así como de los módulos de entrada/salida que se describen en los formatos, **DD05** y **DD06** respectivamente, organizados en forma jerárquica. Las claves de referencia, los nombres de procesos y módulos deberán corresponder con los de dichos formatos.*

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

**DIAGRAMA ESTRUCTURAL DEL SISTEMA**

ÁRFA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## DESCRIPCIÓN DE PROGRAMAS Y RUTINAS

ÁREA \_\_\_\_\_ *Valor de retorno, ver descripción de los parámetros* \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_  
PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

NOMBRE ELEMENTO	PROG / RUTINA	NOMBRE ARCHIVO	Parámetros de entrada			Valor de retorno			CLAVE PROCESO	Nº DE LINEAS CÓDIGO	DESCRIPCIÓN
			Nombre	Tipo	Desc	Nombre	Tipo	Desc			
<i>Nombre del elemento que se describe</i>	<i>Indicar si se trata de una rutina o programa</i>	<i>Nombre del archivo en que se almacena el elemento</i>							<i>Clave de referencia del proceso que se codifica, de acuerdo con los formatos DD05. En caso de rutinas auxiliares (por ejemplo para despliegue gráfico), se recomienda utilizar claves para hacer referencia a rutinas relacionadas</i>		
		<i>Nombre del parámetro de entrada</i>				<i>Nombre del parámetro de retorno</i>				<i>Número de líneas de código del programa o rutina</i>	
		<i>Tipo de dato (carácter, numérico, etc.)</i>									<i>Breve descripción del objetivo del programa o rutina</i>
		<i>Breve descripción del parámetro</i>									

ELABORÓ: \_\_\_\_\_ AUTORIZÓ: \_\_\_\_\_  
FECHA: \_\_\_\_\_ FECHA: \_\_\_\_\_



## DESCRIPCIÓN DE PROGRAMAS Y RUTINAS

ÁREA \_\_\_\_\_ CVE PROYECTO: \_\_\_\_\_ CVE REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

NOMBRE ELEMENTO	PROG./ RUTINA	NOMBRE ARCHIVO	Parámetros de entrada			Valor de retorno			CLAVE PROCESO:	N° DE LINEAS CÓDIGO	DESCRIPCION
			Nombre	Tipo	Desc	Nombre	Tipo	Desc			

ELABORÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_



## RESULTADO DE PRUEBAS

AREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_  
PROYECTO: \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA: \_\_\_\_\_ DE: \_\_\_\_\_

### DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS

*En esta sección deberá hacerse una descripción completa de las pruebas realizadas, incluyendo la descripción de los procedimientos efectuados, los datos de prueba utilizados, etc.*

### DESCRIPCIÓN DE RESULTADOS

*En esta sección deberá hacerse una narración de los resultados de las pruebas, incluyendo observaciones detalladas sobre los posibles errores detectados, así como un diagnóstico de éstos*

*Deberá indicarse si las pruebas fueron satisfactorias, o es necesario repetirlas para corregir errores*

*Si existen errores, deberá programarse una siguiente sesión de pruebas, anotando la fecha en este campo*

- NECESIDAD DE NUEVAS PRUEBAS  
 PRUEBAS SATISFACTORIAS

FECHA PARA LAS SIGUIENTES PRUEBAS \_\_\_\_\_

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## RESULTADO DE PRUEBAS

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_

PROYECTO: \_\_\_\_\_ ABREV. \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS

DESCRIPCIÓN DE RESULTADOS

- NECESIDAD DE NUEVAS PRUEBAS  
 PRUEBAS SATISFATORIAS

FECHA PARA LAS SIGUIENTES PRUEBAS \_\_\_\_\_

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_



## INTEGRACIÓN AL BANCO DE SISTEMAS INSTITUCIONALES

ÁREA: \_\_\_\_\_ CVE PROYECTO \_\_\_\_\_ CVE REFERENCIA: \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE: \_\_\_\_\_

### RESPONSABLE DEL SISTEMA

NOMBRE \_\_\_\_\_ TELEFONO \_\_\_\_\_  
PUESTO \_\_\_\_\_ FAX \_\_\_\_\_  
DIRECCIÓN \_\_\_\_\_

### DESCRIPCIÓN DEL SISTEMA

*En esta sección deberá hacerse una breve descripción del sistema, incluyendo su objetivo, así como sus principales características técnicas.*

*Deberá proporcionarse la información que permita localizar al responsable del sistema.*

*Deben señalarse los elementos que se entregarán al banco de sistemas de la empresa o institución*

*Deben incluirse el nombre y firma de la persona que entrega el sistema y de la que lo recibe por parte del banco de sistemas de la empresa o institución.*

### ELEMENTOS ENTREGADOS

- PROGRAMAS FUENTE
- PROGRAMAS EJECUTABLES
- MANUAL TÉCNICO
- MANUAL DEL USUARIO

#### FORMAS

- DA03
- DADMF04
- DADMF05
- DADME06
- DADME07
- DADMD08
- DADMD09
- DD01

- DD02
- DD03A
- DD03B
- DD03C
- DD04
- DD05
- DD06
- DD07
- DP01

OTROS \_\_\_\_\_

ENTREGÓ: \_\_\_\_\_

FIRMA: \_\_\_\_\_

RECIBIÓ: \_\_\_\_\_

FIRMA: \_\_\_\_\_



## INTEGRACIÓN AL BANCO DE SISTEMAS INSTITUCIONALES

ÁREA: \_\_\_\_\_ CVE. PROYECTO: \_\_\_\_\_ CVE. REFERENCIA: \_\_\_\_\_  
 PROYECTO: \_\_\_\_\_ ABREV: \_\_\_\_\_ HOJA \_\_\_\_\_ DE: \_\_\_\_\_

### RESPONSABLE DEL SISTEMA

NOMBRE: \_\_\_\_\_ TELÉFONO: \_\_\_\_\_  
 PUESTO: \_\_\_\_\_ FAX: \_\_\_\_\_  
 DIRECCIÓN: \_\_\_\_\_

### DESCRIPCIÓN DEL SISTEMA

### ELEMENTOS ENTREGADOS

- |  |                                  |                                  |                                |                               |                               |
|--|----------------------------------|----------------------------------|--------------------------------|-------------------------------|-------------------------------|
| <input type="checkbox"/> PROGRAMAS FUENTE      | <input type="checkbox"/> DA03    | <input type="checkbox"/> DADME07 | <input type="checkbox"/> DD02  | <input type="checkbox"/> DD04 | <input type="checkbox"/> DP01 |
| <input type="checkbox"/> PROGRAMAS EJECUTABLES | <input type="checkbox"/> DADMF04 | <input type="checkbox"/> DADMD08 | <input type="checkbox"/> DD03A | <input type="checkbox"/> DD05 |                               |
| <input type="checkbox"/> MANUAL TÉCNICO        | <input type="checkbox"/> DADMF05 | <input type="checkbox"/> DADMD09 | <input type="checkbox"/> DD03B | <input type="checkbox"/> DD06 |                               |
| <input type="checkbox"/> MANUAL DEL USUARIO    | <input type="checkbox"/> DADME06 | <input type="checkbox"/> DD01    | <input type="checkbox"/> DD03C | <input type="checkbox"/> DD07 |                               |

OTROS: \_\_\_\_\_

ENTREGÓ: \_\_\_\_\_

FIRMA: \_\_\_\_\_

RECIBÍÓ: \_\_\_\_\_

FIRMA: \_\_\_\_\_

FORMA/DI02



## SOLICITUD DE MANTENIMIENTO

ÁREA: \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE. REFERENCIA \_\_\_\_\_  
PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

CORRECCIÓN

MODIFICACIÓN

DESCRIPCIÓN DEL PROBLEMA

*Debe indicarse si la solicitud del mantenimiento es para **corregir** un error detectado durante la operación del sistema, o para **modificarlo** con el fin de adaptarlo a nuevas necesidades del área usuaria.*

*En esta área deberá darse una descripción detallada del problema que origina la solicitud de mantenimiento, incluyendo mensajes de error, ejemplos de los errores, etc.*

SOLUCIÓN PROPUESTA (SOLO PARA MODIFICACIONES)

*En los casos en que la operación del área usuaria se haya alterado y se requiera modificar el sistema para adaptarlo a las nuevas necesidades, deberá describirse en forma detallada cuáles son los cambios requeridos (por ejemplo nuevos datos que deben considerarse, nuevos reportes, etc.)*

*Debe registrarse el nombre de la persona que realizó la solicitud de mantenimiento y la fecha en que la presentó al área de sistemas*

*Debe registrarse el nombre de la persona que atendió la solicitud de mantenimiento y la fecha en que dio respuesta al área usuaria*

REALIZÓ LA SOLICITUD

NOMBRE: \_\_\_\_\_

FECHA: \_\_\_\_\_

ATENDIÓ LA SOLICITUD

NOMBRE: \_\_\_\_\_

FECHA: \_\_\_\_\_

 <b>SOLICITUD DE MANTENIMIENTO</b>	
ÁREA: _____ CVE. PROYECTO: _____ CVE REFERENCIA: _____ PROYECTO: _____ ABREV.: _____ HOJA _____ DE _____	
<input type="checkbox"/> CORRECCIÓN	<input type="checkbox"/> MODIFICACIÓN
DESCRIPCIÓN DEL PROBLEMA	
SOLUCIÓN PROPUESTA (SÓLO PARA MODIFICACIONES):	
REALIZÓ LA SOLICITUD NOMBRE: _____ FECHA: _____	ATENDIÓ LA SOLICITUD NOMBRE: _____ FECHA: _____



## RESULTADO DE SOLICITUD

ÁREA \_\_\_\_\_ CVE. PROYECTO \_\_\_\_\_ CVE. REFERENCIA \_\_\_\_\_

PROYECTO \_\_\_\_\_ ABREV.: \_\_\_\_\_ HOJA \_\_\_\_\_ DE \_\_\_\_\_

CLAVE DE REFERENCIA DE LA SOLICITUD DE MANTENIMIENTO \_\_\_\_\_

ACCIONES EFECTUADAS

*Debe incluirse la clave de referencia de la solicitud de mantenimiento, de la que se están presentando los resultados.*

*En esta sección debe hacerse una descripción detallada de las acciones efectuadas para resolver la problemática presentada en la solicitud de mantenimiento.*

*Debe señalarse si el mantenimiento al que se refiere esta forma quedó concluido o es necesario tomar otras acciones.*

SITUACIÓN

CONCLUIDO

PENDIENTE

ELABORÓ: \_\_\_\_\_

AUTORIZÓ: \_\_\_\_\_

FECHA: \_\_\_\_\_

FECHA: \_\_\_\_\_

<b>RESULTADO DE SOLICITUD</b>	
	ÁREA: _____ CVE PROYECTO: _____ CVE REFERENCIA: _____ PROYECTO: _____ ABREV.: _____ HOJA _____ DE: _____
CLAVE DE REFERENCIA DE LA SOLICITUD DE MANTENIMIENTO: _____	
ACCIONES EFECTUADAS	
SITUACIÓN	
<input type="checkbox"/> CONCLUIDO	<input type="checkbox"/> PENDIENTE
ELABORÓ: _____ FECHA: _____	AUTORIZÓ: _____ FECHA: _____

## CONCLUSIONES

*No existe nada más difícil de emprender,  
más peligroso de dirigir, o más incierto en su  
éxito, que encabezar la introducción de un  
nuevo orden en las cosas.*

*Nicolás Maquiavelo*



## CONCLUSIONES

A finales de la década de los ochentas y principios de los noventas hemos visto una rápida evolución en lo que respecta al desarrollo y principalmente al uso de los sistemas de información en cualquier institución o empresa. Actualmente se observa la rapidez con que toda institución esta considerando potencialmente la tecnología de cómputo como uno de sus recursos principales, especialmente para contar con sistemas de información que ayuden a mejorar el desempeño de las actividades, tanto rutinarias como de toma de decisiones; pero, lo que hace la diferencia entre una institución y otra es la suficiencia y oportunidad del manejo de dicha información.

Con la concepción de las políticas, normas y procedimientos efectuadas en este trabajo de tesis se contribuye a que cualquier institución o empresa pública o privada establezca un proceso inicial para la elaboración de sistemas propios; además permitirá que las organizaciones cuenten con una guía a seguir para desarrollar, controlar y regir las acciones y actividades que redundarán en la obtención de un sistema de información. Así mismo al establecer los criterios mínimos para implantar un proceso de elaboración de sistemas se tendrá como finalidad la mejora de la calidad, esta mejora se logrará en la forma de operar de la empresa, más ordenada y apegada a los procedimientos. Con el establecimiento de las políticas, normas y procedimientos se pretende incrementar la confianza en la elaboración de sistemas a través del aseguramiento de su calidad, con ello se mantendrá un nivel de consistencia en cualquier desarrollo de sistemas y lo mejor aun se tendrá un parámetro para aprender de experiencias pasadas.

Sin embargo, su importancia también radica en realizar y obtener una documentación concisa, completa y específica a lo largo de todas las fases del ciclo de vida para el desarrollo de sistemas, que sirvan principalmente como medio para efectuar el mantenimiento respectivo a cualquier sistema. Es importante mencionar que la aplicación de este documento exige una clara responsabilidad de aceptación de un personal éticamente consiente que siga los parámetros aquí expuestos

A través de la realización de este trabajo advertimos la importancia del uso de una metodología para la elaboración de cualquier producto y principalmente en el desarrollo del software. Además con el uso de una metodología se tendrá control, organización, seguridad y definición en la elaboración de cualquier sistema, así mismo cualquier sistema tendrá una adecuada continuidad ya que invariablemente todo sistema necesitará de mantenimiento para cumplir satisfactoriamente los requerimientos de los usuarios.

En lo que se refiere al uso de la orientación a objetos se vislumbra que sin duda se consolidará como una forma de elaborar los sistemas de cualquier institución o empresa de manera más rápida, principalmente debido a la forma en que se está comportando el mercado de la industria del cómputo. Se observa que entre las aportaciones que proporciona es la creación de estructuras reutilizables (clases) con atributos y comportamientos específicos, y con el establecimiento de librerías de clases. Sin embargo, es importante mencionar que cualquier institución o empresa que desee utilizar alguna metodología con orientación a objetos deberá contar con una buena planeación del sistema, con ello un proceso establecido para la elaboración de sistemas y con personal ordenado y dispuesto a concebir los sistemas de modo distinto. Estamos conscientes que la tecnología del cómputo no es estática, por lo que debemos estar preparados y con una amplia perspectiva en las tendencias de programación actuales, y de manera particular en lo concerniente a objetos, ya que de no tomarse en cuenta, en poco tiempo estaríamos enfrentando el desarrollo de sistemas con herramientas obsoletas.

Es obligación mencionar que el establecer políticas, normas y procedimientos en cualquier ámbito de una institución o empresa es un proceso que involucra trabajo, interés, decisión, dedicación, orden, disciplina y principalmente mejora continua tanto de la dirección como del personal en general que constituye la institución o empresa.

Deseamos también subrayar que las consideraciones expuestas en el presente trabajo en cuanto a organización, calidad y tecnología son con la perspectiva de aplicarse de corto a mediano plazo. Éste fue elaborado pensando en las tendencias de programación y calidad que se perfilan para estar vigentes en la próxima década, así mismo se contempla que conforme pasen los años los estándares de aseguramiento de calidad para el desarrollo de software seguirán evolucionando.

Desde nuestro punto de vista consideramos que la implantación de políticas, normas y procedimientos para el desarrollo de sistemas con un aseguramiento de calidad, es un paso que tarde o temprano se tendrá que dar en cualquier institución o empresa.

Por último, estamos seguros que esta tesis es un documento que será muy apreciado por los diseñadores de sistemas de cualquier empresa o institución y que será de gran ayuda para los Ingenieros en Computación.



## **GLOSARIO DE TÉRMINOS**



## GLOSARIO DE TÉRMINOS

TÉRMINO	SIGNIFICADO
<b>Abstracción</b>	Facilidad mental que permite a los humanos ver los problemas del mundo real con grados variables de detalle, dependiendo del contexto vigente del problema.
<b>Administración de la calidad</b>	Conjunto de actividades de la función general de administración que determina la política de calidad, los objetivos, las responsabilidades y la implantación de éstos por medios tales como: <i>planeación de la calidad, control de calidad, aseguramiento de la calidad y mejoramiento de la calidad</i> dentro del marco del sistema de calidad.
<b>Aseguramiento de calidad</b>	Conjunto de actividades planeadas y sistemáticas implantadas dentro del sistema de calidad, demostradas según se requiera para proporcionar confianza adecuada de que un elemento cumplirá los requisitos para la calidad.
<b>Base de datos</b>	Colección de información, organizada y presentada para servir a un propósito específico, como la facilitación de búsquedas, ordenamientos o procesamiento de los datos.
<b>Campo</b>	Elemento de información contenido dentro de un renglón o registro. Equivalente lógico de una columna.
<b>Clase</b>	Es una plantilla que define los métodos y los datos que componen a un tipo de objeto. Todos los objetos de una misma clase son idénticos en forma y en comportamiento, pero cada uno contiene sus datos individuales que representan su estado.
<b>Clase Abstracta</b>	Una clase abstracta que no tiene instancias se concibe con el propósito de crear una clase base para la definición común que afectará a las clases que deriven de ella en la jerarquía (de clases). El término "clase virtual" es el mismo concepto.

TÉRMINO	SIGNIFICADO
<b>Columna</b>	Conjunto de todos los renglones de una tabla que tienen un atributo común. Contiene un dato individual dentro de cada renglón o registro.
<b>Comité de informática</b>	Grupo de personas pertenecientes a una organización, dedicadas a definir y establecer las directrices de la informática para la organización.
<b>Control de calidad</b>	Técnicas y actividades de carácter operacional, utilizadas para cumplir los requisitos para la calidad.
<b>Dato</b>	Representación codificada de información para usarla en una computadora. Los datos tienen atributos como tipo y longitud.
<b>Dependencia funcional</b>	El atributo A de una relación es funcionalmente dependiente del atributo B, si el valor de A está determinado por el valor de B.
<b>Diccionario de datos</b>	Descripción de los elementos de una base de datos y como están estructurados.
<b>Dictamen técnico</b>	Juicio o resolución sobre recursos informáticos requeridos por las diferentes áreas de la institución o empresa; emitido por la dirección de informática.
<b>Documentación</b>	Especificaciones técnicas, manuales de instalación y del usuario, que contengan la descripción, el diseño, el código y los comentarios que hacen a un sistema accesible, comprensible y fácilmente modificable.
<b>Dominio</b>	El dominio de un campo es el rango de valores continuos o discretos permitidos para el campo.

TÉRMINO	SIGNIFICADO
<b>Elemento</b>	Sinónimo de campo en una tabla. Intersección de un renglón y una columna.
<b>Equipo de trabajo</b>	Conjunto de personas asignadas para la elaboración de un proyecto o de un sistema.
<b>Especialización</b>	Creación de subclases a partir de superclases, donde estas resultan redefinidas.
<b>Estándares</b>	Especificaciones técnicas u otros criterios precisos para ser utilizados por los integrantes de una organización tales como normas, guías o definiciones de características, para asegurar que un sistema cumple con el propósito para el que fue creado. Los estándares internacionales contribuyen a simplificar el desarrollo de sistemas e incrementan su integridad y efectividad.
<b>Estudio de factibilidad tecnológica y económica</b>	Documento formal para evaluar la problemática que gira en torno a un sistema. Su objetivo es presentar alternativas de solución para la elaboración de un sistema, que justifique el aspecto económico con base en una buena relación costo beneficio.
<b>Generalización</b>	Interrelación una clase y una o más versiones redefinidas o especializadas de ella, es decir, la superclase generaliza a las subclases.
<b>Hardware</b>	Toda la maquinaria y el equipamiento (monitor, unidades de disco, teclado, tarjetas de red, etc.), incluyendo los periféricos y componentes electrónicos. Contrástese con software, el cual es un conjunto de instrucciones que le dicen a la computadora qué hacer.

TÉRMINO	SIGNIFICADO
<b>Herencia</b>	Es un mecanismo mediante el cual una clase puede hacer uso de los datos y de los métodos definidos en todas las clases que se encuentran arriba de ella, en una jerarquía de clases.
<b>Herencia múltiple</b>	Cuando en una jerarquía de clases, una clase puede tener dos o más superclases.
<b>Índice</b>	Conjunto de apuntadores ordenados lógicamente por los valores de una llave. Un índice es un elemento de la base de datos que proporciona acceso a los registros de una tabla, mediante el valor de una llave.
<b>Informática</b>	Se concibe como una herramienta de cambio, como una palanca de modernización y como agente de innovación. Es el resultado de la convergencia tecnológica peculiar, que se ha producido a lo largo de ya casi medio siglo, entre las telecomunicaciones, la ciencias de la computación, la microelectrónica, y ciertas ideas de administración y manejo de información.
<b>Instancia</b>	Termino que se utiliza para referirse a un objeto que pertenece a una clase dada. Por ejemplo. Jalisco en una instancia de la clase Estado.
<b>Jerarquía de clases</b>	Una estructura de árbol que representan las relaciones existentes entre varias clases. Las jerarquías representan una especialización de clases, de más general representada por su raíz, a más particular representada por sus hojas. Puede haber cualquier número de niveles, y cualquier número de clases en cada nivel.
<b>Llave</b>	Uno o más campos usados para identificar un registro, frecuentemente se utiliza como índice de una tabla.

TÉRMINO	SIGNIFICADO
<b>Llave foránea</b>	Columna o combinación de columnas cuyos valores se relacionan con la llave primaria de alguna otra tabla. Una llave foránea no tiene que ser única. No deben existir valores de las llaves foráneas, excepto "nulo", a menos que el mismo valor exista en una llave primaria.
<b>Llave primaria</b>	Columna o combinación de columnas que identifican de manera única una tabla. Siempre deben ser diferentes de "nulo" y tener un índice único. Una llave primaria se usa para relacionarse con llaves foráneas en otras tablas.
<b>Manual de instalación</b>	Documento que presenta los procedimientos paso a paso para poner en operación un sistema.
<b>Manual de procedimientos</b>	Documento que contiene los lineamientos o guías a seguir para orientar y normar las actividades por realizar en el análisis, diseño, desarrollo y mantenimiento de sistemas en una organización.
<b>Manual de operación</b>	Documento que contiene el manual de instalación y el manual del usuario.
<b>Manual del usuario</b>	Documento que sirve como guía para la utilización de los módulos, menús, pantallas, registros y campos que conforman un sistema.
<b>Manual técnico</b>	Documento que contiene las especificaciones técnicas de un sistema; tales como: diagrama general del sistema, diagrama de flujo de datos, diagrama de entidad-relación, diagrama de estructura de datos, diccionario de datos, programas fuentes y archivos de programas ejecutables.

TÉRMINO	SIGNIFICADO
<b>Mejoramiento de la calidad</b>	Aquellas acciones tomadas en la organización para incrementar la efectividad y la eficiencia de las actividades o procesos. Su fin es proveer de beneficios adicionales a la organización y a sus clientes.
<b>Mensaje</b>	Es una señal de excitación de un objeto hacia otro, que le indica al objeto receptor que ejecute uno de sus métodos.
<b>Método</b>	Es un procedimiento dentro de un objeto. Los métodos de un objeto forman la interface de éste con otros objetos. Los datos internos del objeto sólo pueden ser cambiados mediante sus métodos.
<b>Módulo</b>	Unidad que forma parte de un sistema; en cuanto a su operación es independiente de otros módulos, pero en cuanto a su utilización puede requerir datos o información proporcionada por otros módulos del sistema que lo contiene.
<b>Nulo</b>	Que no tiene un valor explícito asociado. No es equivalente de cero o blanco. Normalmente se utiliza para representar la ausencia de información.
<b>Objeto</b>	Es una estructura que representa a un elemento de la realidad o un concepto teórico generalizado. Un objeto exhibe cierto estado (representado por sus datos internos) y cierto comportamiento (representado por sus métodos).
<b>Planeación de la calidad</b>	Actividades que determinan los objetivos y requisitos para la calidad. Esos requisitos incluyen los elementos para la implantación del sistema de calidad.
<b>Primera forma normal</b>	Una relación está en la primera forma normal si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.

TÉRMINO	SIGNIFICADO
<b>Producto de software</b>	Conjunto completo de programas para computadora, así como los procedimientos y la documentación asociada, para ponerse a disposición del usuario.
<b>Programa</b>	Conjunto de instrucciones que indican a la computadora cómo llevar a cabo una acción específica.
<b>Registro</b>	Grupo de campos (columnas) cuya información se trata como una unidad. Equivalente lógico de un renglón.
<b>Relación</b>	Dada una serie de conjuntos $D_1, D_2, \dots, D_n$ (no necesariamente distintos) se dice que $R$ es una relación sobre estos $n$ conjuntos si es un conjunto de $n$ tuplas ordenadas $\langle d_1, d_2, \dots, d_n \rangle$ , tales que $d_1$ pertenece a $D_1$ , $d_2$ pertenece a $D_2$ , ..., $d_n$ pertenece a $D_n$ . Los conjuntos $D_1, D_2, \dots, D_n$ son los dominios de $R$ . El valor $n$ es el grado de $R$ .
<b>Renglón</b>	Equivalente lógico de un registro.
<b>Segunda forma normal</b>	Una relación está en segunda forma normal si está en primera forma normal y cada atributo que no forma parte de la llave principal está en forma total y funcionalmente dependiente de ella.
<b>Sistema de calidad</b>	Es la estructura organizacional, los procedimientos, los procesos y los recursos necesarios para implantar la administración de calidad. De hecho está diseñado para la administración interna de la empresa o institución.
<b>Sistemas institucionales</b>	Son aquellos sistemas que engloban datos o información común a las diferentes áreas de la institución.

TÉRMINO	SIGNIFICADO
<b>Software</b>	Instrucciones para una computadora que realizan una tarea en particular.
<b>Subclase</b>	Una clase que es la especialización de otra. Una clase que tiene una clase definida arriba de ella en una jerarquía de clases. Por ejemplo, un León es una especialización (subclase) de Felino.
<b>Superclase</b>	Es una clase que se encuentra más arriba en la jerarquía, con respecto a otra clase. Una superclase es más general que sus subclases. Por ejemplo, transporte es la superclase automóvil.
<b>Tabla</b>	Colección de renglones (o registros) que tienen columnas (o campos) asociadas.
<b>Tercera forma normal</b>	Una relación está en tercera forma normal si está en segunda forma normal y ningún atributo involucrado en la relación es funcionalmente dependiente de algún otro atributo que no es parte de la llave.
<b>Tipo de dato</b>	Identificador que especifica la clase de información (números, valores lógicos, caracteres, texto, etc.) que contiene una columna y cómo será almacenada.
<b>Valor por omisión</b>	Valor que se asigna en forma automática a un campo, aunque puede ser modificado posteriormente.
<b>Variable</b>	Estructura de almacenamiento temporal que recibe un nombre único asignado por el programador.

## **APÉNDICE A**



## **METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS CON ORIENTACION A OBJETOS**

### **DOCUMENTACIÓN**

#### **1. La necesidad de los diagramas**

Los diagramas o representaciones gráficas describen una parte fundamental en el desarrollo de sistemas. En general, se utilizan para definir las especificaciones de un programa y representar el diseño de los sistemas (programas). Proporcionan el proyecto original para la implantación del diseño en código y además son partes esenciales para la documentación del sistema y posteriormente para su mantenimiento.

Los diagramas claros son una parte fundamental en el diseño de sistemas y en el desarrollo de programas durante el *ciclo de vida del software*. Una técnica de diagramación pobre puede provocar la inhibición en el razonamiento, en cambio, una buena puede acelerar el trabajo y aumentar la calidad de los resultados. Por ejemplo, si sólo hay una persona que desarrolla el diseño de un sistema o de un programa, los diagramas ayudarán al razonamiento claro. Cuando hay varias personas que participan en el desarrollo del sistema los diagramas son una poderosa herramienta esencial para la comunicación y el desarrollo del sistema.

Con una técnica de diagramación formal, los desarrolladores tendrán la posibilidad de intercambiar ideas y podrán reunir con precisión los distintos componentes del sistema. Una de las características con que deben contar los diagramas es una simbología estandarizada, a fin de facilitar la comunicación y evitar las confusiones que puedan conducir a un error dentro del desarrollo del sistema por parte del equipo de trabajo.

En cuanto a la fase de mantenimiento en el ciclo de vida del software, los diagramas son herramientas valiosas para saber por parte del equipo de programación cómo funciona el sistema, además de que permite localizar el origen de los errores y el impacto de las posibles modificaciones que se hagan al sistema. Como se sabe, después de un periodo transcurrido esto es lo que va desgastando al software.

En términos generales, podemos decir que los diagramas forman un lenguaje que permite el modelado del software, debido a que proporcionan una forma de representación concisa y sin ambigüedades. Además, forman una parte fundamental en el ciclo de vida del software; más específicamente en la etapa de documentación del sistema, sin olvidar que son esenciales para el análisis y el diseño de software.

## **2. Requisitos de documentación**

Un sistema informático tendrá una buena documentación si incluye los programas, así una documentación debe cumplir con los siguientes puntos:

- ◆ Proporciona una visión de alto nivel del sistema (programa), explicando su propósito y las relaciones entre los diversos componentes.
- ◆ Mejora la comprensión de los sistemas y, en general, del software producido.
- ◆ Describe detalladamente los componentes de los datos y procedimientos que forman el sistema.
- ◆ Es fácil y poco costosa de realizar y actualizar, y existe la posibilidad de producirse automáticamente.
- ◆ Proporciona una forma de representar los requerimientos de diseño de un sistema y del traslado del diseño al código de programa.

### **3. Importancia de la visión en los diagramas**

Las técnicas de diagramación se utilizan para dar una visión de alto nivel y detallada de un sistema. Una de las cualidades es que permiten la localización exacta de un error fácilmente; si una persona deseara determinar en cuál de varios programas se realiza una determinada función, una visión de alto nivel será la mejor ayuda.

Para representar un sistema completo se necesita un diagrama de estructura de datos de alto nivel y otro detallado del sistema. La visión del sistema identifica los componentes del proceso y las relaciones entre ellos mostrando el flujo de datos, el flujo de control, así como el tiempo y la secuencia de la información. La visión de la estructura de datos describe principalmente las entidades y atributos, así como sus posibles relaciones o asociaciones. La importancia de ambas visiones radica en que los ingenieros pueden ver cómo los datos se derivan o utilizan en un determinado proceso.

### **4. Tipos de metodologías**

Los tres tipos de metodologías explicadas en el presente apéndice muestran algunos aspectos en común; entre las características podemos mencionar que toda metodología orientada a objetos:

- ◆ Tiene como unidad básica los objetos; un objeto es una entidad que incluye atributos y operaciones que son soportados por el propio objeto.
- ◆ Modela aspectos dinámicos de los objetos o clases, para ello hace uso de diagramas de transición de estados.
- ◆ Utiliza una notación para definir interacciones entre objetos y jerarquía de clases, mostrando las relaciones de herencia.

#### **4.1. Metodología de diseño orientado a objetos (Grady Booch)**

Esta metodología se basa en un principio de estructuramiento, favoreciendo a la descomposición orientada a objetos sobre la descomposición tradicional. El diseño orientado a objetos se enfoca de sobre manera en los resultados del diseño, más que en el proceso de análisis. De hecho, hace énfasis en la necesidad de producir múltiples vistas (distintos modelos) de la fase de diseño, el fin a perseguir es capturar todas las decisiones de diseño posibles a realizar. Finalmente el producto obtenido en el diseño orientado a objetos comprende tanto un modelo estático como un modelo dinámico.

Durante la fase de análisis la metodología de Booch prescribe la creación de una vista lógica del modelado de objetos, que represente las clases y la estructura de objetos, y las relaciones entre ellos, sin considerar ninguna restricción de codificación. En la fase de diseño el modelo es refinado para comprender las consideraciones pertinentes de la codificación. El resultado que se obtiene de la vista física del modelo es la captura de las decisiones de diseño para la codificación. Cabe señalar que la metodología de diseño orientado a objetos (OOD) asume que durante la aplicación del ciclo de vida de desarrollo del sistema, la fase de análisis y diseño son realizadas iterativamente e incrementalmente.

La metodología de diseño orientado a objetos describe cuatro pasos fundamentales a realizar durante el proceso de modelado, no existe ninguna restricción para la realización de estos pasos en un orden determinado.

- *Identificar clases y objetos en un nivel de abstracción dado.* El análisis del dominio del problema es entendido para buscar las claves de las abstracciones (clases). Como parte del análisis del dominio del problema las clases son buscadas, construidas y documentadas respectivamente.
- *Identificar las semánticas de clases y objetos.* Las responsabilidades de las clases son determinadas en un proceso iterativo. Las responsabilidades serán especificadas para cada objeto de la aplicación,

describiendo su ciclo de vida desde la creación hasta la destrucción, incluyendo sus características de comportamiento. En este paso los productos resultantes son clases refinadas bosquejadas del punto anterior.

- **Identificar las relaciones entre clases y objetos.** Este punto se requiere para el establecimiento de las interacciones entre clases y para identificar relaciones de herencia. Existen dos actividades de ayuda que refinan el diseño previamente obtenido, el descubrimiento de modelos y la determinación de decisiones. El diseño orientado a objetos identifica dos tipos de modelos: el modelado entre clases, que ayuda a reconocer y a simplificar la estructura del modelado de clases; y el modelado entre colecciones de objetos, que conduce a la generalización de mecanismos ya involucrados en el diseño. Las decisiones determinan cómo las clases (objetos) deberían verse con respecto a otras, esto afecta el diseño de la arquitectura del sistema. En este paso Grady Booch sugiere el uso de la técnica de *colaboración responsabilidad de clases* (CRC).
- **Codificación de clases y objetos.** Las decisiones de diseño son realizadas en este paso, incluye la distribución de módulos y procesos.

La metodología de diseño considerada por Booh incluye en su notación las siguientes actividades con sus concementes diagramas:

- ◆ Estructura lógica
  - Diagramas de clases
  - Diagramas de objetos
- ◆ Estructura física
  - Diagramas de módulos
  - Diagramas de procesos
- ◆ Dinámica de clases
  - Diagramas de transición de objetos
- ◆ Dinámica de instancias
  - Diagramas temporales

Los primeros pasos se dedican a aspectos estáticos del sistema de carácter lógico y físico. En cuanto a lo dinámico se trata usando una serie de técnicas de diagramas temporales y transición de estados.

Esta metodología permite que el diseñador parta de la técnica lingüística estilo Aboott, así como de las técnicas de análisis estructurado como precursores admisibles para iniciar un diseño con orientación a objetos (bajo el riesgo de caer en la concepción mental del diseño estructurado).

A grandes rasgos su notación consiste de relaciones entre manchas que simbolizan clases, las cuales se conectan con líneas de distintos tipos con el fin de representar algunas de las características como son la herencia, la utilización, y otras relaciones. Para los objetos distingue tres papeles: actores (objetos que pueden iniciar el comportamiento de otros objetos, no permiten actuar sobre ellos), servidores (sólo pueden ser aceptados por otros objetos) y agentes (que hacen ambas cosas).

Cada clase en el diseño de Booch describe el uso de una plantilla estándar que contiene su identidad, atributos, métodos, y la siguiente información adicional:

- Documentación
- Visibilidad (exportado/privado/importado)
- Multiplicidad
- Superclases
- Realización de la interfaz (implantación) (pública/protégida/privada)
- Diagramas de transición de estados
- Concurrencia(secuencial/por bloques/activa)
- Persistencia (estática/dinámica)
- Complejidad del espacio(texto)

Los diagramas lógicos y las plantillas anteriormente descritas conforman la descripción lógica-física del modelado. Mientras que la descripción dinámica es conseguida mediante diagramas de transición de estados que son de mejor aplicación entre más pequeño sea el sistema.

La metodología de diseño orientado a objetos de Booch ha tenido un seguimiento y evolución en su desarrollo, posee una estructura general, considera relaciones cliente/servidor, de prioridad de uso, relaciones de contención, y lo que le denomina el "diseño global de ida y vuelta"; esto es, un proceso de diseños incrementales conformado en distintos niveles de abstracción y razonamiento, pueden desarrollarse preferentemente de la siguiente manera:

- Identificación de clases e instancias
- Definición de su semántica
- Búsqueda de relaciones entre ellas
- Implementar el diseño como prototipo
- Examen del sistema a efectos de consistencia y cohesión
- Redefinición de las clases, instancias, semántica y estructura, basándose en lo aprendido.

Este proceso tiene su fin cuando se cree que todas las funciones y abstracciones claves han sido definidas y es rigurosamente no lineal.

#### 4.2. Técnica de modelado de objetos (James Rumbaugh)

Esta técnica es una de las metodologías de análisis y diseño con orientación a objetos más madura, completa y eficiente que existe en la actualidad. Se caracteriza por ser abierta (no propietaria), que le permite ser de dominio público y en consecuencia se acopla a las necesidades actuales y futuras de la ingeniería de software. Cabe señalar que la OMT (*Técnica de Modelado de Objetos*) está soportada por numerosas herramientas CASE y ha sido adoptada por numerosas organizaciones de América y Europa; además es una de las metodologías clásicas que se estudian en cualquier curso de orientación a objetos. La OMT está constituida por las fases de análisis y diseño, que en conjunto con la abstracción que ofrece el enfoque orientado a objetos y mediante un proceso iterativo, muestra la importancia de las primeras fases en la elaboración de sistemas. A continuación se describirá brevemente cada una de estas fases con sus respectivas actividades.

##### **Análisis**

Durante esta fase se realiza un entendimiento del problema y se modela la aplicación y sus límites de operación. Por medio de la definición de las entradas iniciales del sistema se describe el problema que se desea resolver y se provee de una vista conceptual del sistema propuesto. Subsecuentemente por medio del diálogo con el cliente y con el conocimiento más a detalle del entorno del problema se pueden abstraer otras entradas adicionales.

El modelo de análisis se concibe como una abstracción resumida y precisa de lo que debe realizar el sistema deseado y no de la forma en que se hará. El modelo que se obtendrá en el análisis debe incluir aquella información que sea significativa desde el punto de vista del mundo real, este modelo debe ser comprensible para el cliente del sistema y debe proporcionar una base útil para extraer los requisitos del sistema, es decir, los que realmente necesita el sistema, que sean congruentes internamente y además se puedan llegar a realizar.

La salida de esta etapa es un modelo formal que se describe a través de tres aspectos esenciales del sistema, por una parte se tienen los objetos y sus relaciones, el flujo dinámico de control de los mismos y la transformación funcional de datos que están sometidos en restricciones.

El *modelo de objetos* describe los objetos del sistema y sus relaciones, lo concibe desde un punto de vista fijo, principalmente se obtienen diagramas de objetos, que muestran las clases de objetos y sus arcos son las relaciones entre las clases del sistema. Entre las actividades a realizar están las siguientes:

- ◆ Descripción inicial del problema.
- ◆ Se identifican clases de objetos.
- ◆ Se identifican y añaden asociaciones entre clases.
- ◆ Se construye un diccionario que incluya descripciones de clases, atributos y asociaciones.
- ◆ Se organizan las clases de objetos a través de la herencia.

El *modelado dinámico* describe las interacciones existentes entre los distintos objetos del sistema, es decir, aspectos que cambian con el tiempo y especifica e instrumenta los aspectos del control del sistema. Esencialmente se obtiene el diagrama de estados, el cual muestra las transiciones entre estados que sufren los objetos causados por los sucesos. Entre las actividades a efectuar están las siguientes:

- ◆ Se identifican escenarios de secuencias típicas de interacción.
- ◆ Se establecen los sucesos entre objetos.
- ◆ Se desarrolla un diagrama de estados para cada clase que tenga un comportamiento importante.

El *modelado funcional* que describe las transformaciones de valores de datos que ocurren dentro del sistema, este modelo hace uso de diagrama de flujo de

datos con nodos que representan procesos y sus arcos son el flujo de datos. Entre las actividades a realizar están las siguientes:

- ◆ Se identifican valores de entrada y de salida del sistema.
- ◆ Por medio del diagrama de flujo de datos se muestran las dependencias funcionales según sea necesario.
- ◆ Se determina lo que hace cada función.
- ◆ Se identifican las restricciones y los criterios de optimización.

El modelo importante a definir completamente es el de objetos, debido a que es necesario saber qué está cambiando o transformándose antes de describir cuándo y cómo cambian precisamente los objetos.

### ***Diseño***

Durante esta fase el diseñador del sistema toma decisiones acerca de la arquitectura global del sistema, para ello lo organiza en subsistemas considerando la estructura de análisis, así como la estructura propuesta. Así mismo el desarrollador construye el modelo de diseño basándose en el modelo de análisis e incorpora detalles para la posterior programación; durante esta fase se pone atención en las estructuras de datos y los algoritmos necesarios para la codificación de las clases. Se escogen algoritmos para la codificación de las funciones del sistema, se refinan y optimizan las estructuras del modelo de objetos, y se determinan las relaciones entre objetos, todo con el fin de producir un diseño eficiente y práctico a la hora de codificar. Las principales actividades a realizar son las siguientes:

- ◆ Se organiza al sistema en subsistemas.
- ◆ Se selecciona la estrategia a utilizar, los almacenes de datos en términos de estructura de datos, archivos o bases de datos.

- ◆ Se refinan las operaciones del modelo de objetos a partir de los otros modelos, se busca una operación para cada proceso del modelo funcional y se establece una operación para cada suceso del modelo dinámico.
- ◆ Se diseñan los algoritmos para la codificación de operaciones.
- ◆ Se seleccionan estructuras de datos que soporten los algoritmos.
- ◆ Se ajusta la estructura de clases, así como las operaciones y métodos para incrementar el uso de la herencia.
- ◆ Se evoluciona las asociaciones analizando el recorrido de éstas.
- ◆ Se organizan las clases y asociaciones en módulos.

#### 4.3. Metodología de análisis y diseño (J. Martín-J. Odell)

Esta metodología establece una perspectiva estructurada y definida en cuanto a etapas de análisis y diseño. Propone un modelo para describir un tipo de objeto en cuanto a su estructura y otro modelo que describe lo que les ocurre a los objetos. Los modelos se representan con diagramas denominados esquemas, para objetos y para eventos. Las etapas de análisis y diseño se describen a continuación:

##### *Análisis de la estructura de objetos*

En el análisis de la estructura de objetos se definen las categorías de objetos, así como las formas en que éstas se asocian. Con este modelo se responde a las siguientes preguntas:

- ¿Qué tipos de objetos hay?
- ¿Cuáles son sus relaciones y funciones?
- ¿Qué sub-tipos y super-tipos son útiles?
- ¿Hay algún tipo de objeto compuesto por otros objetos?

Mediante este modelado el equipo de análisis se encarga de identificar y definir los objetos individuales de un sistema, es el paso inicial que ayudará posteriormente a

la definición de clases y sus estructuras de datos, así como a las categorías de un objeto en particular, o a su asociación con otro tipo de objeto.

Para organizar un volumen de objetos grande este modelo hace uso de la generalización de objetos mediante diagramas de generalización. También define si un objeto estará compuesto por otros objetos; a esto se le denomina jerarquía compuesta. Además propone la representación de objetos compuestos por medio de diagramas de composición.

#### *Análisis del comportamiento de objetos*

El análisis del comportamiento de objetos se ocupa del modelado de objetos en cuanto a su interacción en distintas situaciones, se centra en analizar la secuencia de eventos y cómo cambian los estados de los objetos, se realizan esquemas de eventos, en relación a la secuencia en que éstos ocurren, y los cambios que provocan en determinados objetos. Este modelado responde a las siguientes preguntas:

- ¿En qué estados puede estar el objeto?
- ¿Qué transacciones de estado se pueden dar? (diagramas de transición de estados)
- ¿Qué eventos ocurren?
- ¿Qué operaciones se llevan a cabo? (esquemas de eventos)
- ¿Qué interacciones ocurren entre los objetos?
- ¿Cuáles son las reglas de activación que se utilizan para relacionar el evento?
- ¿Cómo se representan las operaciones en los métodos?

Se establece en esta parte el denominado ciclo de vida de un objeto que puede ser representado mediante diagramas de reja, ante la sucesión de eventos posibles que pueden incidir en un objeto y por lo tanto modificar su estado. El diagrama de reja muestra los estados posibles y los estados de transición permisibles. En este tipo de diagramas se tiene la ventaja de una interacción entre ellos, estableciendo subestados de los estados.

En cuanto a la representación de los objetos y la manera en que se relacionan en el sistema, se usan los diagramas de flujo de objetos. A diferencia de los esquemas de eventos, éstos se usan para representación de procesos grandes y complejos. Los diagramas de flujos de objetos son parecidos a los diagramas de flujo de datos, ya que muestran las actividades, pero difieren en que el DFD sólo muestra la transferencia de datos entre las partes de los esquemas, y los DFO muestran cualquier tipo de transferencia, de cualquier cosa, de una actividad a otra, indicando con esto los objetos y las actividades que los producen e intercambian.

#### *Diseño de estructura y comportamiento de un objeto*

Las etapas de diseño de estructura de objetos y de diseño del comportamiento de objetos son unidas en una sola debido a la conformación natural de estructura de los lenguajes de programación orientados a objetos. En estas etapas se define una organización en cuanto a clases, subclasses, herencia y polimorfismo; y es también en donde se encuentra la actividad central de la orientación a objetos y la reutilización del código; para esta etapa se recomienda el uso de los diagramas de ramificación.

En estas etapas lo que se pretende es responder a las siguientes preguntas:

- ¿Qué clases se implantarán?
- ¿Qué estructura de datos utilizará cada clase?
- ¿Qué operaciones ofrecerá cada clase y cuáles serán sus métodos?
- ¿Cómo se implantará la herencia de clases y cómo afectará ésta a las especificaciones de los datos y las operaciones?
- ¿Cuáles son las posibles variantes de clases?

Es aquí donde se identifican las clases (tipos de objetos), superclases, subclasses, herencias y métodos a utilizar, así como el diseño en detalle de las estructuras de datos, de los métodos bajo técnicas que puedan ser o no por procedimiento. Se crea la entrada para los generadores de código, así como las pantallas y la generación de códigos, para finalizar con el desarrollo de los prototipos.

En la definición de clases se especifica la estructura de datos de cada uno de sus objetos, las operaciones a utilizar para el acceso a los objetos y los métodos correspondientes. También en esta etapa se define la herencia de clases, subclases y la utilización de herencia múltiple.

Se define también el polimorfismo de los objetos, como parte de la herencia se definen ciertos métodos que puedan variarse si se desea. Es decir, cuando un simple mensaje ocurre causará diferentes acciones si éste es recibido por diferentes objetos.

Finalmente, la metodología de Martin-Odell establece que el máximo potencial del diseño con orientación a objetos, radica en la reutilización del código, basándose en el concepto de "Igual que, excepto..." donde después de la manufactura de clases, o bien del retomar clases prefabricadas, se modifiquen en algunas partes de su estructura readaptandolas según sea la necesidad del sistema a desarrollar. Promueven a su vez la creación de clases con alta capacidad de reuso para proyectos o etapas posteriores.

## APÉNDICE B

## NOTES

## **PUNTOS A CONSIDERAR PARA LA ELABORACIÓN DE PROGRAMAS ENTENDIBLES**

### **Definición de estándares y control del proyecto**

La experiencia en la programación dicta que un programa es leído más veces de lo que se escribe, y es responsabilidad del programador hacer legibles sus programas.

La creación de un programa legible y confiable es un proceso creativo, por lo cual resulta complicado poner reglas estrictas que imperen sobre el estilo de programación; sin embargo, es posible establecer principios generales que mejoren la legibilidad de los programas, así como de sus nombres para su posterior mantenimiento. ¿Por qué razón es importante pensar en el mantenimiento del sistema? Bien, la razón es que no siempre los miembros del equipo de trabajo que desarrolla un sistema sean los mismos que le darán mantenimiento; por lo que se requerirá que las personas que se encarguen de esa tarea puedan comprender: la programación, el orden en que son llamados los programas y el nombre de las variables.

En un grupo de trabajo deben establecerse normas que definan la metodología a la que debe someterse todo el personal involucrado en el desarrollo de sistemas, con objeto de obtener productos de alta calidad que resulten fáciles de mantener por cualquier miembro del equipo de trabajo. Los programadores que trabajan en la empresa o institución deberán apegarse a los estándares convenidos para el desarrollo de sistemas.

En el diseño de un sistema se definen las especificaciones necesarias para su desarrollo e implantación conforme a los resultados del análisis. Para poder llevar a cabo su desarrollo es necesario diseñar las estructuras de las bases de datos y archivos necesarios para la integración de la información que utilizará, así como los procedimientos de entradas, salidas y procesos que permitirán satisfacer los

requerimientos identificados en la etapa de análisis; con ello se logrará optimizar el uso de los recursos de cómputo involucrados y se garantizará la facilidad de operación, la integridad de la información y la confiabilidad del sistema.

### **Calidad del software**

Para asegurar en mayor porcentaje el éxito de un sistema, tanto en la etapa de desarrollo como en la de mantenimiento y corrección de errores, se pueden tomar ciertas medidas al momento de desarrollar los programas que lo conformarán.

Existen técnicas para mejorar la calidad del software y a la vez evitar la existencia de errores en el código, o en su caso, hacer más fácil su localización. Estas técnicas darán mayor legibilidad al código escrito, tanto para el programador como para el personal que se ocupe de proporcionar el mantenimiento.

#### **1. Insertar espacios en blanco en el código**

Esto es algo muy simple que se puede hacer para mejorar la productividad del código. Los espacios en blanco permiten que el programador identifique los bloques lógicos del código, lo que lo hace más comprensible. Esta costumbre no representa diferencia en la velocidad de ejecución del programa, por lo que no hay un costo con esta mejora, excepto el papel al imprimirlos.

#### **2. Insertar sangrías en el código**

Hacer esto dentro de las estructuras de control tiene una gran influencia en la productividad al hacer más fácil la identificación de los diferentes ciclos del programa. Lo cual tampoco afecta la velocidad de ejecución del programa.

#### **3. Comentar el código**

Esta práctica resulta útil, tanto para el programador como para la persona que dará mantenimiento al sistema. Las líneas con comentarios son esenciales si el programador usa algún tipo de artificio ("truco") que piense que puede ser poco comprensible para otras personas, e incluso para la misma persona que elaboró el programa transcurrido unos meses. Los comentarios pueden servir también para

"marcar" bloques dentro del código, como cuando se lee un archivo, cuando se efectúan operaciones, cuando se manda a escribir a un archivo de salida, etc. Esta, al igual que las dos técnicas anteriores no afectan la velocidad de ejecución; sólo añaden claridad al programa al momento de leerlo. La experiencia determina que los comentarios al final de una línea son menos claros que los que se escriben en una línea completa.

Para ilustrar los tres puntos anteriores, a continuación se muestra una parte del código sin el apoyo de estas técnicas, y otro con ellas. La diferencia no es lo que hace el código, sino la forma de su presentación.

```
PROCEDURE Init
LPARAMETERS toOrdEntryForm
LOCAL InNumPams, ;
IcFilter
thisform.Left = 11
thisform.Top = 2
thisform.cOriginalFormCaption = thisform.Caption
InNumPams = PARAMETERS()
IF InNumPams = 0
thisform.cOriginalFormName = thisform.Name
thisform.Name=thisform.Name+ALLT(STR(oApp.AddInstance (thisform)))
thisform.Caption = thisform.Caption + ":" + ;
RIGHT(ALLT(thisform.Name), 1)
ENDIF
OrderEntry::Init()
```

Note lo difícil que es leer el código; en él hay varias secciones diferentes que no se distinguen entre sí. Verifique que es más fácil leerlo si se insertan espacios en blanco y sangrías:

PROCEDURE Init

LPARAMETERS toOrdEntryForm

LOCAL InNumParms, ;

lcFilter

thisform.Left = 11

thisform.Top = 2

thisform.cOriginalFormCaption = thisform.Caption

InNumParms = PARAMETERS()

IF InNumParms = 0

    thisform.cOriginalFormName = thisform.Name

    thisform.Name = thisform.Name +

    ALLT(STR(oApp.AddInstance(thisform)))

    thisform.Caption = thisform.Caption + ":" + ;

        RIGHT(ALLT(thisform.Name), 1)

ENDIF

OrderEntry::Init()

Hasta este punto, el código se lee más fácilmente, tal como si se tratara de instrucciones simples en inglés. Si ahora se añaden comentarios, se convierte en su propia documentación.

PROCEDURE Init

LPARAMETERS toOrdEntryForm

LOCAL InNumParms, ;

lcFilter

thisform.Left = 11

thisform.Top = 2

thisform.cOriginalFormCaption = thisform.Caption

```
InNumParms = PARAMETERS()
*~ Cambiar el título y el nombre del formulario antes de llamar
*~ OrderEntry::Init() para asegurarse de que se ha agregado
*~ el sufijo adecuado tanto al nombre como al título
*~ para permitir múltiples instancias de este formulario.
*~ Sólo se permite una instancia por formulario de entrada de pedidos.
*~ por lo que esta instancia está asociada a un formulario Entrada de
*~ pedidos, no nos preocupamos de cambiar el título.
```

```
IF InNumParms = 0
    thisform.cOriginalFormName = thisform.Name
    thisform.Name = thisform.Name +
        ALLT(STR(oApp.AddInstance(thisform)))
    thisform.Caption = thisform.Caption + ";" + ;
        RIGHT(ALLT(thisform.Name), 1)
ENDIF
```

```
OrderEntry::Init()
```

#### 4. Alinear el código

Se han realizado estudios que demuestran que al alinear el código se aumenta la legibilidad de éste. Por ejemplo, si se asignan valores a un conjunto de variables de memoria, deben alinearse los signos de igualdad para aumentar su legibilidad. Como se muestra, en los siguientes ejemplos:

Ejemplo 1.

```
m.lcname=company.iname
m.lcfirstname=company.firstname
m.lcaddr1=company.addr1
m.lcaddr2=company.addr2
m.lccity=company.city
m.lcstate=company.state
m.lczipcode=company.zipcode
```

**Ejemplo 2.**

m.lcname	= company.lname
m.lcfirstname	= company.firstname
m.lcaddr1	= company.addr1
m.lcaddr2	= company.addr2
m.lccity	= company.city
m.lcstate	= company.state
m.lczipcode	= company.zipcode

**5. Detallar los comandos compuestos**

Como una consecuencia lógica de la alineación, se deben detallar los comandos compuestos, haciendo una "lista" que sea más fácil de leer, tal como se muestra en los siguientes dos ejemplos:

**Ejemplo 1.**

```
DEFINE WINDOW myindow FROM 00,00 SIZE 20,40 ;  
IN WINDOW myparent FONT 'MS Sans Serif', 10 ;  
STYLE 'B' TITLE 'MyWindow' SYSTEM CLOSE ;  
FLOAT GROW SHADOW ZOOM
```

**Ejemplo 2.**

```
DEFINE WINDOW myindow ;  
FROM 00,00 ;  
SIZE 20,40 ;  
IN WINDOW myparent .  
FONT 'MS Sans Serif', 10  
STYLE 'B' .  
TITLE 'MyWindow' .  
SYSTEM
```

```
DEFINE WINDOW myindow ;
CLOSE ;
FLOAT ;
GROW ;
SHADOW ;
ZOOM ;
```

## 6. Adoptar convenciones al nombrar los objetos de los programas

Una convención en los nombres consiste en usar un conjunto uniforme de palabras que sirvan para describir las variables. Éstas deben ser fácilmente comprendidas por la persona que lea el código, no sólo para quien las ideó. Los objetos de un programa son las constantes, variables, procedimientos, funciones y tipos; cada uno de ellos representa una entidad del mundo real, y su función se refleja en la función del mundo real a la que representan. De acuerdo con esta concepción, los nombres de los objetos de un programa deben estar estrechamente relacionados con los nombres de las entidades del mundo real que modelan.

Por ejemplo, si un programa obtiene un reporte de los alumnos titulados en la carrera de ingeniería en computación, y se relaciona con las entidades nombre del alumno, año de ingreso y carrera, éstas deben representarse en el programa por medio de objetos que pueden tener los nombres de: *nombre\_alumno*, *año\_ingreso\_alumno* y *carrera\_alumno*. Esta sería una opción, ya que no es recomendable que sólo se les llamara: *nombre*, *año* y *carrera*, porque no será evidente para el lector que estos nombres pudieran referirse a los datos de titulación del alumno. Peor que esto resulta elegir nombres que no tienen relación con las entidades que modelan, como: abreviaturas criptográficas, identificadores de una sola letra o nombres de personas.

Una mala costumbre es elegir nombres cortos que son fáciles de introducir en el teclado, ya que esto puede originar que los programas sean incomprensibles. La variable *f\_esí* significaría algo para el programa y para el programador, pero no para el lector. Observe los dos programas que se presentan a continuación; el primero usa nombres cortos no significativos:

```
Program CF(input, output);
  var t,f : real;
  begin
    read(t);
    f := t *9/5 + 32;
    write(f);
  end.
```

El segundo programa no contiene esta ilegibilidad; al leerlo se piensa en la función del programa, a diferencia del primero:

```
Program Convierte_Celsius_a_Fahrenheit(input, output);
  var Fahrenheit, Celsius : real;
  begin
    read(Celsius);
    Fahrenheit:= Celsius *9/5 + 32;
    write(Fahrenheit);
  end.
```

Si el lector no conoce la fórmula para convertir la temperatura de grados Celsius a Fahrenheit, es poco probable que pueda deducir lo que hace el primer programa. En el segundo ejemplo

## **APÉNDICE NORMA ISO 9000**

## APÉNDICE:

### NORMA ISO 9000

#### [ISO 8402 / 3.1 - "Políticas de calidad"]

La norma define el término de *Política de calidad* como las directrices y objetivos generales de una organización o empresa, concenientes a la calidad; los cuales son formalmente expresados por la alta dirección.

#### [ISO 9000-1 / 4.1 Inciso b -"Objetivos claves"]

Objetivos claves y responsabilidades para la calidad.

La empresa o institución buscará mejorar la calidad de sus propias operaciones o actividades para satisfacer las necesidades del cliente y otros interesados.

#### [ISO 9000-1 / 4.5 - "Facetas de calidad"]

La relación consiste en que tanto el área solicitante como el área o empresa que elabore el sistema deberán estar comprometidas durante todo el ciclo de vida del desarrollo de sistemas. De hecho en este punto se habla de las *Facetas de calidad* que consisten esencialmente de cuatro puntos claves que se mencionan a continuación:

1. La calidad en la definición del producto  
Básicamente consiste en cumplir con las necesidades del producto (sistema) y con los requisitos respectivos, es decir, para el que fuese creado.
2. La calidad en el diseño del producto  
Se hace referencia al conjunto de características del diseño que influyen en el desempeño esperado del sistema. Se verificarán las características que permitan conocer la robustez del sistema, es decir, el desempeño del sistema bajo las condiciones de carga de trabajo variables.

**3. La calidad en la conformidad con el diseño del producto**

El sistema deberá mantenerse constantemente y deberá cumplir con la consistencia del diseño original.

**4. La calidad en el soporte del producto**

El sistema deberá tener un soporte a través de su ciclo de vida con el fin de cumplir los requerimientos y características estipulados por el cliente, y satisfacer las aplicaciones para las que fue creado.

**[ISO 9000-1 / 4.6 - "Concepto de un proceso"]**

Básicamente está fundamentado en el hecho de que todo trabajo se lleva a cabo mediante un proceso, es decir, se definen tanto entradas como salidas del proceso. Dichas salidas deberán ser tanto tangibles como intangibles. Cabe mencionar que el proceso es una transformación que agrega valor. En un proceso se involucra gente y otros recursos. Para la elaboración de un sistema informático se conceptualiza y se da forma a la información, se define los requisitos, el estado que debe guardar la información y la posible retroalimentación de la misma. Otros recursos son el hardware y el software.

**[ISO 9000-1 / 4.9.2 - "Revisión de la dirección"]**

La dirección tiene el deber de llevar sistemáticamente la evaluación de las *Políticas de Calidad*, un medio puede ser aplicando auditorías internas.

**[ISO 9001/1 - "Objetivo y campo de aplicación"]**

La norma refiere que sus requisitos están orientados principalmente para lograr la satisfacción del cliente, previniendo la no conformidad en todas sus etapas partiendo del diseño hasta el servicio.

**[ISO 9001 / 4.1.1 - "Política de calidad"]**

La dirección deberá incluir en sus políticas de calidad los objetivos para la calidad y su compromiso con la calidad. Dichas políticas deberán ser congruentes con las metas organizacionales y con las necesidades y expectativas de sus clientes. También se encargará de que sean entendidas, implantadas y mantenidas en la institución o empresa.

**[ISO 9001 / 4.1.2 - "Organización"]**

La organización deberá estar definida en cuanto a jerarquía, responsabilidades y autoridades relacionadas con las actividades para la calidad. También involucra la iniciativa de acciones para prevenir la ocurrencia de no conformidades relacionadas con el desarrollo del sistema.

**[ISO 9001 / 4.1.2.2 - "Recursos"]**

El área o empresa desarrolladora del sistema, identificará las necesidades y requerimientos fundamentales para la creación, diseño y desarrollo del sistema. También proporcionará una lista de recursos para cumplir con el plan de desarrollo del proyecto y asignará personal capacitado que participe en las actividades de diseño, producción, verificación e instalación del sistema. En caso de no contar con personal capacitado disponible, se encargará de contratar personal factible para el puesto.

**Nota:** Se entiende como recursos: el hardware y el software. Ejemplos tales como PC, workstation, tarjetas de red, paquetería, lenguajes de programación, manejadores de base de datos etc.

**[ISO 9001 / 4.1.3 - "Revisión de la dirección"]**

La dirección es la responsable de revisar las políticas y objetivos de calidad establecidos para asegurar su adecuación y efectividad continua, para ello deberá utilizarse y mantenerse la bitácora respectiva.

**[ISO 9001 / 4.2.1 - "Sistema de Calidad/Generalidades"]**

El proveedor o área desarrolladora debe establecer, documentar y mantener un sistema de calidad como medio que asegure que el sistema elaborado es conforme con los requisitos especificados. También debe prepararse un manual de calidad congruente con los requisitos de esta norma. El manual debe hacer referencia a los procedimientos del sistema de calidad y describir la estructura de la documentación usada en el sistema de calidad.

**[ISO 9001 / 4.2.2 - "Procedimientos del sistema de calidad"]**

La empresa o área desarrolladora debe:

- A. Preparar procedimientos documentados de acuerdo a los requisitos de esta norma y la política de calidad establecida por la empresa o área desarrolladora.
- B. Implantar el sistema de calidad y sus procedimientos documentados

El alcance y detalle de los procedimientos que forman parte del sistema de calidad deben depender de la complejidad del trabajo, de los métodos usados, y de las habilidades, y capacitación requerida por el personal involucrado en llevar a cabo la actividad.

Nota: Los procedimientos documentados pueden hacer referencia a instrucciones de trabajo que definan cómo se realiza una actividad.

**[ISO 9001 / 4.2.3 - "Planeación de la calidad"]**

La empresa o área desarrolladora debe definir y documentar cómo se deben cumplir los requisitos para la calidad. La planeación debe ser consistente con todos los otros requisitos del sistema de calidad del desarrollador y debe estar documentada. Además debe considerar las siguientes actividades para cumplir con requisitos de proyectos o contratos:

- A. Preparación de planes de calidad.

- B. Identificación y adquisición de dispositivos, recursos y habilidades que sean necesarias para lograr la calidad requerida.
- C. Asegurar la compatibilidad de los procedimientos de diseño del sistema, considerando su producción, instalación, inspección, pruebas y servicio, así como la documentación aplicable.
- D. Actualización según sea necesario del control de calidad, de las técnicas y del desarrollo de instrumentación nueva.
- E. Identificación de cualquier requisito de medición incluyendo la capacidad que exceda los avances conocidos, con anticipación suficiente para que se desarrolle esa capacidad.
- F. Identificación de las verificaciones adecuadas en las fases apropiadas de la elaboración del sistema.
- G. Aclaración de las normas de aceptación para todas las características y requisitos, incluyendo aquellas que contengan algún elemento subjetivo.
- H. Identificación y preparación de registros de calidad.

**[ISO 9001 / 4.3 - "Revisión del contrato"]**

El área desarrolladora o empresa externa deberá establecer y mantener procedimientos documentados (como acta de acuerdos) para la revisión del contrato y principalmente para la coordinación de actividades que surjan del convenio efectuado. Será necesario establecer y documentar, apropiadamente, las condiciones contractuales antes de la puesta en marcha del proyecto. Se deben definir las líneas de comunicación con el cliente a través de entrevistas, cuestionarios, reportes etc. Es importante señalar que se deben llevar registros actualizados de las revisiones a los contratos.

Es conveniente que en el contrato se definan de manera clara y concisa los siguientes puntos:

- ◆ El alcance, compromiso y responsabilidades de las partes firmantes del contrato.

- ◆ Identificar y delimitar casos de contingencias con sus correspondientes responsabilidades.
- ◆ Establecer una terminología entendible por ambas partes.
- ◆ Acordar o establecer la responsabilidad del desarrollador en posibles subcontratos de trabajo.
  
- ◆ El desarrollador del sistema tiene la capacidad de exigir requerimientos contractuales.
  
- ◆ El comprador del sistema tiene la capacidad de exigir obligaciones contractuales.

**[ISO 9001 / 4.4.1 - "Control de diseño - Generalidades"]**

El desarrollador del sistema debe establecer y mantener procedimientos documentados para controlar y verificar el diseño del sistema con el fin de que se cumplan los requisitos especificados.

Se tiene la finalidad de conocer los datos de inicio del diseño, los cuales deben ser colectados, documentados y revisados. Se requiere definir las fuentes de información de inicio, evaluar la validez de dicha información y analizar si son completos y claros.

Deberán definirse los métodos de obtención de datos y establecer claramente las características del sistema (condiciones, seguridad etc.). Además se establece un plan de desarrollo, el cual debe cumplir con los siguientes puntos:

- ◆ Definición clara y concisa del proyecto, incluyendo los objetivos y metas que se deseen alcanzar.
  
- ◆ La organización del proyecto en general, deberá definir más específicamente los recursos que se utilizarán en el proyecto.
  
- ◆ El uso de una metodología que conducirá a la elaboración del sistema. Se define el desarrollo del proyecto a través de fases o etapas.

- ◆ Se delimitan perfectamente las actividades y recursos para cada fase, incluyendo una calendarización para cada una.
- ◆ Se definen las herramientas, técnicas, y convenciones para el desarrollo del proyecto.

**[ISO 9001 / 4.4.2 - "Planeación del diseño y desarrollo"]**

El desarrollador del sistema debe elaborar planes para cada actividad de diseño y desarrollo. Los planes deben describir o hacer referencia a estas actividades, y definir la responsabilidad para su implantación. Las actividades de diseño y desarrollo deben estar asignadas a personal calificado y equipado con los recursos (hardware y software) adecuados. En caso de no contar con personal, el desarrollador se encargará de buscar a la gente idónea y factible al puesto. Los planes aceptados deben actualizarse según la evolución del diseño.

**[ISO 9001 / 4.4.3 - "Interrelaciones organizacionales y técnicas"]**

Deben de estar bien delimitadas y definidas las interrelaciones organizacionales y técnicas a utilizar, principalmente las funciones a realizar por el grupo de trabajo y el grupo que proporcione la información para datos a utilizar en el sistema. Dicha información deberá ser documentada, transmitida y revisada regularmente.

**[ISO 9001 / 4.4.4 - "Datos de entrada del diseño"]**

Se deben identificar y documentar los requisitos para los datos de entrada del diseño, relacionados con el producto. Incluyendo los requisitos legales y regulaciones aplicables y en conjunto con el cliente seleccionarlos y revisarlos para su adecuación. Los requisitos incompletos, ambiguos o conflictivos, deben ser resueltos con los responsables del establecimiento de éstos.

Los datos de entrada del diseño deben tomar en consideración los resultados de cualquiera de las actividades de revisión del contrato.

**[ISO 9001 / 4.4.5 - "Resultados del diseño"]**

Los resultados del diseño deben documentarse y expresarse en términos que puedan verificarse y validarse contra los requisitos de entrada del diseño.

Los resultados del diseño deben:

- a) Cumplir con los requisitos de entrada del diseño.
- b) Contener o hacer referencia a los criterios de aceptación.
- c) Identificar aquellas características del diseño que son cruciales para la seguridad y el funcionamiento apropiado del sistema (tales como: requisitos de operación, almacenamiento, manejo, mantenimiento y disposición después del uso).

Los documentos del resultado del diseño deben revisarse antes de su liberación.

**[ISO 9001 / 4.4.6 - "Revisión del diseño"]**

En las etapas apropiadas del diseño deben planearse y realizarse revisiones formales documentadas de los resultados del diseño. Los participantes en cada revisión del diseño deben incluir representantes de todas las funciones involucradas en relación a la etapa del diseño; así también se harán participar a otros especialistas según se requiera. Deben mantenerse registros de tales revisiones.

**[ISO 9001 / 4.4.7 - "Verificación del diseño"]**

En etapas apropiadas del diseño debe realizarse la verificación del mismo, con objeto de asegurar que los resultados generados del diseño del sistema cumplan los requisitos de entrada. Las medidas de control del diseño deben ser registradas (documentadas).

Además de realizar las revisiones del diseño (véase 4.4.6), la verificación del diseño puede incluir actividades tales como:

- ⇒ La realización de cálculos alternativos; como lo serían pruebas con datos reales.

- ⇒ La comparación del diseño nuevo con un diseño similar probado, si está disponible; mediante la verificación de corridas en paralelo.
- ⇒ La adopción de pruebas y demostraciones.
- ⇒ La revisión de los documentos relativos a la etapa del diseño, antes de su liberación.

**[ISO 9001 / 4.4.8 - "Validación del diseño"]**

Debe realizarse la validación del diseño para asegurar que el sistema (producto) cumple con las necesidades y requisitos definidos por el usuario, es decir, que cumpla con las aplicaciones y necesidades del cliente.

Notas:

- ⇒ La validación del diseño sigue a la verificación del diseño, si ésta fue satisfactoria (Véase 4.4.7).
- ⇒ La validación se realiza generalmente bajo condiciones de operación definidas (pruebas con datos reales, corridas en paralelo etc.).
- ⇒ Normalmente la validación se realiza sobre el producto final, pero puede ser necesaria en etapas iniciales previas a la terminación del sistema (producto).
- ⇒ Pueden realizarse validaciones múltiples si hay diferentes usos intencionados.

**[ISO 9001 / 4.4.9 - "Cambios al diseño"]**

Todos los cambios y modificaciones al diseño del sistema debe ser identificados, documentados, revisados y aprobados por personal autorizado. Para este caso el *Comité de Informática* hará las designaciones respectivas antes de los cambios al diseño del sistema. El personal designado debe considerar:

- ◆ **Antes de aceptar un cambio de diseño, su aprobación debe ser confirmada y evaluar detenidamente los efectos posteriores.**
- ◆ **Definir claramente las actividades que serán llevadas a cabo durante la modificación.**
- ◆ **Establecer las herramientas, técnicas y metodologías a utilizarse al efectuar la modificación.**

**[ISO 9001 / 4.5 - "Control de documentos y datos"]**

**El creador del sistema debe generar, establecer y mantener procedimientos documentados. Dichos documentos siempre deberán estar disponibles, para quien los solicite. También es importante definir y clasificar los tipos de documentos generados, en otras palabras, definir claramente los tipos de manuales con que cuenta la institución o empresa.**

**Entre otros puntos a documentar se encuentran:**

- ◆ **Los documentos concernientes a la planeación del sistema, es decir, describir el progreso de las actividades entre el desarrollador y el área usuaria.**
- ◆ **Documentar con diagramas y flujogramas las técnicas utilizadas en el diseño del sistema.**
- ◆ **Documentar las instrucciones, los módulos y procedimientos de programas, definir la sistemática de la codificación, indexación y formato de los documentos.**
- ◆ **Definir claramente las entradas y salidas del sistema, así como describir los planes de verificación.**

**Nota: Los documentos y datos pueden estar en los formatos establecidos o en medios electrónicos, o cualquier otro.**

**[ISO 9001 / 4.5.2 - "Emisión de documentos"]**

Previo a la liberación final del sistema, todos los documentos deben ser revisados y aprobados por personal autorizado. Dichas emisiones, como lo es el manual de procedimientos, deben estar a la disposición de los involucrados en el análisis, diseño, instalación y mantenimiento del sistema, de tal forma que puedan ser consultados en cualquier momento del desarrollo; logrando con esto mayor efectividad y calidad en el desarrollo del sistema. Debe garantizarse que los documentos estén vigentes y previamente aprobados antes de ser puestos a disposición del equipo de trabajo. También se establece el uso de una lista maestra o un control de documentos para impedir el uso de documentos obsoletos o inválidos. Cabe señalar que en uso de archivos de computadora debe ponerse especial atención para su aprobación.

**[ISO 9001 / 4.5.3 - "Cambios en documentos y datos"]**

Especifica que los cambios de los documentos que se involucran en el desarrollo del sistema, deben pasar por una serie de revisiones y aprobaciones por el personal que desarrolló y aprobó originalmente el documento. En caso de ser válido el cambio debe especificarse y detallarse, tanto en el documento como en sus anexos. Es importante volver a emitir los documentos después de efectuados los cambios.

**[ISO 9001 / 4.6.2 - "Evaluación de subcontratistas"]**

El desarrollador debe:

1. Evaluar y posteriormente seleccionar al personal de trabajo con base en su habilidad y tomando en cuenta los factores que aseguren un trabajo con calidad.
2. Definir el control y alcance sobre el personal subcontratado, estableciendo claramente los alcances pretendidos con este personal, y tomado en cuenta un registro que abale la calidad del desempeño del personal.
3. Mantener un registro de calidad aceptable de las personas subcontratadas.

**[ISO 9001 / 4.8 - "Identificación y rastreabilidad del producto"]**

La norma menciona que donde sea aplicable, el proveedor (desarrollador del sistema) debe establecer y mantener procedimientos documentados para identificar al sistema por medios adecuados, desde su recepción, durante todas las etapas de creación del sistema (ciclo de vida del software) y hasta su instalación.

El desarrollador debe establecer y mantener procedimientos documentados, donde y tan extensos como se especifique en la rastreabilidad, cuando ésta sea un requisito especificado, con objeto de contar con una identificación única de aquellas partes de código que sean fundamentales para el desarrollo propio del sistema (ejemplo: procedimientos de consistencia e integridad del sistema). Esta identificación debe registrarse (4.16).

**[ISO 9001 / 4.9 - "Control de proceso"]**

El desarrollador del sistema debe identificar y planear los procesos de instalación producción y servicio, previniendo posibles cambios al diseño del sistema, con objeto de asegurar la calidad del mismo. Entre otros puntos a cubrir deberá considerarse:

1. Establecer procesos documentados para definir la manera de producir, instalar y dar servicio al sistema.
2. Definir los procesos a través de flujogramas.
3. Verificar el cumplimiento con las normas y códigos de referencia, los planes de calidad y los procedimientos documentados.
4. Supervisar y controlar los parámetros adecuados del proceso y las características del sistema.
5. Definir la ejecución de las etapas del proceso y actividades en procedimientos e instrucciones.

6. Establecer los elementos o criterios para la ejecución del sistema, de manera práctica y lo más claro posible, a través de especificaciones escritas, tales como: manuales , diagramas e ilustraciones.
7. Prever el mantenimiento adecuado para asegurar la capacidad efectiva del sistema.

**[ISO 9001 / 4.10.1 - "Inspección y prueba - Generalidades"]**

El desarrollador del sistema (proveedor) debe establecer y mantener procedimientos documentados para las actividades de inspección y prueba, a fin de verificar que se cumplan los requisitos especificados. La inspección y pruebas requeridas y los registros establecidos deben estar detallados en el plan de calidad y en los procedimientos documentados.

**[ISO 9001 / 4.10.5 - "Registros de inspección y prueba"]**

La empresa o área desarrolladora deberá establecer y mantener registros (documentación) que contengan la evidencia que al sistema se le han aplicado las pruebas correspondientes. Estos registros deberán mostrar claramente si el sistema ha pasado o fallado las pruebas de acuerdo con los criterios de aceptación definidos. En caso de que no se pasen las pruebas correspondientes deben aplicarse los procedimientos para el control de productos no conformes.

**[ISO 9001 / 4.11 - "Control de inspección, medición y prueba"]**

El área desarrolladora o empresa externa debe establecer y mantener procedimientos documentados para controlar e inspeccionar las pruebas realizadas al sistema, esto con el fin de demostrar la confiabilidad del producto (software) con los requisitos especificados. Las pruebas a realizar pueden variar, se sugieren casos pruebas, pruebas con datos reales, pruebas de funcionalidad, pruebas de rendimiento etc.

**[ISO 9001 / 4.12 - “Estado de inspección y prueba”]**

El estado de inspección y prueba del sistema elaborado debe identificarse utilizando medios adecuados, que indiquen la conformidad o no conformidad del sistema con respecto a la inspección y prueba realizadas. La identificación del estado de inspección y prueba abarca todas las fases tal como se establece en el plan de calidad o procedimientos documentados, con el fin de asegurar que sólo el producto que ha pasado las inspecciones y pruebas requeridas, se usa o se instala.

**[ISO 9001 / 4.13 - “Revisión y disposición de productos no conformes”]**

Establece que en apoyo a procedimientos documentados como lo son los procedimientos para el desarrollo de sistemas, es válido el retrabajar y reprogramar el sistema en sus distintos módulos, con el fin de satisfacer los requisitos especificados.

**[ISO 9001 / 4.14.2 - “Acción correctiva”]**

El área desarrolladora o empresa externa debe establecer procedimientos para acciones correctivas, las cuales deben incluir:

- ◆ El informe y manejo efectivo de los productos no aceptados por el cliente.
- ◆ Investigar la causa de no aceptación del sistema y encontrar la acción correctiva necesaria para prevenir la no recurrencia de dicho problema.
- ◆ La aplicación de las acciones correctivas que aseguren la efectividad del sistema.

**[ISO 9001 / 4.14.3 - "Acción preventiva"]**

El área desarrolladora o empresa externa debe considerar como acciones preventivas el uso de las fuentes apropiadas de información, tales como los procesos y operaciones de trabajo, las cuales afectan la calidad del sistema. Como ejemplos están: casos pruebas, pruebas con datos reales, pruebas de funcionalidad, pruebas de rendimiento, etc. En concreto: detectar, analizar y eliminar las causas potenciales de no aceptación.

**[ISO 9001 / 4.15.1 - "Generalidades"]**

El área desarrolladora o empresa externa debe establecer y mantener procedimientos documentados para manejo o uso, conservación o mantenimiento y entrega del sistema.

**[ISO 9001 / 4.15.2 - "Manejo o uso"]**

El área desarrolladora o empresa externa debe suministrar métodos de uso que eviten el daño o deterioro del sistema. En otras palabra entregará la copia maestra del sistema así como una copia de respaldo para uso continuo. También entregará manuales y guías de usuario.

**[ISO 9001 / 4.15.5 - "Conservación o mantenimiento"]**

El área desarrolladora o empresa externa debe aplicar métodos apropiados para operar y conservar el sistema.

**[ISO 9001 / 4.15.6 - "Entrega"]**

El área desarrolladora o empresa externa debe tomar medidas necesarias para proteger la calidad del sistema, después de realizar las pruebas pertinentes. Se entregará un número de copias por cada sistema liberado, el tipo de formato y la versión que se libera. También lo concerniente al uso de la licencia.

**[ISO 9001 / 4.16 - "Control de registros de calidad"]**

En forma general la norma menciona que el proveedor (creador del sistema) debe establecer y mantener procedimientos documentados para identificar, compilar, codificar, acceder, archivar, almacenar, conservar y disponer de tales registros.

**[ISO 9001 / 4.18 - "Capacitación"]**

El área informática o empresa externa deberá establecer y mantener procedimientos documentados para cubrir e identificar las necesidades de capacitación. Es responsabilidad de ésta capacitar a todo el personal que ejecute actividades que afecten la calidad y desempeño del sistema. Dicho personal deberá ser capacitado en base a su formación, capacidad y experiencia adecuadas según se requiera. Al efectuar la capacitación es importante no perder de vista las herramientas utilizadas, las técnicas, metodologías y los recursos de cómputo usados al diseñar el sistema

**[ISO 9001 / 4.19- "Servicio"]**

Cuando el servicio sea un requisito especificado, el desarrollador debe establecer y mantener procedimientos documentados para realizar este servicio y para verificar e informar que dicho servicio cumple con tales requisitos.

**[ISO 9004-1 / 4.2 - "Política de Calidad"]**

El conjunto de objetivos y directrices de una organización en lo que a calidad respecta, tal como son expresados por la alta gerencia. La gerencia deberá desarrollar y comunicar su política corporativa de calidad. Afirma que esta política debería ser consistente con otras de la empresa. Así también deberá proporcionar los pasos necesarios para su implantación.

## **BIBLIOGRAFIA**

- "Administración de personal un punto de vista y un método"  
Paul Pigors  
Massachusetts Institute of Technology  
Charles A Myers  
Compañía Editorial Continental S.A.
- "Public Policy Development"  
Robert F. Baker, Richard M. Mickaels, Everett S. Preston  
Wiley Interscience Publication  
John Wiley & Sons
- "ISO 9000 Meeting the new International Standards"  
Johnson Perry Lawrence
- "ISO 9000"  
Brian Rothery  
Segunda Edición, Editorial Panorama
- "ISO 9000 / Una Visión general"  
Eduardo Cadena Gómez.  
Revista, Soluciones Avanzadas. Abril 1996
- "Dirección electrónica: Internet"  
<http://WWW.iso.ch>
- "Norma mexicana IMNC "  
Instituto Mexicano de Normalización y Certificación  
NMX-CC-001: 1995 / ISO 8402: 1994  
NMX-CC-002/1: 1995 / ISO 9000-1: 1994  
NMX-CC-003: 1995 / ISO 9001: 1994
- International Standard - ISO 9000-3:1991  
Guidelines for the application of ISO 9001 to the development,  
supply and maintenance of software.
- "Análisis y diseño de Sistemas de Información"  
James A. Senn  
Segunda Edición, McGraw-Hill

- **"The Analysis, Desig, and Implementation of Information Systems"**  
Henry C. Lucas, Jr.  
Fourth Edition, McGraw-Hill
- **"LAN aplicaciones - Client/Server Databases"**  
Jim Krochmal and Larry Morris  
NRP New Riders Publishing, Carmel, Indiana  
Enterprise Series LAN operating systems
- **"Management Information Systems:"**  
A managerial end user perspective  
International student edition  
James A. O'Brien  
Irwin, Boston, Mma 02116
- **"Introducción a los sistemas de información ejecutiva y a la evaluación de herramientas para su desarrollo"**  
Marisol González Lozano.  
Revista, Soluciones Avanzadas. Julio 1996.
- **"Internet: Tecnología de la información o Sistema de información"**  
Natalia Volkov  
Revista, Soluciones Avanzadas. Julio 1996.
- **"Software Orientado a Objetos"**  
Ann L. Winbland, Samuel D. Edwards, David R. King.  
Addison-Wesley / Diaz de Santos, 1993
- **"Análisis y diseño orientado a objetos"**  
James Martin, James J. Odell  
Primera edición, Prentice Hall.
- **"Modelado y diseño orientados a objetos"**  
Metodología OMT  
James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy,  
Williamiam Lorensen.  
Primera edición, Prentice Hall.
- **"Object technology In application Development"**  
Daniel Tkack, Richard Puttick.  
IBM. International Technical Support Organization 1994

- **"Metodos orientados a objetos"**  
Ian Graham  
Addison- Wesley /Diaz de Santos  
Segunda edición, Delaware E.U.A.
- **'Organización y Administración de Centros de Cómputo'**  
Heriberto Olguín Romo
- **"Herramienta Case "**  
**"Metodología estructurada para el desarrollo de los sistemas"**  
William S. Davis, Paraninfo
- **"An ISO 9000 Approach to building Quality software"**  
Osten Oskarsson & Robert L. Glass,  
Prentice Hall PTR, 1996
- **"Ingeniería de Software"**  
Richard Fairley  
McGraw - Hill, 1994
- **"Ingeniería del Software, Un enfoque práctico"**  
Roger S. Pressman,  
Tercera edición, McGraw-Hill
- **"CASE la automatización del Software"**  
Autor: Carma McClure  
Serie Paradigma, Addison-Wesley Iberoamericana.