



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

31  
24

"CAMPUS ARAGON!"

**ESTANDARIZACION EN LA PROGRAMACION DE  
SISTEMAS EN INFORMIX - 4 GL: FRENOS,  
DETONADORES Y PROPULSORES.**

**T E S I S**

Que para obtener el Título de:

**INGENIERO EN COMPUTACION**

P r e s e n t a:

**MANUEL HERRERA REYES**

Asesor: Ing. Ernesto Peñalosa Romero

TESIS CON  
FALLA DE ORIGEN

México, D. F. 1997



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.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

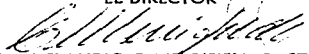
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
ARAGÓN  
DIRECCIÓN

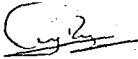
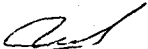
MANUEL HERRERA REYES  
P R E S E N T E .

En contestación a su solicitud de fecha 2 de abril del año en curso, relativa a la autorización que se le debe conceder para que el señor profesor, Ing. ERNESTO PEÑALOZA ROMERO pueda dirigirle el trabajo de Tesis denominado, "ESTANDARIZACION EN LA PROGRAMACION DE SISTEMAS EN INFORMIX-4GL: FRENOS, DETONADORES Y PROPULSORES", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE  
"POR MI RAZA HABLARA EL ESPÍRITU"  
San Juan de Aragón, México., 7 de abril de 1997  
EL DIRECTOR

  
M en I CLAUDIO C. MERRIFIELD CASTRO

  
  
c c p Jefe de la Unidad Académica.  
c c p Jefatura de Carrera de Ingeniería en Computación.  
c c p Asesor de Tesis.  
CCMC/AIR/vr

*Desde que esa aparición  
modelo de la realidad que se ignora,  
vino de una masacre subterránea  
que vive entre la difuminación de la conciencia  
y la punta de mi realidad...*

*A la inquebrantable voluntad de poder  
que de la fe funda con hierro  
el majestuoso imperio.*

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>I</b>
<b>I. FUNDAMENTOS TÉCNICOS</b>	
<b>INTRODUCCIÓN A LAS BASES DE DATOS</b>	<b>6</b>
A. LENGUAJES DE MANIPULACIÓN DE DATOS.	8
B. DICCIONARIO DE DATOS	8
C. ENTIDADES Y RELACIONES	8
D. ATRIBUTOS Y DOMINIOS.	9
<b>1. MODELOS DE DATOS</b>	<b>10</b>
MODELO RELACIONAL	10
CLAVES	11
<b>2. ALGEBRA RELACIONAL</b>	<b>12</b>
OPERACIONES DEL ALGEBRA RELACIONAL.	12
OPERACIONES FUNDAMENTALES DEL ALGEBRA RELACIONAL.	12
UNIÓN	12
INTERSECCIÓN	13
DIFERENCIA	13
PRODUCTO CARTESIANO	14
OPERACIONES ESPECIALES DEL ALGEBRA RELACIONAL.	14
SELECCIÓN	15
PROYECCIÓN	16
REUNIÓN	17
DIVISIÓN	18
<b>3. EJEMPLO DE LA APLICACIÓN DEL ALGEBRA RELACIONAL EN LA CONSULTA DE INFORMACIÓN A UNA BASE DE DATOS RELACIONAL.</b>	<b>19</b>
<b>II. FRENOS</b>	
<b>FRENOS</b>	<b>23</b>
¿QUÉ SON LOS FRENOS?	23
CÓMO Y DÓNDE OPERAN LOS FRENOS	24
CLARIDAD EN EL OBJETIVO DEL ÁREA EN GENERAL.	25
ORIENTACIÓN CLARAMENTE DEFINIDA EN LA PLANEACIÓN.	25
EL PROGRAMADOR DEBE CONOCER LOS OBJETIVOS PARTICULARES Y GENERALES EL SISTEMA QUE SE LE ASIGNA PROGRAMAR.	26
<b>LOS ESTÁNDARES</b>	<b>26</b>
CONSECUENCIAS DE LA INADECUADA ESTRUCTURACIÓN DE LOS ESTÁNDARES.	28
NOMENCLATURAS	31
<b>III. DETONADORES</b>	
<b>EL ORIGEN</b>	<b>33</b>
<b>EL CRITERIO MÁS USUAL</b>	<b>34</b>
<b>UN CASO FRECUENTE</b>	<b>34</b>
<b>LA PARTICIPACIÓN DEL DISEÑADOR.</b>	<b>39</b>

#### **IV. PROPULSORES**

<i>PROLEGÓMENOS FUNDAMENTALES.</i>	42
EVITAR LOS FEUDOS DENTRO DEL ÁREA DE SISTEMAS.	43
ENTONCES: ¿QUÉ HACER?	46
DESAFÍOS	47
<i>EL PROPULSOR</i>	47
<i>¿Y EN LOS SISTEMAS DE PROGRAMACIÓN RELATIVAMENTE PEQUEÑOS?</i>	48
<i>METODOLOGÍA PARA LA CREACIÓN DE ESTÁNDARES.</i>	48
48	
ESTUDIO PRELIMINAR	52
ANÁLISIS	55
DISEÑO	58
CONSTRUCCIÓN DE ESTÁNDARES DE PROGRAMACIÓN	62
PRUEBAS	65
CONTROL DE CALIDAD	66
DESCRIPCIÓN DE PROCEDIMIENTOS	67
CONVERSIÓN DE DATOS	68
IMPLANTACIÓN	69
CONCLUSIÓN	72
<b>GLOSARIO</b>	<b>74</b>
<b>BIBLIOGRAFÍA</b>	<b>79</b>

## INTRODUCCIÓN

## Introducción

A nuestra generación ha tocado contemplar un espectáculo por demás fascinante y determinante en todos los aspectos de nuestras vidas, este fenómeno -para los que nos ocupamos en invertir algunas elucubraciones a este suceso sin igual y, para muchos otros también- nos ha obligado a cambiar de manera radical nuestras formas de vida y de apreciar la naturaleza, en suma, nos ha obligado a generar una nueva filosofía sobre el hombre mismo y su destino. Nos ha planteado nuevos retos e incertidumbres.

Conscientes del avasallamiento tecnológico, prestos estamos a presenciar un ambiente compuesto por tecnologías redefinidas para satisfacer mejor las necesidades de los usuarios.

Virtualmente, hoy día contamos con aditamentos tecnológicos para cubrir totalmente todas nuestras necesidades, cualesquiera que estas sean. Y qué decir de los conceptos y términos que constantemente se manejan, no digamos ya en la jerga científica, sino llanamente en el hablar coloquial de los centros de trabajo, indistintamente del género de actividades que en ellos se desarrollen, si atendemos un poco en este sentido no encontraremos con palabras tales como PC, informática, fax, automóviles con computadora de viaje, tarjetas teleónicas, ... etc. con esto podríamos pensar que la idea central es el control computarizado para todo. Verdad es, que esto no es una panacea, y no puede considerarse como la solución a todos nuestros problemas o un boleto de entrada al paraíso, pero si ha sido un factor decisivo para lograr cristalizar en realidades los sueños plasmados en febriles pincelazos por visionarios y extravagantes pensadores y artistas que nos hacían ver esta tecnología como una especie de cuento de hadas.

Si respondemos a la pregunta siguiente: ¿Cuál es uno de los principales puntos de contacto con el mundo tecnológico, la clave de acceso hacia la vida tecnificada, hacia esa vida cuyo mecanismo se resuelve con el sólo accionar un botón? No se necesita echar mano de mucha labor de reflexión antes de encontrarnos con que se trata de la PC, la computadora personal.

Tenemos que las herramientas pueden ser casi innumerables, más resta un aspecto a considerar que es la optimización de las mismas. Su implementación acertada y precisa es saber emplear la tecnología, o sea "hacer mucho más con menos". Esto en términos industriales es una frase sagrada. No podríamos consentir como aceptable un avión F-5 para trasladarnos al supermercado a realizar nuestras compras, así como impensable sería también, el capacitar ampliamente a un grupo de gente en el manejo de un paquete de aplicación cuando las utilidades a largo plazo que del mismo se obtendrán están evaluadas en un índice no mayor al 50% de la inversión en términos de tiempo y costo. Caso este que se presenta con mucha frecuencia en innumerables empresas y representa una substancial pérdida en el capital de producción.



Cabe considerar que en este fenómeno convergen diversas circunstancias que determinan que esta situación se presente.

En esta serie de aspectos encontramos que en general se desconocen a detalle los criterios para la optimización de los recursos tanto de capital como de trabajo y tiempo. Esto es, que en condiciones ideales se deberá capacitar al personal para el uso de la herramienta de programación o software en general, solamente en los aspectos que serán utilizables en la programación. Ya que esto se traducirá en menos tiempo dedicado a la instrucción, lo cual se traduce en mayor tiempo de trabajo, y con ello menos costo de horas hombre en la actividad propia de la empresa.

Una situación bastante frecuente es cuando nuestra área se ocupa de la programación de las aplicaciones. Esta situación por sí sola ya presupone una amplia gama de circunstancias que pormenorizan las condiciones en que se suscitara el problema; de entrada, tenemos que al encontramos en una área en la que la optimización del tiempo invertido en el desarrollo de los sistemas es fundamental y la eficacia en la planeación encaminada a la plena y puntual satisfacción de las necesidades a que son destinados tiene la prioridad principal. En este sentido, se tiene que echar mano de todos los recursos que el ingenio pueda tener para la óptima utilización de la herramienta de programación; lo que implica que debemos tener un amplio conocimiento de la misma que nos permita tomar decisiones en cuanto a la forma en que será utilizada, es decir, del estilo de programación que será tomado como modelo a seguir subsecuentemente para el desarrollo de sistemas. Esto representa un punto determinante para la optimización de la herramienta puesto que tenemos que considerar las condiciones de la herramienta misma.

Esta es una cuestión que tiene que ser cuidadosamente evaluada puesto que será con estos principios que se tendrán que elaborar todos los reglamentos y estándares para facilitar la comprensión del código de programación no solamente por parte del programador que ha realizado la codificación, sino de todo aquel que de alguna manera tenga participación en el proceso de generación del producto. En condiciones ideales debe ser un código que observe ciertas reglas que tenderán a reducir el tiempo de desarrollo a la vez que incrementar la eficacia en cuanto a la magnitud del listado y su funcionalidad requerida. Es aquí donde tenemos que echar mano de los estándares en la programación.

Los estándares en programación se han presentado en numerosos casos con efectos contraproducentes en cuanto a que frecuentemente causan más problemas de los que resuelven, complicando el proceso de desarrollo de un sistema contrariamente a los objetivos de su implementación.

El presente estudio aborda una situación real específica enfocando el lente analítico sobre aspectos particulares y muy diferenciados de cuanto rodea al desarrollo de los sistemas. Se han tomado como campo de estudio las diversas situaciones que se

presentan al tener que elaborar sistemas de bases de datos en el lenguaje de programación Informix-4GL. Se ha elegido este lenguaje debido a que su utilización es muy común y sus características particulares de operación, lo hacen similar a otros lenguajes usados comúnmente, de este modo, haciendo aplicables a otros lenguajes las observaciones que se efectúan a lo largo de este estudio.

El objetivo de este trabajo es hacer un análisis de los factores que coadyuvan al brote de situaciones conflictivas en la aplicación de los estándares en la programación usando Informix-4GL. Primeramente se detectan los elementos que causan estas averías y su interrelación con los procesos de desarrollo de los sistemas, además de las implicaciones y consecuencias que arrastran al generarse aplicaciones en estas condiciones. Posterior a esto, se hace hincapié en las alternativas y sus posibilidades inherentes a las modificaciones que pudieran hacerse con miras a erradicar esta problemática y los criterios que deberán observarse para la toma de decisiones en este sentido. Finalmente, se presenta una secuencia de los puntos que requieren de esmerada atención para obtener una óptica sólida y precisa de cualquier situación que pudiera presentarse al diseñar los estándares más adecuados a las circunstancias que se tengan en un momento dado.

La elaboración de los ejemplos presentados para clarificar las explicaciones han sido escogidos en la inteligencia de ser, en la medida de lo posible, muy generales de manera tal que aquellos diseñadores de sistemas, analistas o programadores que no utilicen Informix-4GL y lo hagan usando algún otro lenguaje, aunque no sea de 4GL, se hallen en posibilidades de aplicar, sin gran dificultad, los criterios y preceptos que aquí se exponen.

## I FUNDAMENTOS TÉCNICOS

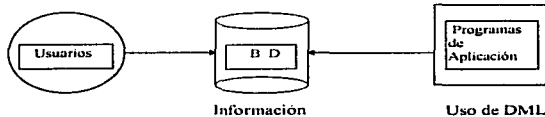
## Fundamentos Técnicos

En este capítulo se exponen los conceptos básicos relacionados con la problemática que se abordará en el presente estudio con las miras de familiarizar al lector con los conceptos fundamentales de las bases de datos; se han considerado desde conceptos básicos como los presentados en el tema de introducción a las bases de datos, hasta conceptos que implican en su estructura el entendimiento de aspectos de mayor profundidad como sucede con la exposición del algebra relacional, a fin de que en un momento dado pueda fungir como texto de consulta. Cabe señalar que debido a las divergencias existentes entre diversos autores versados en el tema, tanto en las clasificaciones como en los conceptos y nomenclaturas de los elementos que forman el ámbito de las Bases de Datos, se han tomado los conceptos que definen con la mayor sencillez y claridad las conceptualizaciones, sin desatender el grado de conveniencia a los objetivos y alcances de este trabajo.

### INTRODUCCION A LAS BASES DE DATOS

En un sentido práctico, podríamos preguntarnos: ¿Qué es una base de datos?

Es un sistema de almacenamiento de datos que cuenta con la característica de poder recuperar, alterar, eliminar o modificar los datos con facilidad en el momento que se requiera.



De aquí se deriva otro concepto que es el de Sistema de Administración de Base de Datos (SABD) que se define como un sistema cuyo propósito general es mantener la información y hacer que esté disponible cuando se le solicite. Un Sistema de Administración de Base de Datos (SABD) como puede apreciarse en la definición, no es un elemento sino un conjunto de ellos, integrados en un sólo sistema. Estos elementos integrantes de un SABD se enumeran a continuación:

1. Información
2. Equipos

3. Programas
4. Usuarios

**1.- Información.** Debe contar con características tales como integración y participación de la información por parte de todos los usuarios.

La información integrada significa que la Base de Datos puede considerarse como una unificación de varios archivos, todos distintos y que elimina en parte cualquier redundancia. Esto a su vez nos descubre otro concepto que es el de la información compartida y se refiere a que los elementos individuales de la información en la BD pueden compartirse entre varios usuarios distintos, en el sentido de que todos ellos pueden tener acceso al mismo elemento de información.

**2.- Equipo.** El procesador o procesadores y la memoria principal es el equipo fundamental para la ejecución del programa del SABD, además de los dispositivos de almacenamiento secundario: cintas, disquetes, etc., sin perder de vista los dispositivos de entrada y salida.

**3.- Programas.** Entre la Base de Datos física y los usuarios del sistema, existe un nivel de programas, como es el manejador de Base de Datos. Este sistema maneja todas las operaciones de manejo de la Base de Datos formuladas por los usuarios.

**4.- Usuarios.** Un usuario de un SABD se define como toda aquella persona que hace uso de la BD o cuando menos interactúa de alguna manera con el sistema ya sea recuperando la información contenida en la BD o proporcionando el mantenimiento necesario para su buen funcionamiento.

En este sentido, se hace necesario definir a cada uno de los diferentes tipos de usuarios que pueden intervenir en un momento dado durante el funcionamiento del sistema. Tenemos que los autores especializados en esta cuestión, muestran grandes divergencias en cuanto a las clasificaciones y denominaciones que proponen, para los objetivos que deseamos cubrir emplearemos nombres que sean descriptivos de la función que desempeña el usuario con respecto del sistema. Finalmente, tenemos definidos cuatro diferentes tipos de usuarios:

- **Programadores de aplicaciones.** Son las personas que trabajan con el sistema mediante llamadas utilizando algún lenguaje de manipulación de datos o DML (por ejemplo C, Pascal, Cobol, Clipper, D-Base), comúnmente estos programas son denominados programas de aplicación.
- **Usuario final.** Este tienen contacto con el sistema sin necesidad de utilizar algún DML, en este caso la comunicación se lleva a cabo mediante programas de aplicación previamente diseñadas por el programador de aplicaciones, de modo tal que sean entendibles por cualquier usuario, facilitándole la interacción con el sistema, modificando, recuperando o ingresando información a la Base de Datos.
- **Administrador de la Base de Datos.** Este usuario interactúa con el sistema atendiendo a todos los aspectos de índole técnica que implican proporcionar los controles que permitan el buen funcionamiento del sistema, como es la constante

actualización de los programas de aplicación de acuerdo con las cambiantes circunstancias que así lo requieran.

- **Administrador de Datos.** Es el personal que se ocupa de establecer la correspondencia entre los datos y los esquemas que se utilizarán para la distribución, clasificación y secuencia conceptual de los mismos.

Además de los puntos presentados, existen otros aspectos que son inherentes a la integración de los sistemas de Bases de Datos:

#### **A. Lenguaje de manipulación de datos**

Según la definición que presenta Henry F. Korth<sup>1</sup>, un Lenguaje de Manipulación de Datos (DML): "es un lenguaje que capacita a acceder o manipular datos según estén organizados por el modelo de datos adecuado". Aquí se puede agregar que un DML es básicamente el lenguaje que instrumentaremos para operar en el SABD agregando, modificando, eliminando y recuperando los datos. Otros autores le denominan Lenguaje anfitrión.

#### **B. Diccionario de Datos**

Un diccionario de datos es un documento que contendrá información acerca de la integración del sistema: terminales asignadas, programas que están implementados y que datos utilizan, distribución y ordenamiento de los datos así como las características constitutivas de los mismos y las prioridades respecto de los usuarios sobre el acceso a la información que se deben observar durante la operación del sistema. En suma puede decirse que es un documento que contiene información sobre la naturaleza de los datos que se almacenan.

#### **C. Entidades y Relaciones**

Con la finalidad de simplificar el proceso de diseño de las Bases de Datos, se ha desarrollado un modelo global y representativo, en un esquema que contenga la descripción de la estructura lógica general de las Bases de Datos.

Una *entidad* en una BD es un objeto cualquiera que existe y puede ser distinguido de otros objetos en la BD, es decir, es un elemento de información en la BD: Secretaria de Salud tiene asignada la clave AAOO, debe considerarse como una entidad de la BD, pues es un elemento único en el universo de la BD.

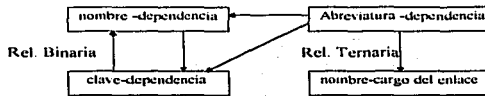
---

<sup>1</sup> Cfr. Korth, Henry F., Silberschatz, Abraham "Fundamentos de Bases de Datos"

Cuando se agrupan un conjunto de entidades con el mismo tipo, se define entonces como un *conjunto de entidades* que, en este caso podrían ser el conjunto de todas las dependencias que tienen una clave asignada.

Existirán además de estas entidades, interacciones o interrelaciones que vincularán a una o a un conjunto de entidades creando una relación entre ellas, o sea que la *relación* es la asociación entre las entidades. Por ejemplo, la relación que existe entre la dependencia Secretaría de Salud y la clave AAOO, se define como una relación que asocia el nombre de la dependencia con la clave AAOO que tiene asignada.

Hay varios tipos de interrelaciones según la forma en que interactúen entre ellas:



#### D. Atributos y Dominios

Si esquematizamos la BD como una tabla que contiene filas y columnas como muestra la gráfica:

Entidad: Dependencia...

Clave	Nombre	Abreviatura
a1	b1	c1
a2	b2	c2
a3	b3	c3

Los *atributos* son las columnas de la BD y, el número de atributos o columnas es el grado de la entidad. Como en este caso, los atributos de la entidad Dependencia son Clave, Nombre y Abreviatura, por tanto, podemos decir que son de tercer grado. El dominio es el conjunto de valores contenidos en una columna independientemente de las filas a las que pertenezcan; por ejemplo el *dominio* del *atributo* nombre es b1, b2, y b3.

Con los elementos hasta ahora presentados hemos cubierto lo referente a la arquitectura fundamental de una base de datos. Ahora bien, en las bases de datos también existen variantes en la manera en que se opera con los datos en ellas, esto es los modelos de datos.

## 1. MODELOS DE DATOS

El modelo de datos es una serie de herramientas conceptuales útiles para describir la naturaleza de los datos en una BD, características tales como las relaciones existentes entre ellos, limitaciones en cuanto a la consistencia de los mismos y de sus relaciones con los demás datos. Según Korth-Silverschatz los diversos modelos de datos más comunes se dividen en tres grupos o modelos lógicos: los basados en objetos, basados en registros y los modelos físicos de datos.

### El Modelo Relacional

El modelo de datos empleado, que ahora nos ocupa, es el modelo relacional. Está clasificado dentro de los modelos basados en registros. G. Gardarin propone una definición breve y concisa de una Base de Datos Relacional: "es un conjunto de esquemas de relaciones y cuyas ocurrencias son unas tuplas de esas relaciones"<sup>2</sup>. Entendiendo por tuplas la correspondencia con la fila de una tabla, así como a la cardinalidad de una tabla, al número de tuplas que esta contenga.

Este modelo de representación consiste en un grupo de tablas con nombre único cada una de ellas y con un número determinado de columnas.

Fecha del evento	Asunto	Ciudad	Seguimiento
06/01/96	Tarifas	83	1
06/01/96	Limpieza	0	1
06/02/96	Electrificación	0	1

Clave	Ciudad	Abreviatura
MROO	83	SEMARNAP
HCFE	0	FIRA
HCFD	0	FIDEC

Esquema general de una Base de Datos Relacional

Una tabla es un elemento constitutivo de una BD relacional que a su vez está integrada por una serie de elementos particulares.

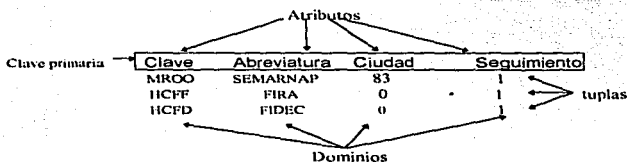
<sup>2</sup> Cfr. Gardarin, Georges "Bases de Datos"



Cada tabla posee un sólo identificador, es decir, una columna o combinación de columnas con la siguiente propiedad: nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas.

Los dominios son grupos de valores de los cuales se extraen los valores reales que aparecen en los atributos.

En la gráfica que se muestra a continuación se presentan los elementos que constituyen una tabla.



Al definir dominios en el medio de las BD relacionales, se sobreentiende que el sistema ofrece recursos mediante los cuales el usuario pueda definir sus propios tipos de datos más complejos.

### Claves

En el diseño de las BD Relacionales es muy importante distinguir los atributos y sus relaciones. Es de primordial importancia, puesto que si bien, por definición los conceptos de unos y otros son distintos, en términos de BD es menester distinguirlos mediante sus atributos. Para satisfacer esta necesidad se han instrumentado el uso de claves la cual es, en primera instancia, un identificador de una tabla o relación.

Una *clave primaria* de una relación, es un identificador único para esa relación. No obstante que también es posible tener una relación con más de un identificador. La *clave primaria* es una clave que es elegida por el diseñador de la Base de Datos de entre las *claves candidatas*. Una característica definitiva en las claves primarias es que ningún elemento de su dominio debe ser repetido. Por ejemplo, en el cuadro anterior, el atributo *Seguimiento* no puede ser clave primaria debido a que su dominio tiene más de una asociación por cada uno de sus valores.

Las *claves candidatas* son un subconjunto de atributos de una relación o tabla que cumplen con una condición que de suyo las convierte en claves candidatas y es que los valores de su dominio sean irrepetibles, que no acepten nulos -de la relación base-, es decir,

que no deben aceptar valor o representación que por convención represente un vacío en el valor de la tupla aplicable. Por tanto, es muy común encontrarnos con tablas que contengan más de una clave candidata. Esta condición de las claves candidatas se le conoce como *condición de unicidad*. Además las claves candidatas deben cumplir con otra condición que es denominada *de minimalidad*, la cual consiste en que no será posible eliminar ningún elemento de la *clave candidata* sin alterar o destruir la propiedad de unicidad.

Podemos afirmar que la relación tiene varias *claves candidatas* de las cuales nosotros escogeremos alguna de ellas como clave primaria y las demás se definen como claves alternativas.

Es preciso aclarar que los conceptos de clave alternativa y clave candidata son conceptos secundarios, pues la esencia reside en la clave primaria, que es la que nos auxiliará en el direccionamiento de alguna tupla en especial por algún valor en ella referenciado, se le ha atribuido esta función dado su carácter de unicidad.

## 2. ALGEBRA RELACIONAL

El álgebra relacional es un grupo de operaciones que se efectúan con las relaciones, es decir, que las relaciones son los operandos de estas operaciones produciendo otra relación nueva como resultado, de este modo cumpliendo con la propiedad algebraica de la cerradura. Estas operaciones se realizan generalmente utilizando lenguajes de alto nivel como el SQL que efectúa operaciones similares a las del Álgebra Relacional.

### Operaciones del Álgebra Relacional

Las operaciones fundamentales del AL se realizan utilizando *operadores* que son similares a los utilizados en la teoría de conjuntos y, las operaciones especiales que comprenden cuatro de su tipo: SELECCION, PROYECCION, REUNION y DIVISION.

### Operaciones fundamentales del Álgebra Relacional

Las operaciones fundamentales son similares a las usadas en teoría de conjuntos: UNION, INTERSECCION, DIFERENCIA Y PRODUCTO CARTECIANO. Todas a excepción del PRODUCTO CARTECIANO requieren que las tablas que fungirán como operandos sean del mismo grado y además sean compatibles respecto de la unión.

#### Unión

Si imaginamos que cada una de las relaciones es un conjunto, y tenemos un conjunto A y otro conjunto B, podemos apreciar fácilmente que la unión sería creación de una nueva relación que contenga todas las tuplas que pertenecen a A y todas las que pertenecen a B o que pertenecen a ambas.

Ejemplo:

Si tenemos las relaciones A y B

A	Clave	Num-com	Abreviatura
	MROO	83	SEMARNAP
	MROO	80	SEMARNAP
	HCFF	0	SEMARNAP

B	Clave	Num-com	Abreviatura
	MROO	80	SEMARNAP
	HCDF	80	FIRA

Nota: Obsérvese que ambas relaciones son homogéneas.

Entonces la unión creará una nueva relación.

A UNION B

Clave	Num-com	Abreviatura
MROO	83	SEMARNAP
MROO	80	SEMARNAP
HCFF	0	SEMARNAP
HCDF	80	FIRA

Debe observarse que tanto en estas relaciones como en los conjuntos no existen tuplas repetidas.

**Intersección**

En esta operación se consideran las tuplas que pertenecen a ambas relaciones a la vez.

Ejemplo: A INTERSECCION B

Clave	Num-com	Abreviatura
MROO	80	SEMARNAP

**Diferencia**

La diferencia se efectúa tomando en cuenta el orden en que se ha de realizar la orden pues A-B no es de ninguna manera igual a B-A y esto se refiere a que en el primer caso, A-B tenemos que tomar las tuplas que pertenezcan a A y no pertenezcan a B y por el contrario en el caso de B-A, deberemos tomar las tuplas que pertenezcan a B y no pertenezcan a A.

Ejemplo: Si tenemos las relaciones A y B y obtenemos A-B y también B-A:

A MENOS B

Clave	Num-com	Abreviatura
MROO	83	SENIARNAP
HICFF	0	SEMARNAP

B MENOS A

Clave	Num-com	Abreviatura
HICDF	80	FIRA

### Producto Cartesiano

Esta operación no es la excepción en cuanto a la similitud con las operaciones de la teoría de conjuntos. En el producto cartesiano la operación  $A * B$  es A veces B, o sea, que cada tupla de A debe ser relacionada con B tantas veces como tuplas contenga B y cada ocurrencia en este sentido generará una nueva tupla en la nueva relación que surgirá de esta operación.

Se considera que al igual que en el álgebra, en la operación de producto cartesiano, el orden de los factores no altera el producto y, además, que la nueva relación tendrá como grado, el resultado de la suma de los grados de cada una de las sumas que sean los operandos.

Ejemplo: Tenemos las relaciones C y D y deseamos obtener  $C * D$

C

Demnto.
SA
SB
SC

D

Evento
1
2

C VECES D

Demnto.	Evento
SA	1
SA	2
SB	1
SB	2
SC	1
SC	2

### Operaciones especiales del Álgebra Relacional

Tomaremos como referencia para nuestros ejemplos subsecuentes y, para fines meramente explicativos, una Base de Datos Relacional imaginaria, de las actividades de un funcionario público durante su periodo de gestión ordinaria, constituida por tres tablas o relaciones que a continuación se muestran.

## ASUNTO

NUMASU	IDENTASU	DIA	UBICACIÓN
SEG1	FIDEC	5	AL
SEG2	DGCSL	15	EL
SEG3	FERRONALES	12	LU
SEG4	FERRONALES	5	AL
SEG5	PFCYP	12	EL
SEG6	ASA	5	AL

## COMPROMISO

FILA	NUMASU	HORAS
A1	SEG1	2
A1	SEG2	2
A1	SEG3	4
A1	SEG4	2
A1	SEG5	1
A1	SEG6	1
A2	SEG1	3
A2	SEG2	4
A3	SEG2	2
A4	SEG2	2
A4	SEG4	3
A4	SEG5	4

## DEPENDENCIA

FILA	INICIALES	CLAVE ASUNTO	UBICACION
A1	UACO	2	AL
A2	CONO	1	EL
A3	C'TAO	3	LU
A4	COSO	2	AL
A5	NARRO	3	SAN

## Selección



2	AL
1	EL
3	EL
3	SAN

PROYECT1 es el nombre de la relación generada, en la cual se han seleccionado las columnas CLAVEASNT0 y UBICACION las cuales son las únicas que aparecen en la nueva relación y también se ha eliminado una de las tuplas que contienen los datos (2,AL), esta eliminación se ha efectuado en virtud de que no se repita esta tupla en la nueva relación. En el segundo ejemplo, observamos una variación de la flexibilidad de la operación Proyección ahora con tres atributos seleccionados.

PROYECT2 = ASUNTO (IDENTASU, DIA, UBICACION)

PROYECT2

IDENTASU	DIA	UBICACIÓN
FIDEC	5	AL
DGCSL	15	EL
FERRONALES	12	LU
FERRONALES	5	AL
PFCYP	12	EL
ASA	5	AL

### Reunión

Para que esta operación sea factible, es necesario e indispensable contar con dos relaciones que tengan por lo menos una columna o atributo con dominio común. Estas dos relaciones, que llamaremos "A" a la primera y "B" a la segunda de ellas, son las que utilizaremos en nuestra reunión para formar una nueva relación que será "C", mediante una conexión entre las tuplas que posean el mismo valor en el atributo que tienen en común, característica que técnicamente es denominada (EQUIREUNION).

En el ejemplo que a continuación se presenta, las relaciones utilizadas son ASUNTO Y DEPENDENCIA que poseen un atributo en común que es UBICACIÓN. Este atributo es el que fungirá como conexión entre las dos relaciones para formar la nueva relación que llamaremos GENERADA.

Empleando las relaciones ASUNTO Y DEPENDENCIA obtendremos la  
REUNIÓN = SOBRE UBICACION

FILA	INICIALES	CLAVE ASUNTO	UBICACION	NUMASU	IDENTASU	DIA
A1	UACO	2	AL	SEG1	FIDEC	5
A1	UACO	2	AL	SEG4	FERRONALES	5
A2	CONO	1	EL	SEG2	DGC'SL	15
A3	CTAO	3	I.U	SEG3	FERRONALES	12

Es pertinente destacar que la ubicación SAN no aparece en la nueva relación y esto es porque este dato no se encontraba en ambas relaciones.

Nota: Es permisible también que se emplee la siguiente notación para expresar la operación reunión.

DEPENDENCIA (UBICACIÓN = UBICACIÓN) ASUNTO

### División

Formalmente, podemos tomar la definición que propone Gardarin: " División .- El cociente resultante de dividir la relación R de esquema R(A1, A2,... An ) entre la relación S de esquema S(Ap+1, ... An) es la relación Q de esquema Q(A1, A2,... An) formada por todas las tuplas que concatenadas con cada uno de las tuplas de S resulta siempre una tupla de R." La notación utilizada es R DIVIDIR ENTRE S O R/S.

La gráfica siguiente presenta la relación A, B1 y B2 y se desea obtener la A DIVIDIR ENTRE B1 y A/B2:

A	CLASE	NUMASU	B1	NUMASU	A/B1	CLASE
	C1	SEG1		SEG1		C1
	C1	SEG2				C2
	C1	SEG3				
	C1	SEG4				
	C1	SEG5				
	C1	SEG6				
	C2	SEG1				
	C2	SEG3				
	C2	SEG4				
	C2	SEG5				
	C2	SEG6				
			B2	NUMASU	A/B2	CLASE
				SEG2		C1
				SEG3		

Dividiendo la relación A entre la relación B1 se ha obtenido la relación A/B1 la cual contiene C1 y C2 como dominio del atributo clase. Cada uno de los elementos de este dominio, si se concatenan con los elementos de la de la relación B1 siempre darán como resultado una tupla de la relación A. Por ejemplo: Si se toma C1 del atributo CLASE de la relación A/B1 y lo concatenamos con cualquier elemento de la relación B1 como pudiera ser



SEG1, obtendríamos (C1, SEG1) el cual corresponde con uno de los elementos de la relación A. Del mismo modo, ocurrirá con el elemento C2.

Nótese que en la relación A/B2 no aparece en el dominio C2, puesto que si hacemos la concatenación con SEG2 tendremos el par (C2, SEG2) resultando esta una tupla que no pertenece a la relación A, y en cambio, la concatenación que se haga con C1 y cualquier elemento de B2 indefectiblemente tendremos como resultado una tupla de la relación A.

### ***EJEMPLO DE LA APLICACIÓN DEL ALGEBRA RELACIONAL A LA CONSULTA DE INFORMACIÓN A UNA BASE DE DATOS RELACIONAL.***

Cuando se opta por utilizar una Base de Datos, esto se hace con el fin de almacenar información de modo tal que en un momento dado tengamos disponibles con facilidad cualquier dato que en otras circunstancias nos resultaría muy tardado recuperar. Esta labor de recuperación puede ser relativamente difícil y tardada, aún en Bases de Datos bien integradas, es por esto que la utilización de el Algebra Relacional se torna fundamental en el manejo de un Sistema de Administración de Base de Datos.

De modo ilustrativo, se presenta un ejemplo que ilustra el poder que puede tener el Algebra Relacional en la recuperación de datos en las Bases de Datos Relacionales.

*Sea la Base de Datos con las tablas ASUNTO, COMPROMISO Y DEPENDENCIA de las cuales se desea saber las iniciales de las dependencias que no tengan asunto SEG2.*

Para resolver este tipo de problemas siempre existen varios métodos, que si bien pueden ser muy distintos, también cumplen con su cometido obteniendo los mismos resultados. Aquí la diferencia estriba en lo largo o embrollado del camino que se siga para lograr acceder el dato requerido y también en la naturaleza de las circunstancias del problema, es decir, no siempre se resuelve un problema de consulta a una Base de Datos simplemente obteniendo la información requerida por cualquier camino, sino a veces hay que tomar en cuenta cosas como la velocidad del equipo con que contamos, el volumen de la información que manejamos, el número de datos que estimamos recuperar, etc... en estos aspectos es donde se toma el criterio para decidir si un determinado método o proceso es el más adecuado o no para la solución de un problema dado.

1.- Considerando que las dependencias tienen varios asuntos simultáneamente, tomaremos la relación COMPROMISO y buscaremos cuales son las dependencias que SI tienen asuntos SEG2.

SELECCIONAR de: COMPROMISO DONDE NUMASU = "SEG2"

**M1**

FILA	NUMASU	HORAS
A1	SEG2	2
A2	SEG2	4
A3	SEG2	2
A4	SEG2	2

2.- Si reunimos M1 con DEPENDENCIA SOBRE FILA = FILA, denotaríamos

M1 { FILA = FILA } DEPENDENCIA y tendríamos el siguiente resultado:

**M2**

FILA	NUMASU	HORAS	INICIALES	CLAVEASNT0	UBICACION
A1	SEG2	2	UACO	2	AL
A2	SEG2	4	CONO	1	EL
A3	SEG2	2	CTAO	3	LU
A4	SEG2	2	COSO	2	AL

3.- Ahora proyectamos M2 para separar FILA , INICIALES, CLAVEASNT0 Y UBICACIÓN.

M3 =!! M2 {FILA, INICIALES, CLAVEASNT0, UBICACIÓN}

**M3**

FILA	INICIALES	CLAVEASNT0	UBICACION
A1	UACO	2	AL
A2	CONO	1	EL
A3	CTAO	3	LU
A4	COSO	2	AL

4.- Restar DEPENDENCIA - M3 para presentar la relación de dependencias que no tienen asunto SEG2.

M4 = DEPENDENCIA - M3

**M4**

FILA	INICIALES	CLAVE ASNT0	UBICACION
A5	NARRO	3	SAN

5.- Finalmente, para obtener las iniciales de las dependencias que no tienen asunto SEG2 hacemos una selección de M4

M5 =!! M4 {INICIALES} y así obtenemos la relación pedida:

M5

INICIALES
NARRO

## II FRENOS

## Frenos

### *¿Que son los frenos?*

Según la definición clásica, en su sentido físico, un freno es un dispositivo que reduce la velocidad de funcionamiento o detiene una máquina.

Existen máquinas en las que no necesariamente deben tener un sólo freno para funcionar, pueden poseer varios y con varias funciones, o sea que se pueden detener varios componentes de una misma máquina sin tener que detenerla por completo. Así mismo, suele suceder que repentinamente funcionen frenos que son indeseables en un momento determinado, obstaculizando un proceso o trabajo. Para esto, se llevan a cabo acciones que corrijan o regulen el funcionamiento de este freno indeseable, e incluso, es posible, en algunos casos y, en algunas máquinas, realizar ajustes que optimicen la velocidad y el rendimiento de dicha máquina, acelerando el proceso de trabajo para el cual se le ha destinado.

En este sentido, cabe hacer una analogía que será muy útil para definir los problemas y situaciones que se suceden en un equipo de trabajo destinado a fungir como el área encargada de crear los sistemas de software en un modelo de empresa más o menos general. Esta analogía esta fundamentada en el hecho de que si bien, el departamento de sistemas no es enteramente una máquina debido al factor humano que interviene en ella, si implica factores que les son enteramente indistintos, como son el hecho de que si hay un decremento en el rendimiento, la causa o el punto decisivo para que esto suceda no solamente se puede encontrar en un punto único e invariable, esto debido a la naturaleza compleja del proceso mismo por el que atraviesa una idea para cristalizarse en un producto de software presentado al usuario

En el hecho de que existan muchos y muy diversos componentes que participan en el proceso de generación de un producto de software, la participación misma del factor humano puede llegar a erigirse en un problema muy complejo en si mismo. A la par que una máquina integrada por mecanismos complejos, el área de sistemas necesita también una constante revisión que permita un monitoreo de las actividades y cómo éstas se efectúan, del mismo modo, para saber en qué momento es pertinente proporcionar mantenimiento adecuado tanto al proceso mismo como a los equipos utilizados. En suma, la principal ventaja de esta analogía es que permite una visión clara del aspecto de conjunto del problema que abordaremos, facilita, por tanto, la concepción de las posibles soluciones que puede tener un problema determinado y minimizan los posibles extravíos y digresiones que suelen darse en los procedimientos de investigación.

Postularé entonces, este concepto en el sentido antes señalado.

### **Cómo y dónde operan los frenos**

Como ya se puntualizó más arriba, las causas que son susceptibles de frenar indeseablemente un proceso, se pueden localizar en innumerables puntos del mismo. Muchas veces, si no la mayoría de ellas, localizar el punto donde se suceden estos problemas no es en sí, el lugar desde donde deben atacarse, es decir, es práctica común en las áreas de sistemas, que se ataquen los síntomas más que el mal mismo. Por ejemplo, si en una área de sistemas cualquiera, se detecta una baja del rendimiento, debido a causas no del todo claras, en un departamento determinado, y se procede a investigar la causa y se encuentra que simplemente hay un decremento en la estadística de proyectos terminados, sin más detalles, suele suponerse que esto se remediará reajustando la política de relaciones con el personal de nivel operativo, lo cual se efectuará tal vez estableciendo una vía de comunicación en la cual se le participe al personal de lo apremiante y trascendente que resulta para la dirección la entrega puntual de los proyectos encomendados. Si es que no se llega a tomar alguna medida punitiva en el peor de los casos, lo cual acarrearía muchas y muy variadas dificultades acorto, mediano y a largo plazo.

Ahora bien, esta medida -en el caso menos drástico-, tal vez tenga un efecto positivo a corto plazo, pero a la larga puede acarrear efectos secundarios que por lo general suelen ser más difíciles de detectar y resolver. Estos efectos pueden originarse debido a que las relaciones entre el personal que labora en esa área se modifican leve, paulatina y negativamente, implicando después, un ambiente de displicencia hacia la actividad misma de su trabajo. En esta circunstancia, no es difícil que se generen situaciones en las que las relaciones entre el personal se tensen de tal manera que en un momento determinado alguna persona decida dar por terminada la relación laboral con la empresa. Lo cual implica tener que contratar a un nuevo integrante del equipo, mismo en el que tendrá que invertirse insumos de tiempo para capacitarlo y ambientarlo en el trabajo antes que de inicio su período productivo. A su vez esto traerá como consecuencia un inevitable aminoramiento en la producción y descontrol en cuanto a la distribución de actividades que se reflejarán en la calidad de los productos si no es que en lo tardío de su generación. Desde luego, no es esto una regla o ley invariable que tenga que acaecer indefectiblemente, no se debe dejar de considerar la naturaleza misma de como está integrado y quienes constituyen el equipo de trabajo.

Al mismo tiempo pueden suceder problemas con el equipo y sugerirse a la dirección administrativa que proporcione las herramientas más sofisticadas que se encuentren en ese momento en el mercado, cuando muy posiblemente no sea esta la mejor solución.

Esta serie de problemas y el cómo se procede a combatirlos, obedece fundamentalmente a una óptica un tanto cuanto distorsionada de la visión global del cometido específico del área de sistemas, a su vez, esta distorsión es generada debido a aspectos no considerados con toda la acuciosidad debida. Hago alusión a los estos aspectos referidos en los puntos siguientes:

### **Claridad en el objetivo de la actividad del área en general**

Esto significa que se debe conocer a quienes se destinan los productos, es decir, las características de trabajo del usuario final o área a la que se otorga el servicio, la naturaleza de la información y, consultarle acerca de su actividad y de cómo podría ser más sencillo, eficiente y cómodo su trabajo. Esta exploración, frecuentemente solo se reduce a saber "¿qué es lo que el usuario necesita?" y se deja de considerar el cómo lo necesita y cual es la filosofía a seguir para el diseño basada en las necesidades del usuario, filosofía que se basa generalmente en las circunstancias mismas del área y administración del departamento de sistemas. Otro aspecto también usualmente olvidado es el criterio que otorga la visión global del departamento al que ese producto será destinado, que acarrea problemas tan diversos como impredecibles, que gradualmente deterioran el óptimo funcionamiento general del área hasta el extremo de tratar de adaptar las necesidades del usuario a las condiciones operativas del departamento de sistemas.

### **Orientación claramente definida en la planeación y dirección de las actividades.**

La consecuencia inmediata de una carencia de claridad en la concepción de los objetivos es una administración ineficiente de los recursos empleados para la labor de integración de sistemas. Entre estos recursos se enumeran:

- a) **Recursos humanos.** Es práctica común en estos casos el que se le asigne a un jefe departamental un proyecto con determinadas características no muy bien definidas, en el sentido de que se aclaran las características de funcionamiento pero no se mencionan ningún momento los aspectos referentes a los patrones requeridos en la interacción con el usuario, es decir, los aspectos de rendimiento, dejándose al libre arbitrio de los diseñadores directos del sistema, entre los que se cuentan a los programadores mismos, los cuales se hallan en la situación de modificar hasta decenas de veces el código de sus programas, después de que estos se declaran formalmente "terminados". Este trabajo extra, implica desde luego el esfuerzo extra también de los programadores lo cual acarrea extralimitarse en el tiempo invertido en el desarrollo de un sistema y por tanto un gasto extra en los salarios del personal.
- b) **Insumos económicos.** En estas condiciones el tiempo de desarrollo de un sistema se prolonga y por tanto incrementa el consumo de energía eléctrica ocupada por las terminales usadas para tal efecto, el cual se traduce también en un repunte notable en el índice del costo económico por sistema realizado. Aunado a este costo, también debe considerarse que el mismo efecto ocurre con el costo en cuanto al salario de los programadores por sistema realizado.<sup>1</sup>

---

<sup>1</sup> Cabe aclarar que se considera que el personal operativo del área de sistemas devenga un salario fijo por días trabajados y no a destajo o por sistema realizado.

**El programador debe conocer los objetivos particulares y generales del sistema que se le asigna programar.**

Generalmente a los programadores sólo se les especifica como es el programa que se les requiere y cual debe ser su función. El programador, entonces, realiza la codificación necesaria para que el producto cumpla con los requerimientos indicados. Finalmente el programa, posterior a una "aprobación por la jefatura del área" es presentado al usuario, que al ponerlo en práctica encuentra que el programa pedido tiene algunos detalles que no se tomaron en cuenta al momento de solicitar el software y que no son adecuados para el trabajo que tiene que realizar el programa. Luego, el producto es restituido al área de sistemas para que realice las modificaciones dando inicio a un tedioso proceso de revisión, modificación, remodelación y pruebas, en el que el final de dicho proceso nunca llega a ser muy claro.

El programador por su parte, corre el riesgo de caer en un embotamiento causado por el tedio y la frustración que implica tener que revisar y modificar un programa que una vez puesto en marcha, no pasará mucho tiempo antes que lo vuelva a remodelar, creándose una situación similar al suplicio de Sísifo<sup>2</sup>. Esto pues, llega a ser motivo más que suficiente para que el programador experimente una sensación de frustración susceptible de vertirse en un franco disgusto por la actividad realizada.

Este punto es especialmente importante debido a que, según las actuales tendencias de la filosofía de la calidad, es imprescindible que el personal que labora en una empresa, no trabaje por que tenga que hacerlo o se sienta obligado a ello sino que desee hacerlo y sienta gusto por su actividad y el reconocimiento de los demás por el esfuerzo realizado.

Además de las dificultades expuestas más arriba, las consecuencias a largo plazo suelen ser muy dañinas en el sentido de que esta situación dificulta y disminuye las posibilidades de efectuar alguna expansión del área de sistemas a cubrir más necesidades de otras oficinas o de dedicar tiempo para la evaluación de algún proyecto que tienda a ampliar la cobertura del departamento de sistemas.

### **Los Estándares**

Los estándares son un recurso frecuentemente empleado para facilitar el trabajo y acelerar el proceso de diseño e implementación de los sistemas. Contribuyen a la claridad en los códigos fuente de los programas, permitiendo que estos puedan ser entendidos con facilidad por todo el personal implicado en la labor de diseño y aprobación de los sistemas. Funcionan como un guía muy importante al buscar un elemento específico que se encuentra dentro de

<sup>2</sup> Sísifo, personaje de la mitología griega, habitaba en los infiernos, el cual estaba condenado a empujar una enorme piedra hacia la cumbre de un monte, cuando llega cerca de la misma, el peñasco rueda en la llanura, y Sísifo vuelve a empujarlo nuevamente hacia la cumbre.



la intrincada red de directorios y subdirectorios que integran el sistema de archivos. En términos generales, puede decirse que los estándares son el orden de cosas que observan todos los elementos de los sistemas, que van de los códigos fuente a la presentación comercial del producto, del árbol de directorios dentro del ambiente de programación a los lineamientos que se deben respetar para asignar los nombres a las bases de datos, de los estilos de programación al tipo de monitores utilizados para la elaboración de las líneas de código, y hasta los microprocesadores utilizados para correr las pruebas que sondean la calidad del producto, en suma, los estándares son las leyes que definen el orden de cosas competente a los productos del área de sistemas, desde el inicio de la concepción de la idea para la creación de un programa hasta la puesta en marcha de mismo en lo que va a ser su campo de operación.

Estos son -en teoría- los objetivos de la instrumentación de los estándares. Objetivos que por otra parte, no siempre se convierten en realidades tangibles y que si en un principio, aparentemente rinden algunas mejoras en los procesos, con el tiempo estas mejoras se van tornando en inconvenientes considerables que los deterioran y se constituyen en francos obstáculos para el funcionamiento adecuado del departamento de sistemas.

Los problemas causados por la adopción de los estándares son debidos a la falta de un criterio adecuado para su instrumentación en el campo de trabajo. Los estándares aplicados en forma inadecuada causan siempre más problemas de los que se tendrían si se hubiera optado por no aplicarlos.

La aplicación de los estándares, en condiciones ideales, y no en forma similar a como se realiza en muchas otros campos de la actividad humana, se debe decidir en la fase inicial de la planeación del área que será destinada al diseño de sistemas. Se establecen reglas que definen las nomenclaturas de todos los archivos, los nombres de los directorios y su ubicación dentro del árbol de directorios y los archivos que estos contendrán; asimismo, se establecen los formatos que deberán tener los nombres de las variables utilizadas en los códigos fuente de los sistemas y los equipos específicos que serán usados en cada una de las etapas por las que atraviesa el desarrollo de un sistema. Esto incluye desde luego, la planeación misma de las jerarquías que serán establecidas en el lugar de trabajo, con el fin de afianzar, por una parte, la adecuada distribución de las cargas de trabajo y por otra, la factibilidad de un monitoreo eficiente y claro de las actividades que se realicen, en la inteligencia de que, en el caso de que surjan problemas en alguna parte del proceso, estos puedan ser detectados con la mayor oportunidad y antelación posible para facilitar su solución.

Para el caso particular que analizamos en el presente trabajo, -la programación en Informix-4GL- es una práctica muy socorrida que en las situaciones en que se tengan que desarrollar sistemas continuamente, con algunas características distintivas comunes a la mayoría de los programas que se realizan en dicho departamento y, debido a las características modulares del lenguaje Informix-4GL -y algunos otros lenguajes en los que no sólo los considerados como 4GL se podrían incluir-, permiten el recurso de crear librerías propias con funciones que los mismos programadores pueden diseñar, perfectamente adaptadas a las necesidades de cada tipo de programa que pueda generarse. Para este tipo de

funciones también se crean estándares que cubren el aspecto de las nomenclaturas para las funciones y, designan cuáles son las funciones que ameritan ser incluidas en los archivos de librerías, posterior a una evaluación. La función más importante que tienen estas medidas son la de anular las dimensiones de la codificación ya de por sí extensa, en el caso de los programas para bases de datos relacionales y de hacer más inteligible el código fuente para las diferentes personas participantes en el desarrollo del sistema en cuestión.

#### **Consecuencias de la inadecuada instrumentación de los estándares.**

Los problemas generados a causa de una implementación inadecuada de los estándares son muchos y muy variados en su naturaleza, tanto por sus consecuencias como por los medios en que se sitúan y la diversidad de sus múltiples y catastróficos efectos.

En general, los efectos más notables e inmediatos causados por la incorrecta adopción de los estándares, son reflejados en una lamentable pérdida de tiempo en resolver problemas que en otras circunstancias no tendrían razón de existir, tiempo que en su aspecto económico y, por tanto comercial, es de vital importancia hasta el grado de que esta ineficiencia puede ser decisiva para la existencia del departamento de sistemas dentro de la empresa, si no es que de la existencia de la empresa misma, en casos en que desafortunadamente no podemos calificar de muy extremos o inusuales. Los problemas más habituales que se suscitan en estas circunstancias, manifiestan sus efectos inmediatos principalmente en los niveles operativos, que son los ejecutores de las ideas concebidas en las áreas de planeación, que son las últimas en ser afectadas directamente por las averías procedentes del nivel operativo, no desde luego, sin considerar que el nivel directivo resiente los efectos con alguna diferencia de tiempo que puede ser o no significativa según los procedimientos implementados para la creación de los sistemas.

#### **• Capacitación al nuevo personal que se integra al equipo de trabajo.**

Para que un nuevo integrante dentro del equipo de trabajo comience a prestar sus servicios como parte activa en la generación de sistemas, primero es necesario imponerlo de todas las circunstancias en que el trabajo que se le asignará tendrá que ser desarrollado. Esto es, presentarle los lineamientos, estándares y normas que rigen el funcionamiento del departamento de sistemas en general, y, en particular, las especificaciones dentro de las cuales se desenvolverá su injerencia en el universo de actividades que dependen del puesto en el que ha sido asignado. Es todo un proceso en sí, en que el nuevo integrante deberá ambientarse con el medio, familiarizarse con todos los adminículos implícitos en el campo de trabajo.

Este "poner al corriente" implica tiempo, implica también una inversión cuya recuperación deberá ser a largo plazo y, durante el tiempo destinado a esta adaptación, el

área se encontrará en un lapso de cambio constante, que a su vez implica ajustes y reajustes en algunas de sus actividades.

Es muy frecuente que en este proceso del aprendizaje de los estándares de programación que tiene que aprender un programador para comenzar a trabajar, sea necesario invertir un tiempo extraordinariamente prolongado, que difícilmente tendrá una compensación con los beneficios que reporte la utilidad de su aplicación. En el lenguaje Informix-4GL es usual la creación de archivos que contienen funciones "estándar" creadas en base a las necesidades de los sistemas que se generen con cierta frecuencia. Utilizar un gran número de funciones almacenadas en estos archivos de librerías presupone la utilización de estándares para nomenclatura, en primera instancia de los archivos que contienen tales funciones y luego el nombre de las funciones incluídas en ellos. Es ahora cuando aparece la cuestión: ¿cuanto tiempo tardará en ser aprendida toda esta serie de nomenclaturas? ¿Que tan útil es invertir esfuerzo en conocerlas todas y saber su forma de operar? Todas estas cuestiones son consideradas solo en el momento inicial de la planeación en que se medita la disyuntiva de decidir si se recurrirá a esta alternativa para "optimizar la labor de programación". En las primeras etapas de su utilización, es cuando se crean las funciones que se consideran mas frecuentes, siendo este el único criterio valdadero para su evaluación. Posteriormente y, con la misma balanza se van evaluando y agregando mas funciones que paulatinamente se acumulan en tales archivos hasta hacerlos tan extensos e intrincados que se hace casi inoperante desplegarlos en el monitor. Además de retardar la compilación de los programas que las utilizan, esto crea la necesidad de desarrollar profijos manuales que expliquen su utilización. Dando lugar, todo esto a un consumo de tiempo insoportado que deteriora enormemente el cumplimiento del cometido principal en el área de sistemas. En consecuencia tenemos que destinar un considerable porcentaje de recursos para ingeniar un curso completo de capacitación al nuevo personal, también para el personal que impartirá la instrucción, espacio físico en que se usará en tal efecto, tiempo que requerirá la habilitación del nuevo personal, etc. todo esto para obtener un beneficio incierto y bastante cuestionable.

#### • Jerarquías

Las jerarquías establecidas entre el personal, como antes se ha marcado, se establecen en base a las necesidades de cada núcleo de trabajo. Aquí el problema consiste en que, en un momento dado en que la situación creada por toda la serie de dificultades que arrastra la inadecuada aplicación de los estándares, afecta a las jerarquías mismas que llegan ha ser ineficientes y muchas veces obligan a actuar, tanto al personal operativo como al directivo, a ignorar toda convención de este tipo violando las reglas y dando lugar a inesperadas eventualidades que conducen al caos. Es cuando, por parte de los directivos se pierde el control de las actividades, se generan situaciones en que cada una de las personas en el área se ve obligada a echar mano de su criterio personal en todo tipo de situaciones, en la mayoría de los casos desentendiéndose de los objetivos fundamentales de la dirección. Y cuando las dificultades llegan al punto en que hay que resolver los problemas que causaron los problemas del problema de la aplicación de los estándares, entender el problema resulta más difícil que fundamentar matemáticamente la certeza del descubrimiento del último teorema de Fermat. Una situación verdaderamente insufrible.

**Formatos para presentaciones de pantallas a los usuarios.**

Siendo el diseño de interfaces usuario-sistema una exigencia tan común y de notable importancia en el mundo de la informática, pues debido a la gran variedad de situaciones en que se presenta el problema de la compatibilidad entre los productos de software, se ha considerado la estandarización como norma obligatoria y necesaria. Se pretenden estandarizaciones que cubran sin excepción todos los elementos que tengan alguna injerencia en la creación de sistemas, donde los diseños de las pantallas presentadas al usuario y a los programadores no gozan de exoneración alguna.

Para este caso en particular, mostraremos un modelo más o menos generalizado de pantalla para la presentación de un reporte, establecido como estándar para la elaboración de sistemas de Bases de Datos Relacionales.

XXXXXXXX (a) TIPO DE TRABAJO: XXX	NOMBRE DE LA INSTITUCIÓN SUBDIRECCIÓN NOMBRE DEL PROGRAMA	(b) DD/MM/AA (c) HH:MM:SS (d) PAG. XXX
<b>CODIGO</b>	<b>DESCRIPCIÓN</b>	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
XXX	XXXXXXXXXX	
TOTAL: XXX		

- Nombre del archivo en que se encuentra el código fuente del reporte corriente.
- Fecha de la generación del reporte.
- Hora en que se ha liberado el reporte.
- Página del reporte presentado en pantalla.

Este formato de reporte bien podría ser adecuado en ciertas circunstancias, pero también, podrá no serlo en otras, es decir, para cada situación diferente cabe también, una solución diferente. Este estándar fue tomado para presentar un reporte con los datos que el usuario pudiera necesitar de cualquier sección de la base de datos en un momento dado.

Lo importante en este diseño es saber si los creadores del mismo tomaron en cuenta las necesidades del usuario dichas por el mismo, si han tomado asiento junto al futuro usuario frente al monitor y observar su ambiente de trabajo, las condiciones en que el mismo debe desarrollarse. Pues existen múltiples situaciones que el diseñador no puede prever y que el usuario experimenta con frecuencia. Por ejemplo, para el análisis de este diseño cabe cuestionarse que tal vez el usuario le sea suficiente que los datos del encabezado sólo aparezcan en la primera pantalla con la finalidad de satisfacer la fundamental necesidad - debido a las condiciones de su actividad- de visualizar la mayor cantidad posible de datos en una sola pantalla. O bien, puede investigarse si realmente es conveniente agregar el dato de la hora con un ciclo de actualización en la consulta al reloj del sistema, o si por el contrario, es mejor suprimirlo, dadas las capacidades del procesador que manejan los servidores o tal vez considerando el número de terminales que se hallen operando simultáneamente durante la elaboración del reporte o tal vez el número de terminales que se hallen trabajando con este mismo programa; después de todas estas consideraciones es posible que resulte que el proceso de refresco retarda significativamente los tiempos de respuesta de los servidores.

Evidentemente, existen grandes posibilidades de que algunos de estos aspectos referentes a la creación e implementación de los estándares en la programación, no considerados para el diseño de los formatos sea contraproducente y dificulte la función para la que fueron creados.

### **Nomenclaturas**

Las nomenclaturas de archivos, programas, variables en los códigos fuente, las bases de datos, tablas, reportes, etc. se implementan con el fin de presentar información acerca del contenido del archivo, ubicación en el árbol de directorios y utilización del mismo, en la cadena de caracteres que se les ha asignado como nombre. Y esto suele hacerse con este único criterio produciendo resultados igualmente nocivos que en el caso de los estándares para los formatos de interface usuario-sistema. No se consideran, por ejemplo, el número de bases de datos existentes o de las tablas y por tanto, si conviene crear, dado ese número de elementos, una nomenclatura especial que deberá ser considerada en el ahora voluminoso manual de estándares.

En estas nomenclaturas, desde luego, se incluyen las pertenecientes a los directorios en que se ubican los diferentes archivos, que muchas veces, no obstante que es suficiente con establecerlas para eliminar la necesidad de estándares en los archivos que contienen, se elabora irreflexivamente otra trama intrincada de normas para sus denominaciones

### **III DETONADORES**

## Detonadores

En el sentido estricto del término, un detonador es un artefacto capaz de hacer detonar un explosivo. En nuestro estudio, y continuando con la analogía antes establecida, el detonador sería el conjunto de todas aquellas condiciones que insensiblemente conducen a inevitables deficiencias en el área de sistemas de una empresa, produciendo una incontrolable catástrofe económica que se traduce en la quiebra de la empresa a quien se brinda el servicio. En fin, el explosivo encuentra analogía con la serie de efectos negativos que se presentan como consecuencia de los aspectos que se descuidaron en el momento de la planeación.

### El Origen

El principio de toda esta serie de anomalías que se han ido describiendo a lo largo del capítulo anterior, se suscita cuando el ingeniero de sistemas se enfrenta al problema, analizando la situación y se impone de las necesidades que tiene que satisfacer el área de sistemas.

Es común que en esta situación, el desarrollo de sistemas o estrategias para la programación se tienda a implantar modelos totalmente nuevos y distintos a los que en ese momento se encuentran en operación y sin atender a los pormenores característicos del funcionamiento del proceso en uso, en la inteligencia de que, de este modo, se invertirá menos tiempo que si por el contrario, se elaboran minuciosos estudios que arrojarán algunos datos sobre los puntos en que el sistema activo presenta deficiencias.

Acto seguido, se procede a la planeación de los márgenes de trabajo con la consigna inicial de que estos márgenes se elaboren en la observancia de ofrecer un control más o menos claro, por lo menos en la fase inicial del nuevo sistema, de cada elemento participante en el proceso de la generación de sistemas.

Esto nos sugiere, en primera instancia, que se ha procedido a la elaboración de los estándares sin contar con las bases que nos permitan definir una filosofía que oriente su instrumentación de manera eficaz, clara y suficiente. Todas las medidas tomadas para la creación de estándares están fundamentadas en base circunstanciales *a priori*, es decir, independientes de la experiencia que implica satisfacer las necesidades encomendadas al departamento de sistemas y en cambio las instrumentaciones se realizan con una limitada gama de posibles situaciones que ha podido prever el diseñador en un momento dado. Cabe recordar que en este punto, como en muchos otros, cada problema es distinto y único en su naturaleza y, por ende, tan distinta y única debe ser su solución. Es entonces, como las soluciones empleadas con indiscutible efectividad en determinada situación, no suelen ser tan eficaces en otras muy similares. Sucede que temporalmente resuelven problemas de importancia en un sistema similar al ideal para aplicar dicha estrategia, pero posteriormente, se van presentando pequeños obstáculos como consecuencia de los detalles que se pasaron inadvertidos, debido a su poca monta en el sistema de estándares y, posteriormente, en un periodo no muy largo y dependiendo de la evolución de las necesidades que tenga que satisfacer el departamento, desde el programador hasta el área de sistemas misma en su conjunto, paulatinamente se gestan grandes deficiencias hasta convertir el conjunto de

normatividades en un obstáculo que merma y entorpece la misión que ha impulsando a la creación de "El departamento de sistemas".

### ***El criterio más usual***

El criterio seguido para la estandarización se integra con el prejuicio de que se sabe por experiencia que la aplicación de estándares es un medio eficaz para eliminar los problemas de desorganización que puedan aparecer durante la elaboración de los sistemas. En el supuesto de que inicialmente se resuelve este problema, simultáneamente se generan otras dificultades que reflejan sus efectos desfavorables en el tiempo de desarrollo de los productos. Aquí, ya nos encontramos con un ominoso bagaje de normas y recomendaciones que embrollan los procesos, erigiéndose en todo un insondable misterio administrativo el entender su funcionamiento. Funcionamiento de tal naturaleza que para ser entendido por el personal que se va integrando al equipo de trabajo, se hace necesario invertir tiempo e insumos económicos para instrumentar un curso de capacitación que habilite al personal a navegar a través de las innumerables denominaciones y nomenclaturas, con frecuencia indecifrables, que forman el ambiente de trabajo.

### ***Un caso frecuente***

En este caso que se expone a continuación, tenemos uno de los efectos perniciosos que produce el elaborar estandarizaciones sin previo estudio crítico de las condiciones que rodean a los problemas que deben resolver. Particularmente, se trata de un conjunto de funciones que se consideran "de uso frecuente" en la programación de sistemas para bases de datos relacionales en lenguaje Informix-4GL. Estas son alrededor de 3000 funciones que se encuentran almacenadas en un archivo de código fuente denominado *funcgloba.4gl* que, por la considerable extensión de su listado no presentaré en su totalidad, sino sólo un conjunto de tres funciones representativas del contenido en el citado archivo:

```

# -----
# Función que verifica la siguiente carga del bloque de temas.
# -----
FUNCTION sigu_tema (smov_p)
  DEFINE smov_p, sedo_l, SMALLINT
  FETCH RELATIVE smov_p uselte_m INTO ctmcve_g
  LET sedo_l = STATUS
  IF sedo_l = 0 THEN
    INITIALIZE sarete_m TO NULL
    LET srinte_m = srinte_m + 1
    IF c_temas(ctmcve_g) = 0 THEN
      LET sarete_m[1].tm_cve = ctmcve_g
      LET sarete_m[1].tm_desc = ctmdesc_g
    CALL antpos_tem(1,2)
  ELSE
    CALL aotpos_tem(1,1)
  END IF

```



```

ELSE RETURN 0
IF sedo_1 = NOUTFOUND THEN
CALL mensa)e(65)
ELSE CALL mensaje(93)
END IF
RETURN -1
END IF
END FUNCTION

```

```

-----
# Función principal de la ventana de ayuda de dependencias.
#-----
FUNCTION h_depen(sien_p,scol_p)
DEFINE sien_p,scol_p,scolumn_1 SMALLINT,
x char
LET sudep_n = 8
OPEN WINDOW wccatal_m AT sien_p,scol_p WITH 14 ROW,77 COLUMNS
ATTRIBUTE(BORDER)
OPEN FORM depen_h FROM "../frm/dependen_h"
DISPLAY FORM depen_h
LET scolumn_1 = centra(LENGHT("CATALOGO DE DEPENDENCIAS"),79)
CALL Cri(depen(scolumn_1))
CLOSE FORM depen_h
CLEAR WINDOW wccatal_m
CLOSE WINDOW wccatal_m
END FUNCTION

```

```

-----
# Función que permite al usuario hacer el criterio de búsqueda
# catalogo de dependencias.
#-----
FUNCTION cri_depen(scolumn_p)
DEFINE scolumn_p SMALLINT,
cwhede_1 CHAR (150)
LET sprime_1 = TRUE
WHILE TRUE
CLEAR FORM
DISPLAY "CATALOGO DE DEPENDENCIAS" AT 1, scolumn_p
LET INT_FLAG = FALSE
IF sprime_1 = FALSE THEN
DISPLAY "Esc -> Consultar, Pausa -> Cancelar" AT 12,03
CONSTRUCT cwhede_1 ON dp_cve, dp_nomb FROM arde_dep."
ELSE LET cwhede_1 = "1=1"
END IF
IF INT_FLAG THEN

```

```

CALL mensaje (69)
EXIT WHILE
ELSE
  IF curcdepen(cwhode_1) = 0 THEN
    LET sprime_n = FALSE
    IF dep_move() != 0 THEN
      EXIT WHILE
    END IF
  END IF
END IF
END WHILE
END FUNCTION

```

La creación del archivo `fungloba.4gl` ha sido exigida por la necesidad de reducir lo más posible el tiempo de desarrollo de los sistemas, se han creado funciones que en base a la experiencia de los mismos programadores se han encontrado como de uso frecuente y han sido agregadas al archivo `fungloba.4gl`. Con la finalidad de minimizar tiempo de desarrollo, estas funciones se integran a los códigos fuente durante el curso de codificación de los programas con miras de evitar que los programadores tengan necesidad de repetir varias veces un bloque de código que puede estar disponible en un sólo comando o función que a su vez será llamado con la función `CALL`. Además, los directivos del área han convenido en que todo aquel que agregue una función nueva al archivo deberá insertarle un encabezado con el formato sugerido, en donde se indique qué es lo que hace la función que encabeza. Además de que se han adoptado estándares para la nomenclatura de las variables creadas por el programador para ese programa en particular, según su función en el código fuente en las que se especifican algunas de ellas en el fragmento siguiente:

**ESTÁNDARES PARA LA NOMENCLATURA DE VARIABLES**

- ⇒ La longitud para los nombres de variables utilizadas en la codificación de los programas deberá invariablemente constar de ocho caracteres.
- ⇒ Los caracteres deberán ser siempre con minúsculas.
- ⇒ El tipo de variable que será indicado según el primer carácter que contenga

i - INTEGER, INT	e - SMALLINT
f - FLOAT, DOUBLE PRECISION	d - DATE
m - MONEY	c - CHARACTER, CHAR
b - BOOLEAN -- SMALLINT	n - DECIMAL, NUMERIC, DEC
s - SERIAL	r - SMALLFLOAT, REAL
r - RECORD	t - DATETIME
v - INTERVAL	a - ARRAY

<sup>1</sup> Se asume que en el compilador de Informix-4GL es factible trabajar con varios archivos de código fuente para generar un único programa ejecutable que, para su compilación, haya tomado alguno de estos módulos o archivos con extensión `*.4gl` como archivos de librerías de funciones.

- ⇒ Los siguientes cinco caracteres (2 AL 6) serán: el nombre alusivo a la variable: ejemplo: cnomb\_g
- ⇒ En el caso del carácter 7, siempre deberá ser un guión bajo: "\_".
- ⇒ El carácter 8 representa el ámbito de la variable

g - GLOBAL  
 p - PARÁMETRO  
 r - REPORTE  
 l - LOCAL  
 m - MODULAR

⇒ Si la descripción de la variable se compone de dos palabras, para formar el nombre alusivo. Se hará de la siguiente manera:

1. Las primeras tres letras corresponderán a las primeras tres letras de la primera palabra.
2. Las dos últimas letras serán las dos primeras de la segunda palabra.
3. No se consideran conjunciones ni preposiciones.

Ejemplo: Descripción de la variable: Nombre del Cliente

Tipo: CHARACTER  
 Alcance: GLOBAL  
 Sintaxis: cnomcl\_g

Alcance  
 Primeras dos letras de la segunda palabra  
 de la descripción de la variable.  
 Primeras tres letras de la primera palabra  
 de la descripción de la variable.  
 Tipo de la variable.

Según las reglas seguidas para la denominación de variables, notamos que cumplen con el objetivo de describir la naturaleza, función y ubicación de la variable en tan sólo ocho caracteres, si pensamos que el objetivo fuera ese precisamente. No obstante, suponer que una buena estructuración de las variables es garantía de solución, es un error común, pues se ha omitido considerar para qué tipo de programas se han de implementar estos lineamientos, falta saber si la cantidad de variables que se manejaran, acorde con la filosofía de diseño son de magnitud ideal para usar este tipo de nomenclatura o si será necesario incluir datos numéricos en las denominaciones de las variables en razón de la tipicidad de las mismas que exijan los diseños de software que se estén realizando. En suma, son una serie de consideraciones que se omite analizar en la inteligencia de un ahorro importante de tiempo

que otorgue celeridad a la implantación de los nuevos procedimientos para la creación de sistemas.

Siguiendo la evolución de este sistema de estándares, notamos que durante el tiempo que se ha trabajado en el desarrollo de los sistemas dentro de estas normas, se han ido acumulando las funciones que se almacenan en este archivo hasta superar la cantidad de 500 funciones, implicando una considerable extensión de código fuente muy superior a las 10000 líneas, lo cual causa que el tiempo que tarda en cargarse este archivo en memoria por cada ocasión que un operario necesite consultarlo o modificarlo, sea muy prolongado dada su gran extensión, retardando al mismo tiempo los demás procesos que se estén ejecutando en la red en ese momento. Los recursos de memoria del sistema son invertidos en tareas que muy posiblemente no sean del todo indispensables y que serían más productivas u óptimas en otros trabajos de más prioridad e importancia.

Observamos que, como es ya práctica generalizada en la mayoría de los sitios en donde se utiliza el lenguaje de programación Informix-4GL, las palabras que constituyen comandos de Informix-4GL están escritas con mayúsculas, práctica que es recomendada por los propios fabricantes y que hasta hoy día no ha tenido ningún inconveniente. El detalle por demás interesante, radica en que el número de funciones que se almacena en este archivo hace muy laborioso y tardado que un programador encuentre la función que sea la adecuada para resolver determinado problema, además de que conocer el funcionamiento de dichas funciones, tan sólo en cuanto al tipo de parámetros que pueden recibir y el cómo hacerlo exige que el programador se tome un tiempo considerable en el análisis para conocer la función y el cómo operarla, en el caso de que sea efectivamente la que necesita, sino, tendrá que continuar buscando entre los cientos de funciones restantes, presentándose como la más viable solución prescindir de las funciones ya establecidas y generar las propias al vapor. Esta decisión provocará cierto descontrol por parte de aquellos que en un momento determinado analicen el código, pues se encontrarán con que en la codificación existe código fuente superfluo que agranda el listado "innecesariamente" pues dará lugar a reflexionar: "debe haber alguna función en el módulo `fungloba.4GL` que satisfaga la necesidad"; y entonces, este listado será turnado nuevamente al programador para que realice las mejoras pertinentes.

Al contemplar la situación encontramos algunas interrogantes que arrojan una visión objetiva de los problemas que hasta ahora se han presentado:

- ¿Cuánto tiempo extra serán utilizados en las revisiones que se requieran por cada programa que presente una fracción de código redundante?
- Desde el punto de vista del programador: ¿Cabe esperar que éste llegue a experimentar un sentimiento de frustración ante un esfuerzo realizado que, difícilmente será reconocido gracias a esta incierta situación causada por los estándares aplicados en las funciones que se han implementado en el archivo `fungloba.4GL`?
- Las probabilidades de rotación de los recursos humanos se incrementa y con ello la inversión que implica capacitar al nuevo personal.

- A largo plazo, la situación suele ser más grave merced a que los mínimos detalles cobran su verdadera magnitud en las consecuencias que acurrean: ¿Cuánto tiempo será invertido en revisar y detallar aspectos de esta índole en la codificación?

### ***La participación del diseñador***

Desde luego, no debemos pensar que todas estas consecuencias son producidas en razón del mal gusto del diseñador de sistemas en la planeación. Es necesario considerar ciertas circunstancias que dificultan enormemente la labor de un diseñador. Tenemos que dentro del sistema de libre mercado, la competencia siempre se está adelantando y por tanto, siempre necesitamos una solución rápida para superarlos. Se presentan innumerables riesgos, desde las políticas de administración hasta los métodos de programación para cada caso en particular, los beneficios de la ventaja competitiva que ofrece la inversión en tecnología y los alcances de los logros que los operadores pueden hacer con este equipo, las implicantes de la capacitación del personal, la situación en constante transformación de las empresas, etc..., todas estas condiciones se manifiestan al diseñador en presiones y exigencias que deben ser satisfechas inmediatamente. Es un pesado fardo que el diseñador lleva a cuestas y que le impide muchas veces aquilatar con precisión y frialdad los factores para la toma de decisiones, a reserva de los métodos que se instrumenten para tal efecto.

Cuando la labor a acometer consiste en brindar un servicio informático de este tipo, el ingeniero a cargo no siempre está en condiciones de tomarse todo el tiempo que requiera para poder realizar un minucioso estudio del problema a resolver y del sistema ya implantado y que se considera obsoleto o insuficiente. Sucede que mientras el experto en sistemas decide qué se hará, los requerimientos de la empresa no admiten dilación alguna por ningún motivo y, por el contrario, exigen con imperiosa urgencia solución en ese mismo momento, la empresa no detendrá su actividad en tanto se diseña y libera el nuevo sistema; es por esta situación que se presenta el problema doble que por un lado, existen las necesidades que exigen respuesta en ese momento y, que el sistema en uso no satisface adecuadamente, por otra parte también, el obligado, detenido e ineludible estudio requerido para hacer las modificaciones inmediatas al sistema para disponer la implementación de un nuevo diseño con nueva filosofía de trabajo y adecuada administración. El diseñador, entonces se encuentra entre la espada y la pared, con dos necesidades a cubrir simultáneamente y las dos igualmente imperiosas y apremiantes.

Esta difícil situación, es la que conduce a tomar medidas que bien podrían calificarse de "desesperadas", a considerar sin más meditación y evaluación la decisión de implementar estándares para todos los elementos y con ello paliar el terror que produce la posibilidad de perder el control que conduciría a generar un caos de consecuencias tan impredecibles como inminentemente devastadoras de todo orden necesario para el desarrollo de sistemas.

Todas estas situaciones indeseables, son sólo anomalías que a la vez son síntomas iniciales de una sucesión de problemas que deterioran y lesionan todo el proceso de la generación de los productos. Son, por decirlo así, los detonadores de un progresivo desastre que difícilmente puede ser combatido y eliminado definitivamente y, sólo podrá, en el mejor de los casos, ser sobrelevado más o menos regularmente sin por ello asumir que se posee un

### III DETONADORES

---

proceso de trabajo del todo óptimo, es decir, siempre se registrarán, aunque muchas veces no tan visibles, pérdidas considerables en diversos puntos del sistema de trabajo.

## **IV PROPULSORES**

## Propulsores

Técnicamente, un propulsor es un mecanismo que impele hacia adelante un avión, un barco o simplemente un cuerpo. La propulsión es un impulso hacia adelante. De lo que se trata en este capítulo es de establecer los lineamientos esenciales, adaptables a la mayoría de las circunstancias que pudieran presentarse, para establecer estándares en la programación adecuadamente y, que estos tengan posibilidades de cambiar sin grandes complicaciones para actualizarse o adecuarse a las vicisitudes más probables que sufran los procesos a mediano y corto plazo. Los propulsores son todos los dispositivos que de una forma o de otra coadyuvan a maximizar y optimizar el rendimiento de las actividades que se originen a raíz de la creación de los sistemas.

La consigna fundamental de la implementación de estándares en la programación es ahorrar tiempo, impeler hacia adelante la creación de un sistema, **propulsar** su proceso de desarrollo en el sentido práctico de rendimiento máximo del producto a menor esfuerzo, resolver numerosos problemas que exigen la necesidad de un orden que involucre a todos los elementos que intervienen en el desarrollo de un sistema, desde la concepción fugaz en la mente del que ha ideado la solución a un problema, hasta la cristalización de un proyecto en pleno funcionamiento que cumple sobradamente con las metas previstas. Los estándares en la programación deben suprimir inversiones de tiempo y esfuerzo humano en actividades triviales pero absorbentes que retrasan la evolución favorable de un sistema en desarrollo.

### ***Prolegómenos fundamentales***

Existen guías que definen esencialmente la dirección y la tendencia que deberán regir todas las consideraciones y observaciones que se efectúen, además de que participan de forma indispensable en los criterios que se adopten para la solución de problemas en la mayoría de los casos.

Antes de establecer cualquier normativa para el diseño de estándares se debe observar el ambiente administrativo de la empresa en que van a integrarse los mismos. Este es un aspecto fundamental, pues es muy frecuente que los problemas que se deseen resolver con la implantación de estándares, sean consecuencias de una situación que tienda a pertenecer más al dominio de lo administrativo que del nivel operativo, y entonces, antes es necesario otro tipo de análisis más enfocado a esta naturaleza de problemas.

En la mayoría de los casos, establecer estándares en la programación ubica al diseñador en posición de elaborar estructuras administrativas - jerarquías, asignación de funciones, etc. - que complementen los objetivos de los estándares. Esto otorga al diseñador la libertad en la toma de decisiones, concernientes a la distribución del personal, trabajo y equipo; en suma, es una poderosa herramienta que le permite explotar al máximo las posibilidades de rendimiento de los recursos humanos y materiales; en contraparte, la inadecuada utilización de la misma puede ser el principio del un desastre posterior, de consecuencias del todo indeseables. Puede ser la diferencia entre un proceso óptimo y el origen de todos los problemas y efectos negativos que se han descrito en los capítulos precedentes.



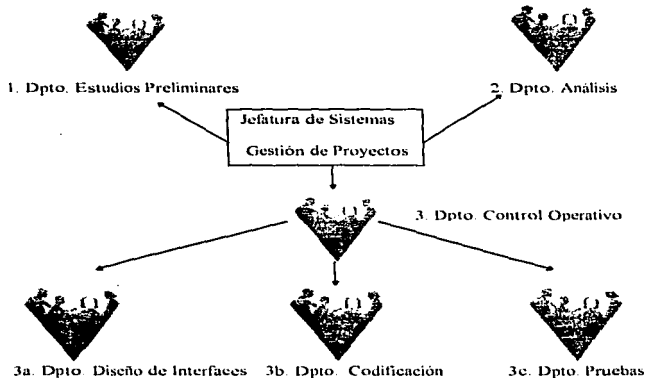
Siempre deberán observarse los aspectos puntualizados más abajo, en el inicio de los trabajos de diseño de los estándares, para ubicar la dirección y la filosofía adecuada a que deberá dirigirse el diseño de los procesos.

Desde luego, no se asume que todo aquel que tenga que diseñar estándares para la programación, necesariamente se encuentre en posibilidad de modificar la estructura administrativa del área de sistemas. Pero los puntos que se exponen a continuación, son lineamientos que en condiciones ideales deben cumplirse y, en condiciones diferentes, a reserva de la magnitud y trascendencia de las diferencias, deberán tender en lo posible hacia ellos:

**Evitar los feudos dentro del área de sistemas**

Con el término feudos, se pretende designar metafóricamente, a la situación que se genera al establecer una división del trabajo en el área, creando cajas negras que tienen asignadas funciones específicas y únicas en el proceso de desarrollo de un sistema. Existe, entonces, la tendencia a que los grupos de trabajo se consideren independientes y traten de especializarse en las actividades correspondientes a su función.

Para ilustrar esto, se considera la siguiente descripción de un área de sistemas hipotética, encargada de generar productos de software en Infornix-4GL:



La creación de un sistema de software implica el paso del proyecto por cada uno de los departamentos, según el avance que se tenga de su desarrollo. Cada departamento deberá entregar al siguiente módulo, según su numeración, el trabajo que se le ha asignado.

Lo que aquí se observa, es la división del trabajo existente entre los diferentes departamentos, esto implica la elaboración continua de informes detallados dirigidos a la Jefatura de Sistemas, que documenten y mantengan al corriente de las actividades que realiza cada uno de los departamentos. Implica también, que los operadores de un departamento sólo son partícipes del proyecto en lo que corresponde a la función que se les ha asignado. Destaca también, el tercer departamento de Control Operativo, el cual subordina a tres departamentos: Diseño de Interfases, Codificación y el departamento de Pruebas

Estas características de organización y distribución del trabajo, son muy frecuentes en razón de la supuesta claridad que ofrecen en cuanto a que presentan un esquema que facilita la detección de anomalías, al posible aprovechamiento de la iniciativa personal de la gente que labora en cada uno de los departamentos, merced a la especialización que trae como consecuencia mayor productividad y que con la jerarquización se amplía el control. Estos son los principales argumentos que "justifican" la implementación de este tipo de distribución del trabajo. Esta práctica, ciertamente, resuelve algunos problemas en los ámbitos típicamente administrativos, pero no resulta muy claro en lo referente a su aplicación y efectividad cuando se maneja sobre actividades relacionadas al desarrollo de sistemas, es decir, en ambientes netamente operativos.

Por su distinta naturaleza de trabajo, las condiciones suelen ser igualmente diferentes y a generar problemas muy diversos con consecuencias e implicaciones no previstas en las estructuras convencionales. Estos reveses se presentan como sigue.

*Se pone en riesgo la unidad del mando* - En el caso particular del área de Sistemas hipotética mostrada mas arriba, encontramos que la Jefatura de Sistemas tiene la mayor jerarquía dentro del área, esto implica que el mando y la dirección de toda la Jefatura de Sistemas se concentra en ese nivel.

Ahora bien, con este diseño de "feudos" en el área de sistemas, es habitual que el mando de la Jefatura de Sistemas se encuentre fraccionado entre los diversos departamentos que participan en el desarrollo de un sistema puesto que al existir departamentos especializados en cada uno de los diferentes procesos que atraviesa un sistema, estos departamentos tienden a definir férreamente los límites de su injerencia en el trabajo y se desentienden absolutamente de toda otra actividad que no se suscriba a esos límites. Tenemos que la fracción de mando que poseen los departamentos del área de sistemas procede de una visión limitada y constreñida a la consideración del proceso en que participan exclusivamente, en oposición, se halla que el mando en la Jefatura de Sistemas procede de una visión de conjunto muy generalizada a causa del desconocimiento de las condiciones peculiares de trabajo en los departamentos como consecuencia del encapsulamiento de los mismos, esto genera, entonces, un conflicto entre la autoridad especializada de los responsables de cada uno de los departamentos y la visión en conjunto del mando procedente de la Jefatura de Sistemas. Todo esto, naturalmente, en menoscabo de la optimización del trabajo.

*Se pone en riesgo la consistencia de las ideas.* - Si la Jefatura de Sistemas emite una idea que es pie de un nuevo proyecto de desarrollo; entonces, el departamento de Estudios Preliminares y Análisis se encargan de evaluar la viabilidad del proyecto. Posteriormente, será el turno del departamento de Control Operativo quien trabaje la idea y en los departamentos que están bajo su rango se efectuarán modificaciones sin considerar la concepción original de la Jefatura de Sistemas, la cual no tiene injerencia en estos puntos del desarrollo dado que sólo los departamentos especializados en sus funciones pueden aplicar su criterio altamente especializado. Es en este tramo del camino de desarrollo de sistemas, cuando la idea original emanada de la Jefatura de Sistemas se deforma y tiende a distorsionarse paulatinamente en razón de las aportaciones creativas de cada uno de los departamentos que participan en el desarrollo.

*Riesgos en la unificación de esfuerzos en los proyectos.* - En la modalidad de estructura de trabajo que analizamos, es evidente que cada departamento trabaja con una filosofía y visión diferente e independiente de las demás. Es decir, que por ejemplo, el departamento de Análisis tiene una visión del desarrollo de un sistema solo en cuanto a las condiciones de utilidad que reportaría dicho sistema, sin reparar en lo más mínimo, en lo que podría significar para el departamento de Control Operativo. O sea, que no podemos contar con que exista una labor de equipo sino un conjunto de equipos con un conjunto de esfuerzos independientes contribuyendo en forma fragmentaria al desarrollo de un sistema. En suma, se tiene un conjunto de procesos en donde no hay interacción entre ellos y por tanto no existe coordinación en la unidad de un objetivo común y con ello, la inexistencia de la concepción de un sistema unitario sino como la unión de partes diferentes, es decir se pierde la integridad final de los sistemas que se desarrollan.

*Varios bancos de datos (Dispersión de información).* - Cada departamento para desarrollar su trabajo requiere de cierta información según sea la función que efectúe. Si hablamos del departamento de Estudios Preliminares por ejemplo, se requerirá información acerca de las capacidades del equipo de que se dispone para correr las aplicaciones que se generen y, especificaciones sobre la naturaleza de la información, además de datos sobre los futuros operarios del sistema. O en el caso del departamento de Codificación, entonces éste necesitará de los códigos de otros programas e información sobre los lineamientos y estructuras de construcción de los sistemas que se encuentren ya en operación. En todo caso cada uno de los departamentos deberán tener acceso en forma independiente a la información que necesiten, lo que da pie a que se generen, en primer término, dificultades para su administración y, con ello aumentando los riesgos que implica la dispersión de la información, menguando las condiciones de seguridad de la misma y facilitando el que mayor número de usuarios que participan en el desarrollo de los sistemas tenga acceso a la información vital, concluyendo todo esto en un sistema de desarrollo con seguridad general muy discutible.

*Comunicación limitada.* - En las circunstancias que rodean a esta estructura es común que los integrantes de cada uno de los departamentos es consciente de la función específica que le es asignada. Esto implica que en cada departamento se genera una situación similar a una

caja negra, en donde el departamento se torna hermético, en otras palabras, el departamento como módulo de un proceso, únicamente tiene considerado el material o elemento de entrada que será procesado en la parte que corresponde a su función, después se le dará salida de ese departamento. Es en este momento donde termina todo vínculo participativo del departamento con los demás procesos. Tácitamente se crea un consenso en todos los departamentos que intervienen en el proceso, donde ningún otro departamento puede tener injerencia alguna en ninguna etapa del desarrollo del sistema que no le corresponda, cada uno deberá entregar el trabajo asignado y no tener opinión sobre lo que le sea "ajeno". Esto es, que la comunicación entre los departamentos solo se realiza en torno a lo que a cada uno de estos le afecta directamente a su función. He aquí la comunicación limitada.

*Altos costos operativos* - Esta situación de pequeños feudos genera dispersión de recursos (material y equipo) como consecuencia de que la comunicación se reduce a un muy estrecho ámbito que tiende a reducirse en cuanto que cada departamento, según lo indiquen la naturaleza de la función que tengan asignada, requiere de equipo específico adecuado a su trabajo. Nos referimos a que cada departamento debe elaborar, en forma independiente, informes o reportes que presenten a la Jefatura de Sistemas el estado de cosas en la evolución de los trabajos lo cual implica un gasto en papelería, tiempo, trabajo del personal que integra los departamentos y energía eléctrica empleada en las terminales e impresoras comprometidas en la elaboración de estos documentos. Otro de los peligros es que se propician mayores posibilidades de que se halle equipo encendido sin utilizar contribuyendo al consumo eléctrico innecesario.

#### **Entonces: ¿qué hacer?**

La planeación de una estructura administrativa destinada al desarrollo de sistemas debe comenzar desde una hoja en blanco, desde lo más simple, sin jerarquías ni departamentos. Debe comenzar con una idea simple imaginando que todo el sistema lo realiza una sola persona que dispone de una PC, impresora, módem y como único usuario de una terminal de red que trabaja solo con una filosofía que es la de desarrollar los sistemas en las mejores condiciones posibles sin desatender los parámetros iniciales que especifican las necesidades que se pretenden cubrir con la implementación del sistema. Esto presupone que el control sobre el trabajo de esta persona imaginaria es relativamente sencillo y factible, es un proceso único, sólo un banco de datos, no hay información ni recursos dispersos, se tiene un estado de cosas en donde el equipo está optimizado y existe una unificación del esfuerzo, la aparición de una idea por parte de este único diseñador podrá ser evaluado con la circunspección y se desarrollará en caso de ser viable.

Evidentemente el trabajo no puede ser hecho por una persona únicamente, sería una labor enorme y tomaría mucho tiempo, se requeriría de más gente que apoyara a este desarrollador, esta gente también usaría el equipo disponible, más espacio, etc... Con los apoyos que se integren, tanto de equipo como de personal y según la magnitud que el trabajo requiera, el diseñador de la estructura de trabajo deberá conservar la integridad inicial del desarrollador que ahora contará con mayor capacidad de trabajo, es decir, con más pares de manos que operarán una terminal por cada par de manos adicional que posea por ejemplo. De aquí en adelante se organizara el trabajo rompiéndolo en partes, o sea,

determinando los componentes que son significativos a cada etapa del proceso de desarrollo del sistema.

Cuando la estructura ya ha sido creada y se encuentra en posibilidades de ser cambiado para optimizar el proceso se debe tender hacia la simplificación y la limpieza en el trabajo y tratando de evitar los conflictos que se han descrito en el apartado anterior.

#### **Desafíos**

Las situaciones anteriormente expuestas nos colocan en una encrucijada en la que se hace patente la necesidad de dos condiciones básicas e indispensables sin las que el trabajo de desarrollo de sistemas se dificulta enormemente:

- **Crear un proceso no una división del trabajo.** Este es el primer punto a considerar puesto que observar esta condición crea una relación de interdependencia en el equipo de trabajo. Todos están compenetrados de la actividad que realiza el "otro" en la medida en que se trata de un sólo trabajo. Las ideas que surjan durante el trabajo tienden a ser más frecuentes por causa de la continua retroalimentación de los integrantes del equipo. De igual modo la responsabilidad, los estímulos y las satisfacciones recibidas por el trabajo realizado repercute por igual sobre todos y cada uno de los integrantes del grupo.
- **Rápida y efectiva respuesta a los cambios de tendencias en los objetivos del desarrollo de sistemas.** En la industria es inherente lo inestable de las circunstancias en que se generan las necesidades a satisfacer, cambian constantemente. Si la estructura de trabajo conserva su unidad fundamental, entonces es factible que al ser un sólo esfuerzo y una dirección, pueda ser adaptable a los cambios, maleable a los cambios de dirección o filosofía para el desarrollo de los sistemas.

#### ***El Propulsor***

Ahora que han sido expuestas las tendencias más generales como antecedentes del trabajo del diseño de estándares para la programación y, habiéndose mostrado las disyuntivas que ofrece, es tiempo ya de abordar el estudio del propulsor en si, es decir, aquello que disparará la evolución del desarrollo de los sistemas, en procesos limpios y optimizando cada actividad que se haga necesaria. El Propulsor es el método, el modo razonado de proceder en forma de una marcha racional del espíritu para conquistar su preeminencia sobre la materia. Es la metodología para el desarrollo de estándares.

La metodología que ahora nos ocupa, cubre los puntos a considerar para la creación de estándares, desde la planeación administrativa de los procesos de desarrollo, hasta las minucias que requiere establecer el orden adecuado en los códigos de los programas. Esto no es obstáculo, por supuesto, para que el método pueda aplicarse en procesos de desarrollo de aplicaciones a los cuales haya necesidad únicamente de crear estándares para la codificación de los programas. En este caso lo que se impone es considerar todas las actividades de desarrollo en función de los códigos de programación, del mismo modo, se

procede en el caso de que lo que haya que modificarse sea el control administrativo, aunque en la mayoría de las veces esto implica el cambio de todos los estándares de desarrollo de sistemas. Desde luego, esta decisión se deberá tomar con estudio a las condiciones específicas del sistema en cuestión.

En ocasión de la estandarización de procesos de desarrollo de sistemas, de los cuales se desconoce todo antecedente de funcionamiento, se debe de tener especial cuidado de obtener datos de la operación regular del Área de Sistemas: entradas de datos, salidas de información, etc. y así contar con elementos pertinentes para la generación de los estándares adecuados.

#### ¿Y en los sistemas de programación relativamente pequeños?

En lo tocante a la magnitud de los procesos de desarrollo de sistemas que se desee estandarizar, es menester puntualizar que el método ha sido diseñado y expuesto con la finalidad de poder ser utilizado en la estandarización de Áreas de Sistemas de volúmenes de trabajo más o menos considerable.

Para la estandarización de procesos que no sean muy complejos o grandes, se deben tomar sólo los puntos de las actividades que se consideren indispensables. Por ejemplo, para proceder a la estandarización de un núcleo de trabajo reducido, muy probablemente será conveniente que en la asignación de responsables participantes en la estandarización, sea una sola persona la que se le asigne la función de Estudio Preliminar, Análisis y Diseño y, a su vez será un sólo elemento participante el que trabaje con los cometidos de Pruebas, Control de Calidad y Descripción de Procedimientos. Estas peculiaridades serán ventiladas en su oportunidad por el jefe de proyecto o el responsable del Área de Sistemas, fundamentándose en las evaluaciones de los objetivos. Según sea la magnitud de los procesos a estandarizar, será el número de personas implicadas en el proyecto de estandarización e igualmente proporcional la minuciosidad en el cumplimiento de los puntos específicos de las actividades que propone el método.

De igual modo la elaboración de la documentación deberá seguir este mismo canon, no así en los puntos correspondientes a los objetivos, que es lo esencial en todo proyecto, en donde se deberán considerar a juicio todos los puntos indicados.

#### Metodología para la creación de estándares

El objetivo de la metodología para la creación de estándares, es el establecer un orden lógico de las actividades a realizar para el desarrollo de sistemas, mediante la utilización de las herramientas adecuadas, la generación de estándares para el sistema y para el control de proyectos y, la identificación de los participantes para cada una de las etapas que cubren el desarrollo de los sistemas, con las miras a lograr:

- Asegurar la satisfacción de los requerimientos del Área de Sistemas, de la empresa y del usuario.
- Incrementar el control de calidad.
- Evitar desviaciones en costo y tiempo.
- Facilitar el mantenimiento.
- Generar la documentación indispensable de los estándares que requiera el proyecto.

- Asegurar la productividad de los recursos humanos.

#### **Atributos y características de la metodología:**

- Flexible.- La metodología es aplicable a las diferentes características de los sistemas en cuanto a las dimensiones de su volumen de trabajo o grado de complejidad de los procesos que implican y de la herramienta de programación que utilicen.
- Funcional.- No necesariamente se tienen que ejecutar todas las actividades que se indican sino que pueden ser eliminadas sin que por ello se corra el riesgo de efectuar una planeación deficiente. Desde luego, la decisión de los pasos que se eliminarán de la metodología debe ser evaluada considerando el tamaño del sistema que se pretende crear y su complejidad de procesos, cuanto más compleja sea la estructura de estándares que se requiere, más meticulosamente y con todo apego, deberán seguirse todos los lineamientos y sugerencias marcados en las actividades.
- Normativa.- Es una guía que norma el desarrollo de estándares puesto que se producen en base a una secuencia lógica que amplía las posibilidades eficiencia y precisión en la generación de los estándares que son necesarios.
- Evaluable.- Sus productos - reportes, documentos de prueba, planes de trabajo, etc.- permiten llevar mejor control del avance de proyecto.
- Participativa.- Involucra a los usuarios del sistema de estándares para el estudio y seguimiento de las evaluaciones y resultados de los diseños y pruebas de los estándares generados.
- Integral.- Está orientada al desarrollo de recursos de información.

#### **Alcance:**

Relativo a la mejor solidez, estabilidad y precisión que brinda la metodología en los siguientes aspectos:

##### **Nivel estratégico:**

- ⇒ Planeación y control estratégico. Porque dirige los esfuerzos hacia el lugar preciso y otorga una visión clara de las necesidades que deben ser cubiertas por el sistema de estándares que se creará.

##### **Nivel táctico:**

Después que se ha reunido la información y esta ha sido evaluada se procede a la generación de los estándares con actividades en conjunto con los usuarios. Estas actividades de creación y desarrollo siguen un patrón claramente marcado y delimitado que abarca los siguientes aspectos:

- ⇒ Planeación de desarrollo o el cómo se elaborarán los estándares.
- ⇒ Planeación de los servicios y necesidades que se pretenden cubrir.
- ⇒ Planeación administrativa. Se consideran las jerarquías de funciones y funcionarios que participan en la creación de estándares.

- ⇒ Planeación de recursos humanos, de tiempo y costo disponibles para el desarrollo de los trabajos.

#### Nivel Operativo:

La metodología dispone de medios eficientes para el monitoreo de las actividades y la evolución de los trabajos en la creación de estándares así como de preceptos que son oportunamente marcados para su ejecución que anticipan las condiciones en que se trabajará con la siguiente etapa del proceso de creación de estándares. En conjunto cubren los puntos referentes a:

- ⇒ Control de desarrollo. Visualiza la evolución de la creación de estándares.
- ⇒ Control de mantenimiento. Es el seguimiento de los procesos en el campo de operación.
- ⇒ Control de recursos. Permite administrar óptimamente los recursos de costo y tiempo disponibles para cada una de las etapas de la metodología.

#### Actividades:

Listado de actividades que establece la metodología para el desarrollo de estándares:

1. ESTUDIO PRELIMINAR. Es la primer actividad que se realiza, consiste principalmente en una labor de observación objetiva del estado de cosas existente en el momento de la observación identificación de los problemas y de las deficiencias/opportunidades que deben entenderse como la identificación de sucesos que pueden mejorarse significativamente con la implantación de los estándares.
2. ANÁLISIS. Esta actividad es fundamentalmente, trabajar con los datos obtenidos en el punto anterior, se resume todo lo que se ha encontrado y se usa para identificar los elementos importantes para la toma de decisiones.
3. DISEÑO. Con ayuda de los elementos obtenidos, el personal designado podrá elaborar las especificaciones pertinentes a cada uno de los estándares de trabajo, acceso de información y de administración que se requieren.
4. CONSTRUCCION DE ESTÁNDARES DE PROGRAMACIÓN. Aquí se crean los estándares relativos a la interacción de los programadores y usuarios en general, con el software en el desarrollo y programación de los sistemas.
5. PRUEBAS. Imprescindible en todo proceso de diseño, es la fase de pruebas que permite identificar fallas en los estándares antes de su funcionamiento en el campo real de trabajo. Para las pruebas se utilizan situaciones recreadas que posibiliten la provisión de anomalías en el cumplimiento del cometido de los estándares.
6. CONTROL DE CALIDAD. Estrechamente ligado con la actividad de pruebas, el control de calidad es la etapa en que se afinan los detalles que han quedado aún imperfectos. Desde una perspectiva global de todo el sistema de estándares, el control de calidad afianza la seguridad de la eficiencia de los estándares cuando estén funcionando.
7. DESCRIPCIÓN DE PROCEDIMIENTOS. Con la colaboración de los usuarios, se labora para la elaboración amena y de fácil comprensión de los manuales de uso y documentación indispensable de los estándares.



8. **CONVERSIÓN DE DATOS** Es la adaptabilidad de los datos e información con que se trabaja a los nuevos lineamientos que se implantarán en el Área de Sistemas en general. Bases de datos, contenido de los archivos, directorios, etc.
9. **IMPLANTACIÓN.** Se realiza la implantación del sistema de estándares y se hacen las evaluaciones finales y pequeños ajustes. En realidad, el proceso de evaluación es un proceso cíclico que se realiza durante toda la vida del sistema conjuntamente con los procesos de mantenimiento.

Al inicio de cada una de las actividades del desarrollo del sistema de estándares, se marcan las áreas que participan en la actividad según su especialidad. Estas áreas, en el entorno del método, son grupos de personas que tendrán asignada su función en la integración del sistema, no deben ser interpretadas como direcciones o departamentos dentro del Área de Sistemas, sino como equipos de trabajo. Los equipos de trabajo o las áreas participantes en el desarrollo y sus funciones particulares, se enumeran a continuación:

1. **Área de Sistemas.** - También nombrada Centro de Cómputo, Departamento de Sistemas o técnicamente Unidad de Sistematización de Datos (U.S.D.), físicamente, es el espacio donde residen las computadoras que integran el núcleo del manejo de la información de una empresa, dependencia oficial o privada e incluso, puede ser un centro totalmente independiente cuyo cometido es la producción de información. En nuestro caso de estudio, se trata del departamento de la corporación que produce las aplicaciones de software para el manejo de la información. Físicamente está constituido por:
  - Área Física.
  - Equipos.
  - Recursos Humanos
  - Mobiliario y Materiales.
2. **Área Usuaria.** - Es el conjunto de diseñadores de sistemas, programadores, y todos aquellos que se ven involucrados en el proceso de desarrollo de los sistemas para la empresa o corporación. Se les define usuarios en razón de que son los usuarios potenciales de los estándares que se implementan para el desarrollo de sistemas, en otros términos, el Área Usuaria no se integra por los usuarios de las aplicaciones que se desarrollan en el Área de Sistemas, sino por el personal que desarrolla los programas de aplicación.
3. **Desarrollo de Sistemas.** - Es el conjunto de personal especializado que tiene asignada la labor de establecer los planes y programas de desarrollo integral para el Área de Sistemas con base a objetivos y políticas de la corporación o empresa a la que pertenezca o brinde servicio el Área de Sistemas.
4. **Desarrollo de Nuevos Productos.** - Se encargan de diseñar y establecer sistemas de planeación y control de proyectos que permitan una mejor administración de los recursos en el desarrollo e implantación de los mismos, también se ocupan de documentar gran parte de los trabajos que se efectúan para el desarrollo del sistema de estándares.

5. Planeación Tecnológica - Recopila y analiza a detalle la infraestructura de los sistemas de información requerida para el diseño de las diferentes aplicaciones de acuerdo con las disposiciones y normas existentes
6. Implantación de Sistemas - Los designados con esta función deberán evaluar antes de su implantación y durante su ciclo de desarrollo los sistemas de estándares elaborados, además establecer en coordinación con los demás grupos de trabajo, los puntos de control y calendarios de producción.
7. Atención a Usuarios - Son los encargados de presentar y difundir las novedades que aporta el nuevo sistema de estándares al personal que integra el Área Usuaría y de dar oído a las inquietudes que esto pueda generar.
8. Producción.- Genera artificialmente en el sistema, las condiciones necesarias para el examen de rendimiento de los nuevos estándares.
9. Operación.- Es el personal responsable de la operación física de los equipos electrónicos y todos aquellos periféricos de la configuración principal con que cuente el Área de Sistemas con apego a los manuales e instructivos existentes.



#### ESTUDIO PRELIMINAR

##### Objetivo

Identificar las deficiencias/opportunidades actuales en el ambiente del usuario, establecer objetivos, determinar si es posible automatizar, de ser así, sugerir escenarios aceptables y preparar el plan de proyecto

Áreas involucradas:

Área usuaria  
Desarrollo de Sistemas

##### ◊ Identificar deficiencias/opportunidades actuales:

- Obtener y analizar información relativa al proyecto. Todos los detalles son de mucha importancia: métodos de trabajo, estabilidad en el volumen de trabajo, etc. Se debe poner especial atención a las posibilidades de cambios en los requerimientos del departamento de sistemas, con que frecuencia o periodicidad se podrían efectuar y la trascendencia de los mismos.
- Coordinar juntas con personal involucrado. Las juntas son con el objetivo de obtener los testimonios verbales de las situaciones que generan conflicto en el

trabajo al desarrollar los sistemas, identificar objetivos, roles y funciones del área usuaria y especificar requerimientos, políticas del usuario y la situación actual.

Este paso debe generar un Reporte de Características Deficientes/Oportunidades: Es un documento en forma narrativa, relativamente corto, que debe contener detalles de la información siguiente:

- Naturaleza de estándares requeridos. Se puntualiza en el tipo de estándares que se pretenden: Administrativos, jerárquicos, para programación de sistemas, etc.
- Relaciones de costos. Inversiones de tiempo y recursos físicos como PC's, papelería, disquetes, etc. necesarios durante la gestación del proyecto.
- Referencia a problemas de: eficiencia, seguridad y operación.

Deberá ser elaborado por los responsables del Desarrollo de Nuevos Productos.

○ Establecer objetivos del sistema de estándares:

- Determinar tipos de estándares requeridos.
- Determinar la magnitud del proyecto. Se marcan los alcances y la trascendencia de los estándares que se proyectan, definiendo a quienes y cuantos usuarios se verán afectados con la realización del proyecto.
- Determinar requerimientos de seguridad y exactitud. Indicar que tan minuciosos deben ser los estándares y prever que el modelo de los mismos no pueda dar pie a confusiones o a interpretaciones ambiguas o poco explícitas.
- Determinar fechas límite. De ser posible, se sugiere utilizar algún tipo de diagrama que represente la evolución del proyecto.
- Determinar limitantes (de recursos humanos, costos y tiempo).
- Determinar en el sistema de estándares actual:
  - Áreas departamentales o procesos en el desarrollo de sistemas que manifiestan problema.
  - Áreas departamentales o procesos en el desarrollo de sistemas que presentan oportunidad.
- Establecer objetivos y beneficios.

Aquí se impone la elaboración de lo que se denomina el Reporte de Objetivos Funcionales que es un documento en forma narrativa, relativamente corto, que contiene los objetivos funcionales y de "Performance" del nuevo sistema de estándares, conteniendo la información siguiente:

- Lista completa de los tipos de estándares requeridos.
- Limitaciones de costo.
- Requerimientos de seguridad y/o exactitud.
- Lista de áreas o departamentos afectados.
- Lista de personas a quien dirigirse dentro de cada uno de los departamentos.

Deberá ser elaborado por los responsables del Desarrollo de Nuevos Productos.

◊ Generar escenarios aceptables:

- Definir posibles soluciones.
- Estimar costo, beneficio, y tiempo para cada posible solución.
- Proponer el mejor escenario.

Corresponde en este caso a la generación del Reporte de Escenarios Aceptables que al igual que los anteriores, es un documento en forma narrativa y de diagramas, breve, pero completo que contiene escenarios que satisfagan los nuevos objetivos del usuario, así como restricciones impuestas. Este reporte consta de:

- Breve resumen del sistema propuesto.
- Resumen de costo/beneficio.
- Funciones de los estándares.
- Procedimientos.
- Beneficios.
- Recursos.
- Plan de desarrollo.

Deberá ser elaborado por los responsables del Desarrollo de Nuevos Productos.

◊ Preparar plan general del proyecto

- Identificar actividades de factor externo, en el desarrollo del proyecto. Es decir, aquellas actividades que si bien no ocurren dentro del Área de Sistemas, si influyen en el trabajo que se desarrolla dentro del área.
- Determinar disponibilidad de recursos requeridos para cada una de las actividades del proyecto.
- Estimar tiempo de desarrollo para cada una de las actividades del proyecto.
- Estimar fronteras y limitantes del proyecto.
- Crear esquema de lo que será el plan general del proyecto, tomando en cuenta las áreas involucradas.

Se elabora el Plan General del Proyecto: documento en forma narrativa y de diagramas que en forma completa presenta un plan general, con información costo/beneficio, objetivos, presupuesto y un calendario de actividades, conteniendo:

- Nombre del proyecto y su propósito.
- Responsable del proyecto o del Área de Sistemas.
- Área específica a mejorar.
- Fronteras y relación con otros sistemas.
- Personal requerido en el desarrollo.
- Fechas de inicio, revisiones y terminación.
- Presupuesto (recursos humanos, equipo y costo).
- Antecedentes y problemas surgidos anteriormente con los estándares.
- Requerimientos del usuario.

- Escenarios aceptables de solución.
- Calendario de actividades.
  - Gráficas.
  - Fechas clave.

Los responsables de la elaboración de este documento serán: el Departamento de Desarrollo de Nuevos Productos, Desarrollo de Sistemas, Planeación Tecnológica e Implantación de Sistemas.



### ANÁLISIS

#### Objetivo:

Obtener en forma gráfica y narrativa las especificaciones que satisfagan el ambiente actual del usuario, los nuevos requerimientos y establezcan las funciones que se realizarán en forma computacional.

Cabe aclarar que para el cumplimiento cabal de los preceptos correspondientes a la actividad de análisis, se debe contar con el Diagrama de Flujo de Datos (D.F.D.) y Diagrama de Estructura de Datos (D.S.D.) del sistema en que se deseen implantar los estándares.

#### Áreas involucradas:

Área usuaria.  
Desarrollo de sistemas.

- ◊ Realizar planeación detallada de la actividad. La planeación debe ser detallada y meticulosa si se desea asegurar el buen éxito del plan que se sigue. Contempla además, actividades diversas en su naturaleza.
  - Conocer organización, misión, roles y objetivos del área o áreas por analizar.
  - Identificar posibles fronteras, políticas y organizacionales.
  - Definir entrevistas para precisar procedimientos y funciones de cada usuario del sistema de estándares.
- ◊ Documentar sistema de estándares actual. Se reúnen todos los elementos - manuales, esquemas, organigramas, etc. - que faciliten el conocimiento del sistema de estándares actual, que incluya la organización administrativa y jerárquica con nombres de personas y sus funciones.
  - Identificar catálogos y elementos distintivos de la información con la que se trabaja.
  - Identificar la información contenida en formatos y reportes de las actividades que realizan los que participan en el desarrollo de los sistemas.

- Conocer relación de entradas y salidas de datos en cada función o etapa del desarrollo de los sistemas.
  - Identificar volúmenes y frecuencias de trabajo en la información, es decir, que tan grande es la magnitud y que tan seguido se presenta la información con que se trabaja.
  - Identificar totalmente el ambiente actual en lo concerniente a los aspectos que pudieran ser accesorios y particulares según la naturaleza del sistema para el que se deseen desarrollar los estándares.
- ◊ Derivar equivalente lógico (opcional). Es opcional en cuanto sea el criterio que utilice el diseñador de estándares, no debe ser evaluado en cuanto al tamaño del sistema a estandarizar sino sólo en cuanto a su complejidad de los procesos. Se recomienda utilizar diagramas.
- Reemplazar nombres de personas específicas por funciones.
  - Eliminar referencias a información particular, como tiempo o secuencia.
  - Eliminar burbujas que indiquen transferencia física de información, si no son esenciales para el modelo lógico.
  - Reagrupar y reacomodar funciones tratando de reducir la conexión entre datos.
  - Eliminar funciones redundantes.
  - Eliminar archivos de paso, copias duplicadas de datos y operaciones físicas de entrada/salida que pueda contener el modelo físico.
- ◊ Definir nuevos estándares lógicos y físicos atendiendo a las deficiencias/oportunidades encontradas y a los objetivos que se persiguen con el trabajo de desarrollo de sistemas del Área de Sistemas.
- Derivar nuevo diagrama de flujo de datos lógico y físico.
  - Producir diagrama de estructura de datos con estándares de nomenclaturas y normalizado.
  - Asegurar integración al plan general de Base de Datos a usarse para el desarrollo de sistemas verificando las ventajas que se han previsto.
  - Elaborar miniespecificaciones que definan los estándares que serán usados para cada uno de los diferentes procesos del desarrollo de sistemas.
  - Complementar nuevo diccionario de datos con observaciones acerca de los estándares aplicables.
- ◊ Realizar análisis de factibilidad. En el presente trabajo, se debe entender por análisis de factibilidad al estudio que nos otorgará elementos para la toma de decisiones respecto de si es conveniente la implementación de nuevos estándares o no, en base a estudios de carácter técnico económico y operativo.

- Cuantificar el modelo físico en base a inversión, tomando en cuenta costos de desarrollo, implementación, operación y riesgos. Aquí se deben contemplar como costos también las inversiones de tiempo
  - Cuantificar el modelo físico en base a oportunidad y calidad de sus resultados. Fundamentalmente la optimización de los equipos disponibles y su eficacia relativa al trabajo específico al que son sometidos.
  - Definir el entorno tecnológico de operación: cómo se debería operar en condiciones ideales
  - Evaluar equipo/software/comunicaciones en cuanto a si es el adecuado a la función que se requiere de estas herramientas
  - Evaluar paquetes de proveedores en caso de que se requiera integrar nuevo software o hardware. Se deben considerar con las diferentes ofertas de equipo que deben ser probadas con las condiciones que generarán los estándares a desarrollar y sin olvidar lo importante que será la calidad en el servicio de soporte de las empresas proveedoras.
  - Elaborar reporte de costo beneficios. El reporte debe ser lo más claro, conciso y sencillo posible. No debe de invertirse muchos recursos en el estudio de factibilidad dado que hasta aquí aún no se ha determinado si es viable la implantación del presente proyecto de estándares, pues en caso de no ser conveniente su implantación, entonces se tendrá que evaluar con respecto a otra posible solución
- ◊ Preparar la información obtenida para la presentación del proyecto para la toma de decisión de su viabilidad.
- Recopilar información relativa a especificaciones detalladas del proyecto.
  - Evaluación del análisis, con respecto a los objetivos originales.
  - Obtener aprobación de terminación de la actividad del análisis.
- ◊ Delimitar el sistema de estándares.
- Elaborar resumen de restricciones reales y ficticias, o sea, las que son de carácter potencial en el momento de la operación del sistema de estándares. Las limitaciones que tiene el nuevo sistema de estándares con respecto a la situación actual.
  - Conducir junta con involucrados para presentación del proyecto. Los involucrados son las personas que aprobarán o desaprobarán la implantación de los nuevos estándares.
- ◊ Actualizar los planes del proyecto.
- Determinar la disponibilidad de los recursos requeridos en las actividades siguientes.

- Estimar costo y tiempo para el desarrollo de cada una de las actividades siguientes.
- Establecer puntos de control y fechas de revisión.
- Determinar las actividades siguientes.

Realizar el Reporte de Análisis de Factibilidad, dividido en tres puntos:

- Situación actual:
  - Antecedentes.
  - Diagrama conceptual del medio ambiente del problema y/o del área de oportunidad.
  - Diagrama de flujo de datos.
  - Descripción de procesos.
  - Problemas y/o áreas de oportunidad detectadas.
  - Requerimientos básicos a satisfacer.
- Opciones de solución:
  - Descripción y diagrama conceptual de los estándares proyectados.
  - Requisitos para el desarrollo y la operación de cada una de las opciones de solución encontradas.
  - Análisis de costo/beneficio de todas las opciones de solución encontradas.
- Recomendación:
  - Descripción de la opción seleccionada.
  - Razones principales de la selección
  - Factores críticos de éxito.
  - Beneficios esperados (cualitativos y cuantitativos).
  - Costos estimados en inversión inicial y de operación.
  - Plan maestro de desarrollo, es decir, cómo serán planeados y con qué perspectiva.

Deberá ser elaborado por los responsables del Desarrollo de Nuevos Productos.



## DISEÑO

Objetivo:

Identificar la jerarquía de estándares e interfaces entre los procesos, para implementar las especificaciones del análisis y los procedimientos computacionales.

Áreas involucradas:

Desarrollo de Sistemas



- Derivar estándares de interfaces entre los procesos de desarrollo de sistemas. Se debe de elaborar según lo indicado en el apartado "Entonces : ¿qué hacer?" al principio de este mismo capítulo.
  - Definir jerarquía de cada estándar de las actividades que se realicen en el proceso de desarrollo de sistemas.
  - Definir función de cada estándar.
  - Definir relación de los componentes de los estándares.
  - Crear diagrama de estándares en base a la información que presentan o trabajan.
- Evaluar diagrama de estándares.
  - Revisar cohesiones y conexiones.
  - En caso de existir, corregir problemas de diseño.
- Diseñar estándares.
  - Definir en forma general la función en cada uno de los estándares.
  - Especificar el objetivo de cada estándar.
  - Estimar periodicidad, calidad y volumen de la información que proporciona cada proceso estandarizado.
  - Definir la oportunidad de la información, en términos de tiempo y lugar.
  - Definir los elementos de control y seguridad de la información.
  - Evaluar el impacto en recursos de cómputo e involucrar a servicios de cómputo.
  - Diseñar procedimientos de seguridad estandarizados.
  - Evaluar responsabilidades de usuario y de sistemas
  - Adecuar y preparar infraestructura, en acuerdo con los estándares de la instalación.
- Diseñar Base de Datos de la información que se maneja para el diseño de los sistemas (DBA). Desde este paso, hasta el final de la presente actividad, se utilizarán los conceptos y pasos para el desarrollo de bases de datos del mismo modo que en un sistema de información común, para tal efecto existe la documentación a utilizar en este proceso que se puede encontrar en las obras de desarrollo de Bases de Datos existentes en el mercado. Ahora bien, las recomendaciones presentadas aquí, indetectiblemente deben ser complementadas con las observaciones particulares que a cada Base de Datos corresponda atendiendo a la literatura especializada en el Desarrollo de Bases de Datos.
  - Analizar diagrama de estructura de datos.
  - Analizar información de diccionario de datos.
  - Diseñar Base de Datos física.

- Diseñar procesos, relativos a respaldo, recuperación y reorganización de Base de Datos.
  - Consultar diseños con operación y soporte para revisión, corrección y aprobación (DBA).
- ◊ Agrupar diseño.
- Diseñar completamente estándares de datos de entrada/salida (pantallas y reportes).
  - Especificar posibles adecuaciones o cambios pertinentes a la Base de Datos creada.
  - Definir completamente el esquema estandarizado.
  - Diseñar interfaces y flujos de datos con sistemas y/o áreas periféricas.
  - Diseñar completamente reportes y pantallas que obtendrá el sistema.
  - Especificar en forma general lineamientos y procedimientos administrativos.
  - Definir procedimientos de migración o de interfaces hacia otros sistemas.
  - Asignar responsabilidades de usuarios y de sistemas.
  - Diseñar solución administrativa.
- ◊ Actualizar los planes del proyecto.
- Determinar la disponibilidad de los recursos requeridos en las actividades siguientes.
  - Estimar costo y tiempo para el desarrollo de cada una de las actividades siguientes.
  - Establecer puntos de control y fechas de revisión.
  - Determinar las actividades siguientes.

Elaborar Reporte de Especificación detallada del diseño, cuyo contenido debe ser aportado por los diferentes responsables de los diversos procesos en el desarrollo del sistema de estándares en base a los trabajos de diseño logrados por el área de Desarrollo de Sistemas. Cada uno de ellos tiene asignado su correspondiente punto particular del reporte. A continuación se expone la distribución:

- Descripción del sistema de estándares - Responsable: Atención a Usuarios
  - Objetivo.
  - Beneficios.
  - Características principales.
- Diseño conceptual del sistema. - Responsable: Desarrollo de Nuevos Productos
  - Relación de Procesos.
  - Diagrama Conceptual.
  - Filosofía de Validación/Operaciones.

- Batch.
- En línea.
- Interactivo.
- A tiempo real.
- Filosofía de diseño computacional.- Responsables: Desarrollo de Sistemas y Planeación Tecnológica.
  - Utilización de Infraestructura.
  - Uso de Archivos/B.D./Tablas.
- Diagrama de flujo de datos.- Responsable: Desarrollo de Nuevos Productos
- Descripción de procesos - Responsable: Desarrollo de Nuevos Productos
  - Descripción.
  - Puntos de control.
  - Requisitos de funcionamiento.
- Descripción de Entrada/Salida.- Responsables: Desarrollo de Nuevos Productos  
Desarrollo de Sistemas
  - Diseños.
    - Pantallas
    - Reportes
- Procedimientos manuales.- Responsable: Desarrollo de Nuevos Productos  
Implantación de Sistemas
  - Descripción.
  - Archivos manuales.
  - Puntos de control
  - Requisitos de funcionamiento.
- Procedimientos computacionales -Responsable: Desarrollo de Sistemas
  - Diagrama de los procesos del sistema.
  - Estructura de Bases de Datos.
  - Diagrama de flujo del sistema.
  - Descripción de procesos (Detallado).
  - Puntos de control.
- Documentación de Programas - Responsable: Desarrollo de Sistemas
  - Diagramas entrada/salida.
  - Objetivo de los programas.
  - Especificaciones
  - Formato de parámetros de control de fecha y opciones.
  - Descripción de validaciones.
  - Descripción de archivos.
  - Descripción de vistas lógicas.
  - Entradas y salidas referenciadas.
    - Descripción.
    - Diseño de reportes.
    - Diseño de pantallas.
    - Diseño de formatos.

- Instalación computacional
  - Plan previo de instalación
    - Responsabilidad de sistemas.
    - Responsabilidad usuario.
- Relación de módulos/bibliotecas.



### CONSTRUCCIÓN DE ESTÁNDARES DE PROGRAMACIÓN

#### Objetivo.

Realizar el diseño interno de los estándares -los que lo requieren- y construir los estándares para el lenguaje computacional específico que será utilizado, para finalmente lograr su integración en el código de los sistemas.

#### Áreas involucradas

Desarrollo de Sistemas

- ◊ Seleccionar etapas de trabajo en las labores de programación de los sistemas. Aún cuando lo que se aborda en esta actividad es lo referente a la codificación de los programas y los trabajos más inmediatamente ligados a ello, esta división de las etapas de programación se debe abordar del mismo modo que el proceso de desarrollo en general según lo indicado en el apartado "Entonces : ¿qué hacer?" al principio de este mismo capítulo.
  - Asignar personal para el trabajo en cada una de las etapas.
  - Revisar continuamente esta secuencia y alterar si hay conveniencia.
- ◊ Creación
  - Efectuar creación de los estándares para la codificación de programas:
    - Estructurado de los directorios y subdirectorios del ambiente de programación en general y lugares lógicos donde se ubicarán las aplicaciones.
    - Formato de los programas:
      - Datos del programa: en cada archivo de código de programación deben considerarse qué datos deben ir como encabezado en forma de comentario, por ejemplo: Nombre del programador, fecha de inicio del trabajo de codificación, revisiones, proyecto al que pertenece el programa, etc.

- Declaración de variables. Debe ser hecha de modo tal que una variable en cualquier sitio del programa pueda ser fácilmente identificada por cualquiera de los implicados en la codificación del programa, aún por aquellos que no han trabajado en su codificación.
- Formato de funciones en un archivo de librería funciones de programación. También las funciones auxiliares que se generen en los listados de código en un mismo archivo, deben seguir un criterio para decidir si se elabora una función o si se deben agregar líneas de código en el listado principal
- Formato de reportes e interfaces de salidas en general. Las interfaces de operación para el desarrollo de los sistemas son generalmente la cara de la eficacia que presentarán los sistemas al usuario, en donde deberán ser precisos, sin redundancias en la información o datos que presenten y útiles para el objetivo que se ha marcado para ellos. Deberán considerarse los puntos siguientes:
  - El diseño de la interfaz deberá ser específico a cada necesidad indicada en el objetivo de las misma
  - La interfaz debe tender a generar un ambiente amigable con el usuario.
  - La cantidad de la información que se presente deberá ser la adecuada a la función del usuario o a su requerimiento.
  - La interfaz debe estar disponible donde en el tiempo y sitio preciso en que se le requiera, optando por el método correcto para las salidas y las entradas de datos e información.
- Nomenclaturas: Deberán ser elegidas en razón de la cantidad de elementos que se pretendan designar con ellas, a fin de evitar los problemas que se han indicado en los capítulos Frenos y Detonadores. Tenga presente que toda nomenclatura debe tender hacia la descripción esencial y necesaria del elemento que denomina.
  - Directorios y subdirectorios
  - Archivos en general.
  - Programas.
  - Variables.
  - Funciones.
  - Bases de Datos.
  - Tablas.
  - Campos de las tablas.

- Efectuar pruebas macro o en gran escala. En una prueba de escritorio simular el funcionamiento cada estándar en la programación mediante la elaboración de un sistema que abarque el proceso completo.
  - Reunir documentación de la actividad en un informe que abarque todas las medidas que se pretende implantar y los beneficios que aportan.
  - Elaborar herramientas de apoyo para la etapa de pruebas:
    - Transacciones de prueba.
    - Datos de prueba.
    - Programas de servicio a los usuarios.
  - Documentar variaciones si las hubiera.
  - De ser necesario, actualizar especificaciones.
- ◊ Realizar pruebas de integración a los códigos de los programas. Se realizan con el fin de que no se creen situaciones de confusión entre las nomenclaturas y de tratar de problemas no previstos en la fase de creación de los programas.
- Evaluar estándares en los aspectos siguientes:
    - Codificación estructurada: nomenclaturas de variables y funciones.
    - Parámetros de funciones o entre aplicaciones.
    - Modularidad. Referente a la interacción de los módulos de programas que integran un sistema completo con el fin de detectar anomalías que se produzcan en la identificación de los componentes del sistema en que se trabaja.
    - Simplicidad. La que debe ser conservada lo más posible en todos los procesos de programación y administración.
  - Revisar estructuras de control para variables y tipos de variable.
  - Revisar diseños de accesos a los archivos de trabajo.
  - Corregir normativas, de ser necesario.
  - Revisar efectividad.
  - Revisar resultados y condiciones de error.
  - Efectuar integración al sistema de estándares general.
- ◊ Actualizar plan del proyecto.
- Elaborar, recopilar y ordenar el detalle de la actividad construcción.
  - Obtener autorización de la siguiente fase.
  - Determinar la disponibilidad de los recursos requeridos en las actividades siguientes.

- Estimar costo y tiempo necesarios para el desarrollo de cada una de las actividades siguientes
- Establecer puntos de control y fechas de revision
- Determinar las actividades siguientes

Se generan notas sobre los procesos y estandares creados para nomenclaturas de variables, archivos y usuarios, con las combinaciones de caracteres alfanumericos y los significados de cada uno de ellos, con sus características particulares



## PRUEBAS

### Objetivo

Generar un plan de pruebas y un conjunto de pruebas de eficiencia, de trayectoria normal y de error, a partir de las especificaciones de las actividades de Analisis y Diseño

Áreas involucradas

Área usuario  
Desarrollo de sistemas  
Producción

- Generar plan de pruebas. En suma, el plan de pruebas debe revisar el funcionamiento del sistema de estándares, recuérdese que no se trata de comprobar la perfección del sistema sino la localización de problemas o fallas desconocidas en condiciones de trabajo. Estas pruebas se deben realizar de acuerdo a la naturaleza particular del sistema y sus procesos en general
- ◊ Preparar pruebas de eficiencia
  - Se establecen situaciones que someten los estándares a condiciones de trabajo que permiten evaluarlos respecto de su rapidez y facilidad de operación
- ◊ Preparar pruebas de trayectoria normal
  - Se someten los estándares a condiciones promedio de trabajo del sistema.
- ◊ Preparar pruebas de trayectoria de error.
  - Ubicar los estándares en condiciones de operación susceptibles de error y se evalúan las posibilidades de detección mediante las condiciones creadas por los estándares.
- ◊ Instalar equipo, software y comunicaciones para la realización de las pruebas.
- ◊ Actualizar plan de proyecto.

- Determinar la disponibilidad de los recursos requeridos en la siguiente actividad.
- Estimar tiempo y costo necesario para el desarrollo de la siguiente actividad.
- Establecer puntos de control y fechas de revisión.
- Determinar actividades siguientes.



### CONTROL DE CALIDAD

**Objetivo:**

Llevar a cabo el plan de pruebas.

**Áreas involucradas:**

Área usuaria  
Desarrollo de sistemas.  
Operación.

◇ Ejecución de pruebas

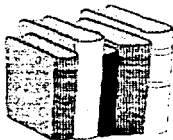
- Integrar estándares a paquetes de aplicación y programas externos.
- Realizar pruebas de eficiencia.
- Realizar prueba total del sistema de estándares, emulando operación normal.
- Realizar pruebas de trayectoria de error.
- Validar resultados y corregir desviaciones.
- Documentar desviaciones.
- Actualizar los documentos relativos al diseño de estándares, con cambios efectuados.
- Validar y corregir cambios para que sean integrados en los manuales de operación.
- Validar plan de capacitación a usuarios.
- Documentar resultados de pruebas en reportes breves que especifiquen los aspectos evaluados y sus resultados.

◇ Revisión de pruebas. Se presentan los resultados de las pruebas para evaluación general.

- Efectuar presentación de resultados de pruebas.



- Efectuar presentación de cambios al diseño original y deficiencias propias del sistema de estándares.
- ◊ **Corregir deficiencias resultantes.**
  - Identificar causas y origen de las deficiencias.
  - Efectuar correcciones.
  - Ejecutar de nuevo el control de calidad.
- ◊ **Aprobación de los resultados de las pruebas.**
  - Obtener autorización de implementación del sistema de estándares por el usuario, en la inteligencia de que el usuario debe de aprobar la conveniencia del adecuado funcionamiento de los estándares.
- ◊ **Actualizar los planes del proyecto.**
  - Determinar la disponibilidad de los recursos requeridos en la siguiente actividad.
  - Estimar costo y tiempo necesarios para el desarrollo de la siguiente actividad.
  - Establecer puntos de control y fechas de revisión.
  - Determinar actividades siguientes.



### DESCRIPCIÓN DE PROCEDIMIENTOS

Objetivo:

Obtener una descripción formal de las partes manuales del nuevo sistema de estándares y como van a interactuar los usuarios con la parte automatizada, elaborando los manuales e instructivos necesarios.

Áreas involucradas

Desarrollo de Nuevos Productos  
 Implantación de Sistemas

- ◊ **Construir soporte administrativo.**
  - Elaborar en documentos los lineamientos, políticas y procedimientos.
  - Obtener revisión y aprobación de documentos.
  - Tramitar firmas de autorización de documentos.
- ◊ **Desarrollar manuales de usuario y operación.**

- Elaborar manuales de usuario y operación.
  - Fijar el plan de capacitación.
  - Elaborar guías generales del sistema de estándares.
- ◊ Actualizar los planes del proyecto.
- Determinar la disponibilidad de los recursos requeridos para la siguiente actividad.
  - Estimar el tiempo y costo necesarios para el desarrollo de la siguiente actividad.
  - Establecer puntos de control y fechas de revisión.
  - Determinar las siguientes actividades.



#### CONVERSIÓN DE DATOS

Objetivo:

Preparar los datos actuales para el nuevo ambiente.

Áreas involucradas:

Área Usuaría.  
Desarrollo de Sistemas  
Operación.

- ◊ Elaborar plan de conversión.
- Definir requerimientos de interlace o migración del sistema de estándares actual. Es decir se deben identificar los aspectos en el sistema de estándares actual que deberán ser emigrados y su orden de conversión.
  - Especificar estrategias de conversión. Se deben utilizar las estrategias de conversión de datos fijadas para el desarrollo de sistemas, las cuales se pueden encontrar en las obras especializadas en el desarrollo de sistemas de información.
  - Elaborar plan de conversión con fundamento en la estrategia de conversión elegida.
  - Definir requerimientos de recursos humanos de sistemas y del usuario para ejecutar la conversión.
  - Establecer responsabilidades del usuario y de sistemas, si es necesario.

- Buscar asignación de tiempo, prioridad y equipo para la ejecución de la conversión. Se debe de conservar la secuencia de la conversión en todo momento acorde con el plan.
- ◊ Ejecutar la conversión.
  - Procesar y revisar el sistema de carga inicial. Entendiéndose por esto el primer bloque de estándares que se han implantado.
  - Ejecutar inicialización de archivos y Bases de Datos con los nuevos estándares.
  - Revisar carga inicial de conversión de datos a los nuevos estándares.
  - Coordinar operación de ser necesario.
  - Validar resultados con el usuario.
  - Obtener aprobación de operación satisfactoria del usuario.
- ◊ Actualizar los planes del proyecto
  - Determinar la disponibilidad de los recursos requeridos en la siguiente actividad.
  - Estimar tiempo y costo necesario para el desarrollo de la siguiente actividad.
  - Establecer puntos de control y fechas de revisión.
  - Determinar actividades siguientes.



### IMPLANTACIÓN

**Objetivo:**

Elaborar el plan de actividades para llevar a cabo la implementación del sistema de estándares.

**Áreas involucradas:**

Área Usuaría  
Operación  
Desarrollo de Sistemas

- ◊ Implantar el sistema. Los puntos de trabajo que se indican en esta actividad se hallan en estrecha relación con la estrategia de conversión de datos que se halla elegido, de modo

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

que, los puntos que se indican a continuación pueden ser variables de acuerdo al criterio del equipo que encabece la implantación de los estándares.

- Difundir los lineamientos y procedimientos. (Manuales y Documentos informativos)
  - Revisar el cumplimiento de estándares y su adecuada ubicación en el sistema.
  - Implantar el nuevo sistema de estándares en toda el área de trabajo general.
  - Efectuar inicialización del sistema de estándares.
  - Coordinar actividades de inicio con el usuario.
- ◊ Capacitar a usuarios y operadores considerando las diferentes estrategias de capacitación ya sea por casas de capacitación comerciales o dentro de la misma Área de Sistemas asignando al personal adecuado.
- Efectuar última revisión al plan de capacitación.
  - Efectuar capacitación.
- ◊ Iniciar aplicación del sistema de estándares.
- Obtener aprobación para suspensión al sistema de estándares anterior.
  - Respalda la documentación del sistema de estándares anterior.
- ◊ Obtener la aceptación del sistema de estándares en operación.
- Validar resultados con el usuario.
  - Obtener aprobación de alta del sistema de estándares, en el área de trabajo.
- ◊ Elaborar reportes finales.
- Documentar lista de distribución de manuales, resultado de la capacitación e instalación.
  - Elaborar reporte de operación normal registrando las mejoras previstas y aspectos de trabajo que no presenten deficiencias.
  - Documentar costo real del proyecto.
- ◊ Afinar sistema de estándares.
- Monitorear el rendimiento del sistema de estándares.
  - Monitorear el costo de operación en tiempo y esfuerzo además del costo económico.
  - Analizar puntos débiles dentro del sistema de estándares, es decir, los aspectos que presenten susceptibilidad de fallas notables.
  - Corregir desviaciones que estén dentro del alcance original.
  - Actualizar estudio costo/beneficio.

- ◊ Obtener evaluación final del sistema de estándares. Para esta evaluación es necesario antes establecer los parámetros de desempeño:
  - Documentar resultados de actividad de afinación.
  - Efectuar comparación de costos reales contra costos planeados.
  - Elaborar presentación al usuario para integrarlo al uso de los nuevos lineamientos de trabajo.
  - Efectuar presentación y oficializar la terminación del proyecto.
  - Solicitar el llenado de la forma de evaluación del proyecto. Esta forma deberá ser elaborada tomando como base los aspectos de rendimiento fundamentales para el cumplimiento de los objetivos inicialmente planteados, pueden ser cuestionarios entrevistas.
  - Elaborar evaluación final:
    - Operación del Sistema Computacional.
    - Funcionamiento y claridad del Sistema Administrativo.

**Realizar el Reporte de Evaluación del Sistema de Estándares**

**Conclusiones**

- Problemas y/o áreas de oportunidad detectadas.
- Grado de cumplimiento de los objetivos establecidos inicialmente.

**Recomendación de cambios (inmediatos y posteriores) en cuanto a:**

- Mantenimiento
  - Correctivo
  - Adaptativo a posibles nuevas condiciones de trabajo.
  - Mejoras
- Ajustes al calendario de trabajo y frecuencia de procesos de desarrollo de los productos que crea el Área de Sistemas.
- Integración/desglose de los nuevos procesos.
- Optimización de costos de operación.
- Optimización de costos de respuesta.

Deberá ser elaborado por los responsables del Desarrollo de Nuevos Productos.

**DOCUMENTOS GENERADOS**

Manual de Operación.

Manual de Usuario.

Procedimientos de Control.

Esquemas de Formatos Programación.

Con el objeto de mantener actualizados y óptimos los estándares implementados, el personal que labore con los estándares tanto en su diseño como en su utilización pueden aportar las sugerencias que consideren necesarias, afín de adecuarla a las circunstancias de trabajo específicas de su cambiante contorno de trabajo.

Estas sugerencias deben dirigirse por escrito a la dirección del Área de Sistemas.

Cada sugerencia debe ser analizada por los implicados en el diseño de los sistemas de estándares y, de proceder, deben ser integradas a las actualizaciones difundiendo la información de tal evento a todos los usuarios potenciales de esta nueva norma.

Dependiendo de lo trascendente de la actualización o del volumen de actualizaciones los directores del Área de Sistemas deciden cuando debe ser renovada en su totalidad la gama de actualizaciones o si es adecuado evaluar la posibilidad de implantar un nuevo sistema de estándares.

## CONCLUSIÓN

Los estándares son un medio para facilitar el trabajo. El trabajo no sólo en lo relativo a los sistemas de información o de programación, sino en casi todo lo que conforme actividades sociales, de hecho, las leyes asentadas en códigos, que rigen nuestra conducta social y hacen posible la convivencia pacífica, son todas ellas estándares de comportamiento social. Existen estándares en la forma de vestir en todos los grupos humanos. Los modales en la mesa, de alguna manera son estándares del buen gusto. Hay estándares de muchas clases porque hay actividades de muchas clases. Para que algo funcione mejor, no necesariamente se debe estandarizar, sino determinar si es necesario hacerlo.

En un sistema o proceso informático siempre participan innumerables circunstancias y factores, ninguno de ellos es inútil o intrínsecamente irrelevante, una buena estandarización debe de contemplarlos todos sin excepción porque la optimización de un sistema se halla en relación con todos los factores que le afectan y los que el sistema modifica de su entorno.

En la gran mayoría de los procesos de desarrollo de sistemas informáticos o de estándares para los mismos, es fundamental la dirección y la adecuada ejecución del Estudio Preliminar, ya que es el pie de todo el proceso de desarrollo posterior, además, es esencial para conservar la visión de lo que se quiere lograr con la implantación de los estándares, es decir, las actividades son dirigidas en el sentido de que tienden a hacia un único objetivo presente en las mentes de todos los que participan en su elaboración. Las actividades que se desarrollan después del Estudio Preliminar no son menos importantes pero, es menester considerar que cuando alguno de ellos llega a fallar en su cometido, es factible que se puedan redirigir en un tiempo relativamente breve, siempre y cuando no participen factores o circunstancias imprevistas o extraordinarias. Esta metodología, entre otras cosas, es obsequiosa en cuanto a la linealidad de los objetivos que se persiguen con la implantación de los estándares y afianza el objetivo previsto en cada uno de los procedimientos.

Cuidar de un aspecto y no del otro es a veces esterilizar el primero y malograr el segundo. Un sistema informático es un mundo en pequeño, sus posibilidades son muchas y

muy diversas, necesita armonía y esta no puede existir sin atinada combinación y no hay combinación atinada si cada cosa no está en su lugar, si no ejerce sus funciones o las suspende en el tiempo oportuno. Cuando un sistema deja sin utilizar una de sus funciones, el sistema es una herramienta a la que le faltan piezas, cuando las emplea mal, es una herramienta destemplada. La estandarización es fría pero ve claro, quitarle límites sin perder de vista lo que se desea con ello. La metodología de estandarización sometida al juicio prudente de la situación y todo documentado según su complejidad, dirigido. He aquí un sistema optimizado, como un sistema viviente y generado por selección natural, bien adaptado a su medio, haciendo de su cometido un éxito.

## GLOSARIO



**Acceso de usuario común.** El estándar que deben seguir todas las aplicaciones de *software* diseñadas.

**Administración de recursos de información.** Un concepto cuyo objetivo es conseguir que la información sea manejada como recurso corporativo.

**Administrador de Bases de Datos (DBA Database administrator).** El individuo responsable del mantenimiento físico y lógico de la base de datos.

**Algoritmo.** Un procedimiento que se puede usar para resolver un problema particular.

**Almacenamiento de información.** Una instalación de almacenamiento central basado en la computadora para toda la información de diseño de sistemas (se conoce también como diccionario o enciclopedia).

**Amigable.** Referente a un sistema en línea que permite que una persona que tiene relativamente poca experiencia interactúe de manera exitosa con el sistema.

**Análisis de sistemas.** El análisis, el diseño, el desarrollo y la puesta en marcha de sistemas de información con base en la computación.

**Analista de sistemas.** Una persona que realiza análisis de los sistemas.

**Aplicación.** Un problema o tarea en que se puede aplicar la computación.

**Archivo ASCII.** Un archivo de texto genérico que se desarrolla a partir de los caracteres de control específicos del programa.

**Archivo de respaldo.** Duplicado de un archivo existente.

**Archivo de transacción.** Un archivo que contiene registros de actividad de datos (transacciones); se usa para actualizar el archivo maestro.

**Archivo.** 1) Una recopilación de registros relacionados. 2) Un área denominada en un dispositivo de almacenamiento secundario que contiene un programa, datos o material textual.

**Arquitectura.** El diseño de un sistema de computación.

**ASCII. (American Standard Code for Information Interchange).** Un sistema de codificación.

**Aseguramiento de la calidad.** Un área de especialidad que se encarga de la supervisión de la calidad de todos los aspectos del diseño y la operación de sistemas de información.

**Atributo.** Un campo de la base de datos relacional.

**Base de datos relacional.** Véase *DBMS relacional*.

**Base de datos.** 1) El recurso de datos de una organización para todo el procesamiento de la información con base en la computación donde están integrados y relacionados para reducir al mínimo la redundancia de datos. 2) Un término alternativo para el *software* de administración de datos basada en la microcomputación. 3) Lo mismo que en un archivo en el contexto del uso de la microcomputadora.

**C.** Un lenguaje de programación transportable que se puede usar para desarrollar *software* C++.

**C++.** Una versión del lenguaje de programación C orientada a objetos.

**Cámpo.** La unidad lógica de datos más pequeña. Como ejemplos se pueden citar el número de empleado, el apellido y el precio (Comparese con *elemento de datos*).

**Campo clave.** El campo de un registro que se usa como un identificador del acceso, la salida y la recopilación de registros.

**Capacitación basada en computadora (CBT).**

Uso de las computadoras para la capacitación y la educación.

**Captura de datos.** La transcripción de datos de fuente en un formato accesible para la máquina.

**Capturista.** Una persona que utiliza dispositivos de captura por teclado para transcribir datos en un formato accesible para la máquina.

**Carga.** La transmisión de datos de una PC o una terminal a la computadora mainframe.

**Cargar.** Transferir programas o datos de un almacenamiento secundario a uno primario.

**Centro de información.** Una instalación en que se ponen a la disposición recursos de computación a varios grupos de usuarios.

**Centro de servicio.** Una compañía que ofrece casi todo tipo de servicios de procesamiento de información por una cuota.

**Ciclo de vida del sistema.** Una referencia a las cuatro etapas de un sistema de información basado en la computación: creación, desarrollo, producción y conclusión.

**Ciclo.** Una secuencia de instrucciones del programa que se ejecutan en forma repetitiva hasta que se cumple una condición particular.

**Codificación.** Aplicación de las reglas de un código.

**Código.** 1) Las reglas que se usan para convertir una configuración en bits en caracteres alfanuméricos. 2) El proceso de recopilación

de instrucciones de computación en forma de un programa de computación. 3)

**El programa de computación actual**

**Código de programación.** Véase *código*

**Código fuente.** Véase *programa fuente*

**Comando.** Una instrucción dirigida a una computadora que invoca la ejecución de una secuencia de instrucciones programada previamente.

**Compatibilidad.** 1) Referente a la capacidad de una computadora para ejecutar programas de otra computadora y acceder a la base de datos de la misma a la vez que se comunica con esta. 2) Referente a la capacidad de un dispositivo de *hardware* particular para tener una interfaz con una computadora particular.

**Compilador.** Sistema de *software* que realiza el proceso de compilación.

**Compilar.** Traducir un lenguaje de programación de alto nivel, como COBOL, en lenguaje de máquina para preparar la ejecución.

**Computación personal.** Un entorno de computación en el que los individuos usan microcomputadoras tanto para aplicaciones domésticas como comerciales.

**Computadora.** Un instrumento electrónico capaz de interpretar y ejecutar comandos programados para entrada, salida, cómputo y operaciones lógicas.

**Comunicaciones de datos.** La recopilación y la distribución de la representación electrónica de la información desde y hacia instalaciones remotas.

**Conectividad.** Referente al grado en que los dispositivos de hardware, el software y las bases de datos se pueden relacionar funcionalmente entre sí.

**Configuración.** La computadora y sus dispositivos periféricos.

**Conjunto.** Un concepto de bases de datos de red que sirve para definir la relación entre dos registros.

**Conversión piloto.** Un método para la conversión del sistema por medio de la cual se instala el nuevo sistema en conversión paralela, directa o en fases como un sistema piloto solo en una de varias áreas para las que se destina.

**Coordinador con el usuario.** Una persona que actúa como una interfaz técnica entre el departamento de servicios de información y el grupo de usuarios.

**Datos.** Representaciones de hechos. Materia prima de la información.

**Datos fuente.** Datos originales que normalmente implican el registro de una transacción o la documentación de un evento o concepto.

**DBMS.** Véase *sistema de administración de base de datos*.

**Departamento de servicios de información.** La entidad organizativa que desarrolla y mantiene sistemas de información con base en la computación.

**Diagrama de flujo de datos (DFD).** Una técnica de diseño que permite la documentación de un sistema o programa en varios niveles o en generalidad.

**Diseño de sistema general.** La parte del proceso de desarrollo de sistemas en que se define de manera general el sistema objetivo.

**Diseño detallado del sistema.** La parte del proceso de desarrollo de sistemas en que se define de manera detallada el sistema objetivo.

**Disposición física.** Una especificación detallada de entrada y/o salida que ilustra detalladamente el sitio exacto en que se debe ubicar/ introducir la información en una pantalla VDT o localizar en una salida impresa.

**Entorno operativo.** 1) Una interfaz de DOS amigable. 2) Las condiciones en que funciona el sistema de una computadora.

**Equipo periférico.** Cualquier dispositivo de *hardware* distinto del procesador.

**Ergonomía.** El estudio de las relaciones entre las personas y las máquinas.

**Error.** Un error de lógica o de sintaxis en un programa, un error de lógica en el diseño de un sistema de computación o una falla en el *hardware*.

**Error sintáctico.** Un formato inválido para una instrucción del programa.

**Especialista en centros de información.** Alguien que trabaja con los usuarios de un centro de información.

**Especialista en comunicaciones de datos.** Una persona que diseña e instala redes de computadoras.

**Especificación previa.** Un método para el desarrollo de sistemas de información por medio del cual los usuarios determinan sus necesidades de procesamiento de información durante las etapas iniciales del proyecto, respetando estas especificaciones a través de la instauración del proyecto.

**Especificaciones del sistema.** Necesidades de procesamiento de información definidas por el

usuario para un sistema de información propuesto.

**Especificaciones funcionales.** Especificaciones que describen la lógica de un sistema de información desde la perspectiva del usuario.

**Evaluación posterior a la puesta en marcha.** Un examen crítico de un sistema basado en computación después de que se ha puesto en marcha productiva.

**Función.** Una operación definida con anticipación que realiza operaciones matemáticas, lógicas, estadísticas, financieras y de cadenas de caracteres en una hoja de cálculo o una base de datos.

**Hardware.** Los dispositivos físicos que comprenden un sistema de computación.

**Información.** Datos que se han recolectado y procesado de manera significativa.

**Ingeniería de sistemas.** Véase *Ingeniería de software*.

**Ingeniería de software.** Un término acuñado para enfatizar un método para el desarrollo de software que integra la exactitud de la disciplina de la ingeniería.

**Instrucción.** Un enunciado de lenguaje de programación que especifica una operación de computación particular que se debe realizar.

**Interactiva.** Relativa a la comunicación en línea e inmediata entre el usuario terminal y la computadora.

**Interfaz de usuario.** Una referencia al software, el método o los despliegues que permiten la interacción entre el usuario y el software de aplicaciones o sistema que se está usando.

**Lenguaje de cuarta generación (4GL).** Un lenguaje de programación que usa instrucciones de alto nivel similares al inglés para recuperar y dar formato a datos para consultar peticiones e informes.

**Lenguaje de programación de alto nivel.** Un lenguaje con instrucciones que combinan varias instrucciones a nivel máquina en una instrucción.

**Lenguaje de programación de bajo nivel.** Un lenguaje que comprende el conjunto fundamental de instrucciones de una computadora particular.

**Lenguaje de programación.** Un lenguaje que los programadores usan para comunicar instrucciones a una computadora.

**Mantenimiento del sistema.** El proceso de modificación de un sistema de información para cubrir las necesidades variables.

**Memoria.** Véase *RAM*.

**Metodología del desarrollo de sistemas.** Procedimientos estandarizados escritos que describen las actividades del proceso de desarrollo de sistemas y definen las responsabilidades individuales y de grupo.

**Módulo.** Una tarea en un programa que es independiente de otras tareas.

**Opciones por omisión.** Opciones de software establecidas previamente que se supone válidas a menos que el usuario especifique lo contrario.

**Operador.** La persona que realiza actividades basadas en el hardware que son necesarias para mantener el nivel operativo de los sistemas de información.

**Operador de computadora.** Quien realiza tareas asociadas con la operación de rutina de una minicomputadora, una macrocomputadora o un sistema de supercomputación.

**Operador de sistema.** Véase *sysop*.

**Operadores lógicos.** Se usan para combinar expresiones relacionales lógicamente en una hoja de cálculo y en software de base de datos (tales como AND, OR ). Véase también *Operadores relacionales*.

**Operadores relacionales.** Se usan en fórmulas de hojas de cálculo y bases de datos para representar la relación de igualdad entre dos expresiones ( $=$  igual que,  $<$  menor que,  $>$  mayor que,  $<=$  menor o igual que,  $>=$  mayor o igual que,  $<>$  diferente de). Véase también *Operadores lógicos*.

**Paquete de software.** Uno o más programas diseñados para realizar una tarea de procesamiento particular.

**Procesamiento de datos.** Uso de la computadora para realizar operaciones con datos.

**Programa.** 1) Instrucciones de computación estructuradas y ordenadas, de manera tal, que, cuando se ejecutan, hacen que una computadora realice una función particular. 2) La acción de producir software de computación. (Se relaciona con software).

**Programa fuente.** El código del programa original. (Se conoce también como código fuente; compárese con programa objeto.)

**Programa objeto.** Un programa a nivel objeto que resulta de la compilación de un programa

fuente. (Comparese con programa fuente.)

**Programación estructurada.** Técnica de diseño por modo de la cual la lógica de un programa se dispone de manera jerárquica en módulos lógicos.

**Programación.** La acción de escribir un programa de computación.

**Programador.** Alguien que programa el uso de los recursos de hardware para optimizar la eficiencia del sistema.

**Programador de aplicaciones.** Un programador que escribe programas para aplicaciones empresariales específicas.

**Programador de sistemas.** Un programador que desarrolla y mantiene el software del sistema.

**Programador/analista.** El título de quien realiza tanto funciones de programación como de análisis de sistemas.

**Pseudocódigo.** El código no ejecutable de un programa que se usa como una ayuda para desarrollar y documentar programas estructurados.

**RAM (Random-Access Memory; memoria de acceso aleatorio).** El área de la memoria en que deben residir todos los programas y datos antes de que se puedan ejecutar los programas o manejar los datos (Es lo mismo que memoria principal, memoria y almacenamiento primario).

**Red de computadoras.** Una integración de sistemas de computadoras, terminales y medios de comunicación.

**Registro de base de datos.** Datos relacionados que se leen de (o se escriben de) la base de datos como una unidad.

**Relación.** Un conjunto de campos relacionados (como una relación de empleados) que describe un evento o un concepto (se conoce también como registro lógico).

**Respaldo.** Referente al equipo, los procedimientos o las bases de datos que se pueden usar para reiniciar el sistema en caso de un mal funcionamiento del mismo.

**Servicio de información.** Una red comercial en línea que permite el acceso a usuarios remotos a una variedad de servicios de información (es lo mismo que red de información).

**Sintaxis.** Las reglas que rigen la formulación de instrucciones en un programa de computación.

**Sistema.** Cualquier grupo de componentes (funciones, gente, actividades, eventos y demás factores pertinentes) que tienen una interfaz con otros, con los cuales se complementan, para

lograr uno o más objetivos definidos con anticipación.

**Sistema de administración de base de datos.** Un paquete de software de sistema para la creación, el manejo y el mantenimiento de la base de datos.

**Sistema de computación departamental.** Sistemas de computación que se usan como sistemas independientes de apoyo para un departamento específico, al igual que como parte de una red de minicomputadoras departamentales, todas conectadas a una computadora centralizada grande.

**Sistema de información.** Un sistema basado en la computación que ofrece tanto una capacidad de procesamiento de datos como la información necesaria para la toma de decisiones gerenciales.

**Software.** Los programas que se usan para dirigir las funciones de un sistema de computación.

**Software de aplicaciones.** Software diseñado y escrito para realizar tareas específicas personales, empresariales o de procesamiento.

**Software de bases de datos.** Software que permite que los usuarios logren crear y mantener una base de datos y extraigan información de la misma.

**Sysop (System operator; operador de sistema).** El promotor que ofrece apoyo de hardware y software para un sistema electrónico de avisos.

**Transacción.** Un evento procedural en un sistema que requiere de actividad manual o basado en computación.

**Tupla.** Un grupo de campos relacionados (un renglón) en una base de datos relacional.

**Usuario final.** El individuo que introduce datos a la computadora o que usa la salida de la computadora (es lo mismo que usuario).

**Usuario.** Véase *Usuario final*.

**Variable.** Una localización de almacenamiento primario que puede adoptar diferentes valores numéricos o alfanuméricos.

## BIBLIOGRAFÍA

- Alfaro, J. (1997). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2000). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2001). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2002). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2003). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2004). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2005). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2006). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2007). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2008). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2009). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2010). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2011). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2012). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2013). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2014). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2015). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2016). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2017). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2018). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2019). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2020). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2021). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2022). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2023). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2024). *El mundo de los niños*. Madrid: Alianza.
- Alfaro, J. (2025). *El mundo de los niños*. Madrid: Alianza.

