



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

SISTEMA CONTABLE FINANCIERO PARA PLATAFORMAS LINUX

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN INFORMÁTICA

**P R E S E N T A:**

ROSA VERÓNICA RUÍZ REYES

ASESOR: L. C. CARLOS PINEDA MUÑOZ

CUAUTITLAN IZCALLI, EDO. DE MEX.

2005

m 344959





Universidad Nacional  
Autónoma de México



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

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

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

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



SECRETARÍA NACIONAL  
AVENIDA DE  
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
UNIDAD DE LA ADMINISTRACION ESCOLAR  
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
UNIDAD DE LA ADMINISTRACION ESCOLAR  
DEPARTAMENTO DE EXAMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO  
DIRECTOR DE LA FES CUAUTITLAN  
PRESENTE

ATN: Q. Ma. del Carmen Garcia Mijares  
Jefe del Departamento de Exámenes  
Profesionales de la FES Cuautitlán

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicarle a usted que revisamos la TESIS:

Sistema Contable Financiero para plataformas linux

que presenta la pasante: Rosa Verónica Ruíz Reyes  
con número de cuenta: 9214678-7 para obtener el título de  
Licenciada en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

ATENTAMENTE  
"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 26 de mayo de 2004

PRESIDENTE	<u>Ing. Sergio Acosta Torres</u>	
VOCAL	<u>L.C. Carlos Pineda Muñoz</u>	
SECRETARIO	<u>L.A. Guadalupe Ayvar Cebrero</u>	
PRIMER SUPLENTE	<u>L.I. Armando Carmona Bonilla</u>	
SEGUNDO SUPLENTE	<u>L.C. Jaime Navarro Mejía</u>	

## **DEDICATORIA:**

### **A ti Dios Mío, por que me permitiste llegar a este punto de mi vida**

#### **A mi Madre:**

Sabiendo que jamás existirá una forma de agradecer el esfuerzo, la lucha y el cariño que me has demostrado a lo largo de mi vida, deseo expresarte que los esfuerzos y logros alcanzados hasta el momento también han sido tuyos y forman parte de uno de los éxitos más grandes que he podido alcanzar. Te amo mamá.

#### **A mi Padre:**

Dondequiera que estés, sé que me cuidas y que te alegra que haya logrado hacer realidad uno de tantos sueños.

#### **A mis Hermanas y Familia:**

Les agradezco su cariño, comprensión, apoyo y confianza. Ustedes forman parte del gran aliciente que me impulsa a seguir adelante.

#### **A Miguel Angel:**

Por el apoyo, amor, paciencia y confianza que siempre me has demostrado, que me ayudo a que esta vida de lucha sea menos pesada, te dedico este trabajo con admiración, cariño y respeto para ti.

#### **A Elvia:**

Mi mejor amiga, sin ti no lo hubiera logrado, gracias por tu amistad y tu cariño, recuerda que siempre cuentas conmigo.

#### **A mi Asesor de Tesis:**

Por todo el apoyo, el tiempo y la paciencia aportada para la culminación del Proyecto de titulación.

# INDICE

## CONTENIDO .....

INTRODUCCIÓN	1
OBJETIVOS	2
ALCANCE DEL SISTEMA	3
HERRAMIENTAS DE DESARROLLO	5
RESULTADOS ESPERADOS	7
<b>CAPITULO 1. GENERAL PUBLIC LICENSE</b>	
1.1 Qué es General Public License.	8
1.1.1 Licencia Publica General de GNU.	11
1.1.2 Cómo aplicar los términos de la GPL.	18
1.2 Ventajas de GPL en el Ambiente corporativo.	18
1.2.1 Usuarios de este tipo de Software.	20
1.3 Futuro de la Plataforma GPL. Copyleft para todos.	21
1.3.1 Concepto de Open software Source OSS.	22
1.3.2 Concepto de Freeware.	23
1.3.3 Tipos de Freeware.	23
1.3.4 Concepto de Software Propietario.	25
1.3.5 OSS y la competencia en el sector de la informática.	25
1.3.6 OSS como reacción ante el corporativismo.	26
1.3.7 Otras características de la GPL.	27
1.3.8 El problema de las patentes. ¿Por qué constituyen una amenaza para el OSS?.	27
1.3.9 Comercialización del OSS.	28
1.3.10 Servicios relacionados con el OSS.	29

1.4	Tecnología Intranet.	30
1.4.1	Qué es una Intranet.	31
1.4.2	Internet / Intranet ¿Cuál es la diferencia?	32
1.4.3	Cómo funciona una Intranet.	33
1.4.4	Qué es lo que necesitamos para tener una Intranet.	33
1.4.5	Qué beneficios aporta una Intranet a la empresa.	35
1.4.6	Qué servicios añade una Intranet a la empresa.	37
1.4.7	Servicios Adicionales que aporta una Intranet a la empresa	38
1.4.8	¿ Qué procedimientos nos llevan a la realización óptima de una Intranet.	40
1.4.9	Ventajas genéricas de las intranets.	41
1.4.10	Aplicaciones de una Intranet.	41
1.4.11	Cómo mantener la Intranet segura.	42
1.4.12	Perspectivas de futuro de las intranets.	43
1.4.13	Nuevas Tecnologías de la Información.	44
1.4.14	Recursos Intranet.	45
1.4.14.1	Acceso a Base de Datos.	48
1.4.15	Los nuevos Usos de La Intranet.	48
1.4.15.1	Una Nueva Forma de Trabajar.	49
1.4.15.2	Aplicaciones de las Intranets para las empresas	50
1.4.15.3	Características Funcionales.	52
1.4.15.4	Requisitos Técnicos	53
1.4.15.5	Facilitar la Capacitación y compartir el Conocimiento.	53
1.4.15.6	Automatizar Tareas de Rutina	54
1.4.15.7	Proveer Fácil acceso a la información.	54

## CAPITULO 2. ANÁLISIS Y DISEÑO DE SISTEMAS.

2.1	Introducción.	55
2.2	Análisis de Información.	56
2.3	Clasificación del Análisis de Información	60
2.3.1	Análisis del Modelo Funcional	60
2.3.2	Análisis del Modelo de Datos	62
2.3.3	Diagrama de Estructura de Datos	62
2.3.4	Carta de Entidades	64
2.3.5	Carta de Burbujas	64
2.3.6	Esquema General del Sistema	65
2.3.7	Mapa de Acceso Lógico	66
2.3.8	Mini especificación	67
2.3.9	Formato de Pantalla y Reportes	68
2.3.10	Otras Técnicas	70
2.3.11	Análisis de Transacciones	71
2.3.12	Análisis del uso de los Datos	74
2.4	Características y Requerimientos del Sistema	75
2.5	Metodología.	77
2.5.1	Ciclo de Vida clásico del Desarrollo de Sistemas.	78
2.5.1.1	Investigación Preliminar	79
2.5.1.2	Determinación de los requerimientos del sistema.	80
2.5.1.3	Diseño del sistema.	81
2.5.1.3.1	Criterios Técnicos para el Diseño	83
2.5.1.3.2	Diseño de la Salida	84
2.5.1.3.3	Diseño de Archivos	85
2.5.1.3.4	Diseño de Interacciones con la Base de Datos	85
2.5.1.4	Desarrollo de software.	85
2.5.1.5	Prueba de sistemas.	86

2.5.1.6	Estrategias de Pruebas del Software	86
2.5.1.6.1	Prueba de Unidad	87
2.5.1.6.2	Prueba de Integración	90
2.5.1.6.3	Prueba de Validación y Verificación	93
2.5.1.6.4	Prueba de Recuperación	95
2.5.1.6.5	Prueba de Seguridad	96
2.5.1.6.6	Prueba de Resistencia	96
2.5.1.6.7	Depuración	97
2.5.1.7	Implantación y Evaluación.	99
2.5.1.7.1	Otros Métodos de Implantación	100
2.5.1.8	Mantenimiento de Software	106
2.5.1.8.1	Efectos Secundarios del Mantenimiento	109
2.5.2	Método de desarrollo por análisis estructurado.	111
2.5.2.1	Elementos del análisis estructurado.	111
2.5.2.2	Análisis estructurado con otros métodos de desarrollo.	113
2.5.3	Método del Prototipo de Sistemas.	114
2.6	Herramientas para el Desarrollo de sistemas.	116
2.6.1	Herramientas para análisis.	117
2.6.2	Herramientas para Diseño.	117
2.6.3	Herramientas para Desarrollo.	118

### **CAPITULO 3. HERRAMIENTAS DE LIBRE DISTRIBUCIÓN.**

3.1	Qué es el software Libre.	119
3.2	Cómo obtener el Software Libre.	121
3.2.1	Categorías del Software Libre y No Libre.	121
3.3	Sistema Operativo Linux. Introducción.	125
3.3.1	Historia de Linux.	126
3.3.2	Características del sistema Linux	128

3.3.3	Licencia	130
3.3.4	Fuerza del Sistema Operativo Linux	132
3.3.5	¿ Quién usa Linux ?	134
3.3.6	¿ Por qué deberíamos usar Linux ?	135
3.4	Base de Datos MySql. ¿Qué es SQL?	137
3.4.1	Breve historia de MySql.	137
3.4.2	Por qué utilizar MySql.	139
3.4.3	Principales características	139
3.5	Lenguajes de Programación.	140
3.5.1	Introducción	140
3.5.2	Breve Historia de los lenguajes	140
3.5.3	Tipos de Lenguajes	144
3.5.3.1	Lenguaje de bajo nivel	146
3.5.3.2	Lenguaje de alto nivel	146
3.5.3.3	Intérpretes y Compiladores	147
3.5.4	Breve Historia de HTML	147
3.5.5	Breve historia de PHP.	149
3.5.6	Qué es PHP.	150
3.5.7	Qué se puede hacer con PHP.	152
3.5.8	Apache Web Server	152

## CAPITULO 4. CONCEPTOS GENERALES

4.1	Concepto de Contabilidad.	155
4.1.1	Características de la información contable.	156
4.2	La información Financiera.	158
4.2.1	Definición de Contabilidad Financiera.	158
4.2.2	Utilidad de la información financiera.	158
4.3	Informes Contables	161

4.3.1	Especificación de Informes en base a datos requeridos.	162
4.3.2	Información acerca de los activos.	163
4.3.3	Información acerca de los pasivos.	164
4.3.4	Información acerca del capital contable.	164
4.3.5	Información acerca de las utilidades netas.	164
4.3.6	Información acerca de los ingresos.	164
4.3.7	Información acerca de los gastos.	165
4.4	Generalidades de los Estados Financieros.	165
4.4.1	El balance general	166
4.4.2	Formas de Presentación.	166
4.4.3	Importancia del Estado de Resultados.	167
4.4.4	Formas de presentación.	167
4.5	Objetivos de los Estados Financieros.	168
4.6	Razones Financieras.	170
<b>CONCLUSIONES</b>		<b>172</b>
<b>BIBLIOGRAFÍA</b>		<b>175</b>
<b>ANEXO1. CASO PRÁCTICO</b>		
	<b>Análisis</b>	<b>178</b>
	<b>Diseño</b>	<b>192</b>
	<b>Desarrollo</b>	

## "SISTEMA CONTABLE FINANCIERO PARA PLATAFORMAS LINUX"

### INTRODUCCIÓN

Cada vez es mayor el número de profesionales de diversas ramas, dirigentes de empresas, empleados, etc., que tienen la necesidad de entender la información financiera que emiten las empresas y sus bases de preparación. Las crisis financieras por las que el mundo en general, y nuestro país, en particular han venido atravesando en forma periódica han despertado una muy generalizada necesidad de interpretar los hechos económicos y financieros que se difunden.

Por esto el procesamiento de la información ha constituido una de las tareas básicas de cualquier civilización, debido al crecimiento económico y demográfico, existen necesidades crecientes de administrar grandes cantidades de datos relacionados con diversos fines comerciales, administrativos y científicos entre otros. Todos estos datos interrelacionados en conjunto, forman lo que se denomina un sistema.

En pleno siglo XXI, aún se observa que algunas empresas aún conservan un tradicional sistema de procesamiento de datos, el cual consiste en archivar manualmente en carpetas su información. Cabe destacar que muchas de estas empresas no adquieren un sistema que les permita automatizar sus operaciones debido al alto costo de estos, por lo que la presente tesis, nos permite dar a conocer la opción de tener un software de calidad que permita la automatización, en la obtención de la información financiera sin que requiera de una gran inversión.

Este proyecto, nos permite desarrollar un sistema para el control de la información financiera de forma general, que puede ser aplicable a cualquier tipo de empresa.

El desarrollo de la tesis se llevará a cabo a través de herramientas de Libre Distribución, lo cual demostrará que dichas herramientas nos permiten desarrollar Software de calidad.

**OBJETIVOS:**

**OBJETIVOS GENERALES:**

1. Diseño y Desarrollo de un sistema de contabilidad, basado en Software de Libre Distribución, en una Intranet.
2. Empleo de Herramientas de Libre Distribución (Linux, PHP, MySql, HTML, JavaScript)

**OBJETIVOS PARTICULARES:**

1. Manejo de información contable-financiera veraz y oportuna a través de herramientas de libre distribución.
2. Aplicar el diseño de Base de Datos relacionales para la mejor Distribución y manejo de los Datos
3. Determinar las relaciones del sistema con otras áreas académicas.
4. Desarrollar una interfaz gráfica amigable utilizando lenguajes de programación modernos en un ambiente de Intranet.

## ALCANCE DEL SISTEMA

El sistema estará dividido de la siguiente manera:

### Administración:

Solo tendrá acceso el usuario designado como administrador de la aplicación, para la configuración de la misma y para el mantenimiento de los catálogos del sistema además de realizar las consultas, capturas e impresión de reportes de la aplicación.

### Operación:

Tendrán acceso todos los usuarios designados por el administrador de la aplicación, tendrán acceso a los diferentes módulos del sistema de acuerdo con el perfil designado por el administrador.

### Estructura del Sistema:

El sistema inicialmente solo contendrá las siguientes funciones:

- **Archivo**
- **Reportes**
- **Razones Financieras**
- **Configuración**

**Archivo:** El usuario podrá realizar la captura y el mantenimiento correspondiente al catálogo de cuentas, además de realizar la captura e impresión de las pólizas.

**Reportes:** El usuario podrá hacer la impresión de los reportes auxiliares por un rango de fechas y/o un rango de cuentas, realizará la impresión del balance general y del estado de resultados en forma de reporte.

**Razones Financieras:** El usuario podrá realizar la impresión automática de la razón del ácido y la razón de liquidez, el sistema le permitirá definir sus propias formulas de acuerdo a sus necesidades.

**Configuración:** El usuario Administrador dará de alta los usuarios y definirá los perfiles para cada uno de ellos, dará el mantenimiento correspondiente al catálogo de empresas, configurará las formulas para la impresión de los estados financieros, creará los periodos, realizará los traspasos de saldos y generará los respaldos.

**Observaciones del Sistema:**

- Es de vital importancia que el sistema sea configurado por el administrador del sistema, de forma general, antes de iniciar operaciones.
- Una vez configurado. Los usuarios dados de alta pueden comenzar la captura de los catálogos y de las pólizas.
- Basándose en esta captura se podrán generar los reportes mencionados.

## HERRAMIENTAS DE DESARROLLO

Las Herramientas utilizadas para el Desarrollo del sistema son de libre distribución (GNU), por lo que se pueden obtener sin ningún costo, dada esta situación el montaje inicial del sistema y su uso no conlleva ninguna inversión. (Más detalle sobre el software de libre distribución en el Capítulo 3).

### Software

El ambiente está conformado por un servidor de base de datos y un servidor WEB.

### El Sistema está compuesto por los siguientes elementos:

- GNU/LINUX como Sistema Operativo Abierto
- Apache como Servidor WEB
- PHP como módulo de ampliación de Apache para acceder a la base de datos
- MySQL como Base de Datos
- Una interfaz WEB a MySQL

### Apache

Es el servidor de WEB por excelencia. Ha sido uno de los mayores éxitos del software libre y su supremacía entre los servidores de WEB no se ve amenazada.(ni siquiera por el servidor de Web de Microsoft).<sup>1</sup>

Su desarrollo continuo y su portabilidad han llevado a plataformas como Windows NT y Windows 95, a parte de casi todos los sistemas UNIX.

Una de las principales características de Apache es su extensibilidad basada en una gran modularidad de su código fuente, lo que ha facilitado la aparición de módulos de extensión como PHP el cual evitará

---

<sup>1</sup> Para Consultar datos estadísticos sobre el servidor Apache . <http://www.netcrafter.com>

el uso de cgi-bins por completo, facilitando enormemente la programación de aplicaciones en el lado del servidor, especialmente en el campo de acceso a bases de datos.

### **PHP: Hypertext Preprocessor (Preprocesador de Hipertexto)**

El principal objetivo del lenguaje es permitir a los desarrolladores de aplicaciones basadas en el web escribir páginas que se generan de forma dinámica, sencilla y rápida.

El PHP desde la versión 3.0 para mantener una cierta asociación con su progenitor, ha sido ampliado con aspectos tan importantes como conceptos de programación orientada a objetos. Lo más espectacular de PHP es su facilidad para acceder a las bases de datos, entre ellas MySQL.

### **MySQL**

Es un sistema de gestión de bases de datos (SGBD) SQL, sus principales objetivos han sido la velocidad y la robustez. Es un SGBD sencillo y rápido que se adapta perfectamente a entornos en donde el volumen de datos sea del orden de megabytes.

En el mundo de GNU/LINUX es MySQL junto a PostgreSQL los principales sistemas de gestión de bases de datos de uso libre y con código fuente.

MySQL esta al igual que PHP en fase de pleno desarrollo, se están publicando de forma regular nuevas versiones del sistema, así como herramientas que son básicas en cualquier SGBD actual. Se detalla más sobre la herramientas de libre distribución (Linux, MySQL, PHP y Apache) que serán utilizadas para el desarrollo del sistema dentro del capítulo 3.

## RESULTADOS ESPERADOS

- La información que suministre el sistema debe ser consistente, correcta, precisa y oportuna.
- La información debe ser proporcionada en un formato que facilite su interpretación.
- La información servirá de apoyo a las áreas contable-administrativas
- Permitirá además un mejor aprovechamiento de los recursos disponibles.
- La implementación debe ser fácil y eficiente con pocas necesidades de administración.

La finalidad principal de realizar este sistema es que las áreas involucradas en el manejo y administración de este tipo de información, se vean enormemente beneficiadas. Al contar siempre con la información de forma veraz y oportuna, que les ayude eficientemente en la toma de decisiones.

## CAPITULO 1. GENERAL PUBLIC LICENSE

### 1.1 QUE ES GENERAL PUBLIC LICENSE (GPL)

**GPL (GNU General Public License)** es un tipo de licencia sobre la propiedad intelectual en la cual únicamente se exige que aquellos desarrollos hechos con material licenciado bajo GPL sean a su vez GPL<sup>1</sup>.

La licencia Pública General (GPL) tiene como objetivo garantizar al usuario la libertad de compartir y cambiar software libre, es decir, asegurarse de que el software es libre para todos sus usuarios. Esta licencia Pública General es aplicable a la mayoría del software de la Free Software Foundation así como a cualquier otro programa cuyos autores se comprometan a usarlo.

Siendo la informática y en particular la programación una ciencia, del mismo modo que lo son la química, la física o las matemáticas aunque no todos quieran admitirlo- la verdad es que no hemos avanzado demasiado. **Paradójicamente** a lo que se cree, no existe una gran evolución en la sección informática más importante, es decir: **El software no ha evolucionado excesivamente.**

Desde la aparición hace 20 años de los PC's, el usuario siempre ha demandado lo mismo. Software que reemplace la máquina de escribir y software que reemplace la máquina registradora. Hace 20 años ya había programadores resolviendo los problemas de la pequeña y mediana empresa mediante máquinas de imprimir ticket's, programas de contabilidad, etc. Y no importa que paradigma de programación se usara y no importa cual fue la tecnología de objetos o metodología de trabajo porque ahora, 20 años mas tarde, están llegando los mismos clientes, con los mismos problemas y hay que dedicarse a resolver el mismo problema que ya habían resuelto nuestros "antepasados". Es como si un médico que encontrara una solución contra el cáncer sólo la usara para sus enfermos y no lo compartiera con el resto del mundo. Es decir, si la medicina evolucionara de la misma forma que la programación aun estaríamos curando la lepra, la peste, la malaria o las paperas.

---

<sup>1</sup> <http://www.gnu.org>

En definitiva si la **medicina evoluciona, si la física evoluciona, si la genética evoluciona ... es sobretodo porque comparten sus avances**. Saben que no tiene sentido que después de encontrada una solución tengamos que dejar que los otros pierdan 10 años más en dar la solución al mismo problema.

Definitivamente mientras sigamos con estos ideales de construcción de aplicaciones cerradas, secretas, ocultas, y sobretodo protegidas para que nadie pueda leerlas ni aprender de nuestros logros estaremos condenados al fracaso. No es posible la evolución del software si no entendemos que hemos de compartir las soluciones globales con el resto del mundo.

La licencia GPL trata de hacernos entender que el camino a seguir para la evolución pasa por la compartición de las soluciones. Por tanto, la licencia es más bien una trampa moral. Si, es un tipo de licencia sobre la propiedad intelectual en la cual únicamente se exige que aquellos desarrollos hechos con material licenciado bajo GPL sean a su vez GPL.

Si se adopta un programa bajo licencia GPL es que se ha entendido cual es el sentido real de GPL, y por lo tanto si se realiza algún desarrollo bajo esta licencia hay que compartirlo.

**Mientras el término libre ("free") implica que el código fuente es distribuido con el programa, de ninguna manera significa que la aplicación en sí misma sea "gratis" necesariamente.**

**La licencia GPL simplemente evita que un distribuidor se aproveche del trabajo llevado a cabo por una comunidad de desarrolladores, incluyendo también restricciones en su uso. Hoy la licencia GPL gobierna la gran mayoría de programas que corren bajo Linux, incluyendo el sistema operativo en sí mismo<sup>2</sup>.**

Las licencias que cubren la mayor parte del software están diseñadas para privarnos de la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizar la

---

<sup>2</sup> <http://www.gnu.org>

libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios.

Esta Licencia Pública General se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Librerías).

Cuando hablamos de software libre, definitivamente estamos refiriéndonos a libertad, no a precio. Las Licencias Públicas Generales están diseñadas para asegurarnos que tenemos la libertad de distribuir copias de software libre (y cobrar por ese servicio si así se desea), que recibamos el código fuente o que podamos conseguirlo si se requiere, también se puede modificar el software o usar fragmentos de él en nuevos programas libres.

Para proteger los derechos necesitamos que se establecieran ciertas restricciones que prohíben a terceros negarle a otros estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica.

Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente o a cambio de una contraprestación, se debe dar a los receptores todos los derechos que se adquirieron al obtener el software.

- Debe asegurarse que ellos también reciban, o puedan conseguir, el código fuente.
- Y se deben mostrar estas condiciones de forma que conozcan sus derechos.
- Protegemos sus derechos con la combinación de dos medidas:

**1. Ponemos el software bajo Copyright y**

**2. Se ofrece la licencia, que da el permiso legal para copiar, distribuir y/o modificar el software.**

También, para la protección de cada autor y la nuestra propia, queremos asegurarnos de que todo el mundo comprende que no se proporciona ninguna garantía para este software libre.<sup>3</sup>

Si el software se modifica por terceros y éste a su vez lo distribuye, es necesario que los receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.

Por último, cualquier programa libre está constantemente amenazado por patentes sobre el software. Queremos evitar el peligro de que los redistribuidores de un programa libre obtengan patentes por su cuenta, convirtiendo de facto el programa en propietario. Para evitar esto, hemos dejado claro que cualquier patente debe ser pedida para el uso libre por todo el mundo, o no ser pedida.

En definitiva, para conocer mejor bajo que condiciones queda el software desarrollado bajo licencia GPL, se exponen a continuación los términos exactos y las condiciones para la copia, distribución y modificación.

### 1.1.1 Licencia Pública General de GNU <sup>4</sup>

#### **Términos y condiciones para la copia, distribución y modificación**

0. Esta Licencia se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el poseedor del Copyright diciendo que puede ser distribuido bajo los términos de esta Licencia Pública General. En adelante, "Programa" se referirá a cualquier programa o trabajo que cumpla esa condición y "trabajo basado en el Programa" se referirá bien al Programa o a cualquier trabajo derivado de él según la ley de Copyright. Esto es, un trabajo que contenga el programa o una porción de él, bien en forma literal o con modificaciones y/o traducido en otro lenguaje. Por lo tanto, la traducción está

---

<sup>3</sup> <http://www.gnu.org>

<sup>4</sup> <http://www.gpl.com>

incluida sin limitaciones en el término "modificación". El beneficiario de esta licencia será denominado "usted".

1. Cualquier otra actividad que no sea la copia, distribución o modificación no está cubierta por esta Licencia, pues está fuera de su ámbito. El acto de ejecutar el Programa no está restringido, y los resultados del Programa están cubiertos únicamente si sus contenidos constituyen un trabajo basado en el Programa, independientemente de haberlo producido mediante la ejecución del programa. El que esto se cumpla, depende del funcionamiento del programa.

2. Usted puede copiar y distribuir copias literales del código fuente del Programa, según lo ha recibido, en cualquier medio, siempre que de forma adecuada y bien visible publique en cada copia un anuncio de Copyright adecuado y una denegación de garantía, mantenga intactos todos los anuncios que se refieran a esta Licencia y a la ausencia de garantía, y proporcione a cualquier otro receptor del programa una copia de esta Licencia junto con el Programa.

Puede cobrar un precio por el acto físico de transferir una copia, y puede, según su propia elección, ofrecer garantía a cambio de unos honorarios.

3. Puede modificar su copia o copias del Programa o de cualquier porción de él, formando de esta manera un trabajo basado en el Programa, y copiar y distribuir esa modificación o trabajo bajo los términos del apartado 1, anteriormente mostrado, suponiendo que además cumpla las siguientes condiciones:

- Debe hacer que los ficheros modificados lleven anuncios prominentes indicando que los ha cambiado y la fecha de cualquier cambio.
- Debe hacer que cualquier trabajo que distribuya o publique y que en todo o en parte contenga o sea derivado del Programa o de cualquier parte de él sea licenciada como un todo sin recargos, sin condiciones adicionales, a todas las terceras partes y bajo los términos de esta Licencia.

- Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de Copyright y un anuncio de que no se ofrece ninguna garantía (o por el contrario que sí se ofrece garantía) y que los usuarios pueden redistribuir el programa bajo estas condiciones, e indicando al usuario cómo ver una copia de esta licencia. (Excepción: si el propio programa es interactivo pero normalmente no muestra ese anuncio, no se requiere que su trabajo basado en el Programa muestre ningún anuncio).

4. Estos requisitos se aplican al trabajo modificado como un todo. Si partes identificables de ese trabajo no son derivadas del Programa, y pueden, razonablemente, ser consideradas trabajos independientes y separados por sí mismos, entonces esta Licencia y sus términos no se aplican a esas partes cuando sean distribuidas como trabajos separados. Pero cuando distribuya esas mismas secciones como partes de un todo que es un trabajo basado en el Programa, la distribución del todo debe ser según los términos de esta licencia, cuyos permisos con respecto a otras licencias se ampliarán a la totalidad del trabajo completo, y por lo tanto a todas y cada una de sus partes, con independencia de quién la escribió.

Por lo tanto, no es la intención de este apartado reclamar derechos o disputar sus derechos sobre trabajos escritos totalmente por usted mismo. El intento es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa.

Además, el simple hecho de reunir un trabajo no basado en el Programa con el Programa (o con un trabajo basado en el Programa) en un volumen de almacenamiento o en un medio de distribución no hace que dicho trabajo entre dentro del ámbito cubierto por esta Licencia.

- Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2), como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, suponiendo que además cumpla una de las siguientes condiciones:

- Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia en un medio habitualmente utilizado para el intercambio de programas.
- Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa, en formato electrónico, del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas.
- Acompañarlo con la información que recibió ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).

5. Por código fuente de un trabajo se entiende la forma preferida del trabajo cuando se le hacen modificaciones. Para un trabajo ejecutable, se entiende por código fuente completo todo el código fuente para todos los módulos que contiene, más cualquier fichero asociado de definición de interfaces, más las secuencias de comandos empleados para controlar la compilación e instalación del ejecutable. Como excepción especial el código fuente distribuido no necesita incluir nada que sea distribuido normalmente (bien como fuente, bien en forma binaria) con los componentes principales (compilador, núcleo y similares) del sistema operativo en el cual funciona el ejecutable, a no ser que el propio componente acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace mediante la oferta de acceso para copiarlo de un cierto lugar, entonces se considera la oferta de acceso para copiar el código fuente del mismo lugar como distribución del código fuente, incluso aunque terceras partes no estén forzadas a copiar el fuente junto con el código objeto.

6. No puede copiar, modificar, sublicenciar o distribuir el Programa excepto como prevé expresamente esta Licencia. Cualquier intento de copiar, modificar, sublicenciar o distribuir el Programa de otra forma

es inválida, y hará que cesen automáticamente los derechos que le proporciona esta Licencia. Sin embargo, las licencias correspondientes a terceros que hayan recibido copias o derechos de usted no cesarán en sus derechos mientras estos continúen cumpliéndola.

- No está obligado a aceptar esta licencia, ya que no la ha firmado. Sin embargo, no hay nada más que le proporcione permiso para modificar o distribuir el Programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. Por lo tanto, si modifica o distribuye el Programa (o cualquier trabajo basado en el Programa), está indicando que acepta esta Licencia para poder hacerlo, y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.
- Cada vez que redistribuya el Programa (o cualquier trabajo basado en el Programa), el receptor recibe automáticamente una licencia del licenciatario original para copiar, distribuir o modificar el Programa, de forma sujeta a estos términos y condiciones. No puede imponer al receptor ninguna restricción más sobre el ejercicio de los derechos aquí garantizados. No es usted responsable de hacer cumplir esta licencia por terceras partes.

7. Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitada a asuntos relacionados con patentes) se le imponen condiciones (ya sea por mandato judicial, por acuerdo o por cualquier otra causa) que contradigan las condiciones de esta Licencia, ello no le exime de cumplir las condiciones de esta Licencia. Si no puede realizar distribuciones de forma que se satisfagan simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación pertinente, entonces, como consecuencia, no puede distribuir el Programa de ninguna forma. Por ejemplo, si una patente no permite la redistribución libre de derechos de autor del Programa por parte de todos aquellos que reciban copias directa o indirectamente a través de usted, entonces la única forma en que podría satisfacer tanto esa condición como esta Licencia sería evitar completamente la distribución del Programa.

Si cualquier porción de este apartado se considera inválida o imposible de cumplir bajo cualquier circunstancia particular ha de cumplirse el resto y la sección por entero ha de cumplirse en cualquier otra circunstancia.

No es el propósito de este apartado inducirle a infringir ninguna reivindicación de patente ni de ningún otro derecho de propiedad o impugnar la validez de ninguna de dichas reivindicaciones.

Este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, que se realiza mediante prácticas de licencia pública. Mucha gente ha hecho contribuciones generosas a la gran variedad de software distribuido mediante ese sistema con la confianza de que el sistema se aplicará consistentemente. Será el autor/donante quien decida si quiere distribuir software mediante cualquier otro sistema y una licencia no puede imponer esa elección.

Este apartado pretende dejar completamente claro lo que se cree que es una consecuencia del resto de esta Licencia.

8. Si la distribución y/o uso de el Programa está restringida en ciertos países, bien por patentes o por interfaces bajo copyright, el tenedor del copyright que coloca este Programa bajo esta Licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera. En ese caso, esta Licencia incorporará la limitación como si estuviese escrita en el cuerpo de esta Licencia.

9. La Fundación de Software Libre puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo. Dichas nuevas versiones serán similares en espíritu a la presente versión, pero pueden ser diferentes en detalles para considerar nuevos problemas o situaciones.

Cada versión recibe un número de versión que la distingue de otras. Si el Programa especifica un número de versión de esta Licencia que se refiere a ella y a "cualquier versión posterior", tiene la opción de seguir los términos y condiciones, bien de esa versión, bien de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, puede escoger cualquier versión publicada por la Free Software Foundation.

10. Si quiere incorporar partes del Programa en otros programas libres cuyas condiciones de distribución son diferentes, escriba al autor para pedirle permiso. Si el software tiene copyright de la

Fundación de Software Libre: algunas veces hacemos excepciones en estos casos. Nuestra decisión estará guiada por el doble objetivo de preservar la libertad de todos los derivados de nuestro software libre y promover el que se comparta y reutilice el software en general.

### **AUSENCIA DE GARANTÍA**

11. Debido a que el programa se licencia libre de cargas, no se ofrece ninguna garantía sobre el programa, en toda la extensión permitida por la legislación aplicable: excepto cuando se indique lo contrario por escrito, los poseedores del copyright y/u otras partes proporcionan el programa "tal cual", sin garantía de ninguna clase, bien expresa o implícita, con inclusión, pero sin limitación a las garantías mercantiles implícitas o a la conveniencia para un propósito particular. cualquier riesgo referente a la calidad y prestaciones del programa es asumido por usted.

Si se probase que el programa es defectuoso, usted asume el coste de cualquier servicio, reparación o corrección.

12. En ningún caso, salvo que lo requiera la legislación aplicable o haya sido acordado por escrito, ni el poseedor del copyright ni ninguna otra parte que modifique y/o redistribuya el programa según se permite en esta licencia será responsable ante usted por daños, incluyendo cualesquiera daños y perjuicios generales, especiales, incidentales o resultantes producidos por el uso o el uso inadecuado del programa (con inclusión, pero sin limitación a la pérdida de datos o a la generación incorrecta de datos o a pérdidas sufridas por usted o por terceras partes o a un fallo del programa al funcionar en combinación con cualquier otro programa), incluso si dicho titular u otra parte ha sido advertido de la posibilidad de dichos daños.

### **FIN DE LOS TÉRMINOS Y CONDICIONES**

### 1.1.2 ¿CÓMO APLICAR ESTOS TÉRMINOS A LOS NUEVOS PROGRAMAS? <sup>5</sup>

Si se desarrolla un nuevo Programa, y quiere que sea de la mayor utilidad posible para el público en general, la mejor forma de conseguirlo es convirtiéndolo en software libre que cualquiera pueda redistribuir y cambiar bajo estos términos.

Para hacerlo, debemos añadir los siguientes anuncios al programa. Lo más seguro es añadirlos al principio de cada fichero fuente para transmitir lo más efectivamente posible la ausencia de garantía. Además cada fichero debería tener al menos la línea de "copyright" y un indicador a dónde puede encontrarse el anuncio completo.

<una línea para indicar el nombre del programa y una rápida idea de qué hace.>

Copyright (C) <nombre del autor>

Este programa es software libre. Puede redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General de GNU según es publicada por la Free Software Foundation, bien de la versión 2 de dicha Licencia o bien (según su elección) de cualquier versión posterior.

Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGÚN TIPO DE GARANTÍA, incluso sin la garantía MERCANTIL implícita o sin garantizar la CONVENIENCIA PARA UN PROPÓSITO PARTICULAR.

Se debe añadir también información sobre cómo contactar al autor mediante correo electrónico y postal.

## 1.2 VENTAJAS DE GPL EN EL AMBIENTE CORPORATIVO

El entorno GPL (Linux en general), está lleno de software técnico, de comunicaciones, utilidades, etc. Sin embargo, en cuanto a software de **gestión empresarial**, las cosas no son tan variadas. Hay poco, y menos en fuente abierta.

---

<sup>5</sup> <http://www.gnu.org>

Algunos factores caracterizan el software de gestión empresarial: <sup>6</sup>

1. Ha de estar adaptado al **idioma** del lugar donde ha de aplicarse.
2. Ha de estar adaptado a la **legislación** del lugar donde deba aplicarse y está cambia, como poco, de país a país.
3. Requiere **conocimiento y experiencia en el sector** empresarial donde debe aplicarse, algo que tienen los gestores de empresa, pero no los programadores general.
4. Las empresas lo convierten fácilmente en **dinero**, y por tanto están dispuestas a **pagar** por él.

Esto trae algunas consecuencias:

- El software de gestión suele ser muy específico de cada país.
- Es un campo poco atractivo para las grandes multinacionales.
- Los frecuentes cambios de legislación y de necesidades del usuario corporativo, hacen que este software tenga una gran demanda de soporte y servicios.

Antes de seguir, hay que indicar que esta "especificidad" con respecto a cada país, no implica una exclusión de diferentes países de habla española; todo lo contrario, tratemos de aprovechar todo lo que pueda haber en común y que cada uno en su país. Particularice lo que necesite.

Por otra parte, cuando un cliente quiere un programa para "llevar su empresa", podemos encontrarnos con:

1. Hay un software estándar que se adapta a la perfección a la empresa del cliente. Dado que cada empresa tiene su propio estilo (en el que suele residir la clave de su supervivencia), esta situación está lejos de ser la más frecuente.

---

<sup>6</sup> <http://www.gpl.com>

2. Hay algo que se aproxima, pero no es lo ideal; habría que modificarlo, o el usuario habrá de adaptar su empresa al software existente (actitud peligrosa).
3. No hay nada que se parezca, hay que hacerlo a medida.

Los desarrolladores de grandes paquetes estándar, se encontraron hace tiempo con la problemática de que dos empresas dadas, incluso del mismo sector, podían tener formas muy diferentes de llevar el negocio, necesitando por tanto programas muy diferentes. Esto limitaba seriamente la utilidad de eso que se llama "programas estándar".

La solución a esto fué hacer programas muy configurables, o dicho de alguna manera, "programas programables". Esto hizo que dichos paquetes se volvieran enormes, y con frecuencia, muy complejos de configurar.

Esta curiosa evolución del software de altas prestaciones, no deja de ser una versión degenerada del concepto de fuente abierta; es decir, el software es modificable, pero los fuentes siguen sin estar disponibles.

Por tanto, cabe concluir que un software de gestión con fuente abierta, tenga una excelente aceptación empresarial.

Por otra parte, las empresas que ya tienen paquetes cerrados, no van a liberar su fuentes ni a participar en nada que amenace su mercado cautivo. A las multinacionales, como la experiencia demuestra, el tema no les interesa (salvo que sea SAP o similar).

### 1.2.1 USUARIOS DE ESTE TIPO DE SOFTWARE

En primer lugar, a las empresas usuarias, con necesidades muy variables y presupuestos limitados.

El software bajo Linux en fuente abierta, permite a las empresas:

1. Ahorrar el costo de las licencias del sistema operativo, sin incurrir en "pirateo"
2. Disponer de un software que puede ser atendido por muchos posibles proveedores de asistencia, no uno sólo.
3. Pagar exactamente por el servicio que necesitan. Ni más ni menos, ya que es algo que pueden contratar aparte, sobre algo que de base, es gratis. Esto significa que si no necesitan mucho soporte, ahorrarán dinero, y si lo necesitan, podrán gastar su dinero en pagar dicho soporte, no la "tasa" que supone el costo de licencia.

En segundo lugar, interesa a todas las pequeñas empresas de servicios informáticos. Un desarrollo en fuente abierta, a través de Internet, con la colaboración de otros desarrolladores, va a permitir:

1. Abordar proyectos imposibles de otra forma, por su envergadura o especialización.
2. Tener programas más depurados, al ser vistos, probados y utilizados por mucha más gente.
3. Reducir costos de desarrollo bárbaramente.
4. Eliminar dependencias de programadores individuales concretos, al haber muchos programadores en Internet que conocerán el mismo código.

### 1.3 FUTURO DE LAS PLATAFORMAS GPL <sup>7</sup>

#### La GPL: copyleft para tod@s

Utilizando un brillante juego de palabras, tan del gusto de los Hackers, Stallman inventa el concepto de *copyleft*, con el propósito político de garantizar la libre circulación de los saberes contenidos en el

---

<sup>7,8</sup> <http://www.gnu.org>

software y la posibilidad de que todos contribuyan a su mejora. El copyleft se sirve de las leyes internacionales del copyright para darles la vuelta (*all rights reversed*: "todos los derechos del revés") pues protege el uso en lugar de la propiedad.<sup>8</sup>

El autor se reserva los derechos para que su obra pueda ser utilizada por cualquiera con la única condición de que nadie recorte o elimine esos derechos de libre uso: en el momento en que alguien suprime o añade nuevas condiciones que limiten en algo su disponibilidad (por ejemplo, distribuyendo código binario modificado sin posibilidad de acceder a las fuentes modificadas) estaría vulnerando la licencia y perdería el derecho a servirse de ese software. Obligando a transferir esos derechos a cualquiera que copie ese software, lo modifique o no, se beneficia quien está de acuerdo con mantener su futuro trabajo con copyleft, mientras que quien quiera desarrollar software propietario no podrá utilizar código libre y deberá empezar desde cero.

La GPL o Licencia Pública General es la plasmación jurídica del concepto copyleft. Con el tiempo, la GPL se ha convertido en el cimiento del software libre, su baluarte legal, y para muchos constituye un extraordinario ejercicio de *ingeniería jurídica*: con la GPL se asegura que trabajos fruto de la cooperación y de la inteligencia colectiva no dejen nunca de ser bienes públicos libremente disponibles y que cualquier desarrollo derivado de ellos se convierta como por ensalmo en público y libre.

La GPL se comporta de un modo "virico" y, como un rey midas del software, convierte en libre todo lo que toca, es decir, todo lo que se deriva de ella.

### 1.3.1 OPEN SOURCE SOFTWARE OSS <sup>9</sup>

El Software de Fuente Abierta es un nuevo fenómeno en la industria de software basado en la premisa de que éste no es tanto un bien de consumo como un tipo de información.

---

<sup>9,10</sup> <http://www.gnu.org>

Sobre esta base se articula toda una filosofía concerniente a la producción de código y a su distribución. Una de sus más importantes conclusiones es que el código debe ser accesible y modificable, lo que conlleva una nueva visión de la industria informática.

Este tipo de software se caracteriza por su dinamismo en la producción y depuración de sus componentes, su creciente diversidad, su flexibilidad para adaptarse a las distintas necesidades y su disponibilidad a bajo o nulo coste.

### 1.3.2 DEFINICIÓN DE FREEWARE <sup>10</sup>

El término "Freeware" hace referencia, en principio, al software gratuito, pero, como es corriente en la popularización de un tecnicismo, se ha vuelto excesivamente vago.

Normalmente se denota con "Freeware" a todo aquel software cuyos términos de distribución son más amplios que los del software propietario. Es por esto que los puristas del OSS evitan este vocablo utilizando circunlocuciones como "software de libre distribución".

### 1.3.3 TIPOS DE FREEWARE <sup>11</sup>

Podríamos distinguir las siguientes categorías de "Freeware":

#### 1. Shareware

Este tipo de software se caracteriza por ser de distribución gratuita, pero exigir que, de ser usado regularmente, se pague una licencia (en general de bajo coste). Nótese que normalmente las fuentes no están incluidas.

---

<sup>11</sup> <http://www.gnu.org>

## 2. Software para uso no profesional

Esta categoría engloba a todos aquellos productos que son de uso gratuito para entidades sin afán de lucro, pero exigen el pago de una licencia corporativa.

## 3. Librerías y archivos ejecutables gratuitos

Es el software distribuido de manera gratuita, pero que no incluye el código fuente. Es relativamente común confundirlo con OSS, aún cuando son cosas bien diferentes. El OSS no es necesariamente gratis, pero siempre debe hacer disponibles sus fuentes.

## 4. Software de Fuente Abierta no revisado y comercializable

Constituye la primera división del OSS. Este tipo de distribución incluye las fuentes, que son modificables, pero no trata la revisión de las mismas por terceras personas. Además, trabajos realizados sobre el código original pueden ser comercializados bajo otro tipo de licencias. Esta licencia se utilizó en la creación del BSD-UNIX, con un núcleo cerrado de programadores que se encargaban, en solitario, del desarrollo del proyecto.

## 5. Software de Fuente Abierta revisado y comercializable

Éste es el segundo tipo de OSS. En este caso, las fuentes modificadas son revisadas y susceptibles de ser incorporadas al proyecto, lo que da al proceso de desarrollo mayor dinamismo. Nótese que sigue siendo posible comercializar derivados del producto bajo otras licencias. Este tipo de distribución es la que se utiliza en el popular servidor de Web Apache.

## 6. Software de Fuente Abierta revisado y no comercializable

Es el sistema conocido como "Copyleft". Extiende el anterior tipo de licencia obligando a los trabajos derivados a seguir siendo OSS. La [licencia GPL](#) es prototipo de esta categoría. Toda la producción de GNU, desde el kernel de Linux hasta Gnome está bajo ella.

### 1.3.4 SOFTWARE PROPIETARIO <sup>12</sup>

#### Definición de "Software Propietario"

Entendemos por software propietario aquella concepción de la industria informática en la que los estándares y el diseño relacionados con un proyecto están en manos de una o varias compañías de tal modo que quedan fuera del acceso del público general.

### 1.3.5 OSS Y LA COMPETENCIA EN EL SECTOR DE LA INFORMÁTICA <sup>13</sup>

Es evidente que el OSS choca frontalmente con la propiedad del software, en realidad, surge como reacción contra ella. Así, compañías que lideran el sector del software propietario ven en el OSS una importante amenaza a sus beneficios.

Esta inevitable competencia entre dos formas de entender la producción de software es tal vez la mayor que jamás se ha producido en el sector. Hay que tener en cuenta la fuerte predominancia de una única empresa en el campo de las aplicaciones propietarias, y las estrategias de mercado de otras compañías, que ven en el potencial del OSS una forma de debilitarla.

El OSS está siendo utilizado por muchos como plataforma de lanzamiento de sus productos, consiguiendo una difusión mucho mayor que la que hubieran obtenido mediante las vías tradicionales.

Posteriormente, utilizan la experiencia y cuota de mercado adquiridas para desarrollar una aplicación propietaria de mayores prestaciones. Este es posiblemente el caso del proyecto JINI de Sun, del Mozilla de Netscape y de varios otros.

Es por esto que las nuevas iniciativas de OSS se han vuelto extremadamente cuidadosas a la hora de redactar los límites de sus licencias.

---

<sup>12,13</sup> <http://www.gnu.org>

### 1.3.6 EL OSS COMO REACCIÓN ANTE EL CORPORATIVISMO <sup>14</sup>

Los grupos OSS basan su filosofía en los siguientes principios:

- El corporativismo que gobierna hoy día la producción de software es desde un punto de vista social, negativo, ya que arrebató a los usuarios la libertad de compartir y acceder a la información. Las consecuencias de esto se traducen en una menor capacidad de elección y en un peor ambiente social, ya que la ley obliga a los individuos a no compartir la información, lo que va en contra de su naturaleza. Todo esto va en perjuicio de la sociedad.
- Desde el punto de vista productivo, la no disponibilidad del código fuente lleva a menudo a la repetición del trabajo, lo que no produce más que frustración a los programadores y gastos inútiles a los empresarios. La accesibilidad de las fuentes podría hacer que este esfuerzo fuese invertido en la solución de nuevas necesidades, y ayudaría a un más rápido avance del conocimiento y de la civilización.
- Los programadores no desaparecerán si desaparece el software propietario, ya que existen formas de financiación alternativas a la restricción de la copia en la distribución de los productos. Se citan a menudo como posibles iniciativas la distribución comercial de colecciones de software, dedicando parte de los beneficios al mantenimiento del desarrollo del OSS, la imposición de impuestos sobre la producción de hardware o la financiación directa y voluntaria de las casas de hardware de productos de software abierto.
- Por otro lado, se juzga que la programación, como ha sucedido con otros campos de la ciencia y de las artes, es suficientemente atractiva como para hacer que se siga trabajando en ella aunque los beneficios no sean exageradamente altos, al igual que se desarrollaron nuevas teorías matemáticas, físicas, o estilos artísticos en épocas de recesión económica.
- Se considera que el software propietario va en contra de la filosofía de la libre competencia. Las empresas en el sector estarían abusando de un sistema legal diseñado en un contexto diferente

---

<sup>14</sup> <http://www.gnu.org>

del actual para obtener beneficios inmorales a costa de la sociedad. Este es el caso de la patente de segmentos de software o conceptos relacionados con él, que vicia enormemente, si no imposibilita, el libre mercado dentro del sector. Se postula, que el software abierto acabaría de raíz con estos problemas, al devolver al usuario una capacidad de elección que ahora se encuentra mutilada, y al programador la posibilidad de competir en igualdad de condiciones.

### 1.3.7 OTRAS CARACTERÍSTICAS DE LA GPL <sup>15</sup>

Es interesante destacar que la licencia GPL no implica restricciones a la distribución comercial del software, sino que únicamente asegura la libertad en el uso, modificación y redistribución posteriores. Esto permite que surjan compañías que se dediquen a la distribución comercial del software e incluso ofrezcan soporte técnico a particulares y/o empresas.

Esta es, en efecto, la llave que ha permitido la entrada en el OSS de compañías como **Red Hat Software**, **Debian**, **Slackware** y otras, que ofrecen, a un costo relativamente bajo, sistemas operativos completos, con interfaces de instalación profesionales y provisto de un conjunto amplio de herramientas de sistema. También ha permitido la oferta de soporte técnico por gran cantidad de empresas, que ayudan a salvar el escollo en la distribución de software gratuito que constituye la falta de garantía.

### 1.3.8 EL PROBLEMA DE LAS PATENTES. PORQUÉ LAS PATENTES CONSTITUYEN UNA AMENAZA AL OSS <sup>16</sup>

Ha sido práctica común por parte de las empresas de software propietario la patente de segmentos de código e incluso de conceptos generales relacionados con la ingeniería informática. Existen numerosísimas soluciones de software que son propiedad de compañías concretas.

---

<sup>15</sup> <http://www.gpl.com>

<sup>16</sup> <http://www.gnu.org>

En algunos casos estos procedimientos son enormemente ambiguos y en el marco actual de la programación, es difícil escribir porciones de código de un tamaño considerable sin interferir con algún derecho intelectual.

Para empeorar aún más la situación, los organismos encargados de estas patentes han permitido registrar ideas que, bajo estándares de ingeniería de software son relativamente evidentes y constituyen caminos únicos cuando es necesario tener en cuenta la eficiencia del código.

Así, toda la comunidad de desarrolladores excepto algunas grandes compañías con importantes recursos económicos y numerosas patentes, se ve enfrentada a demandas legales u obligada a renunciar a soluciones eficientes que hubiese sido capaz de volver a desarrollar por si misma.

Dado el inmenso volumen de patentes en vigor, los nuevos programas se ven desbordados por una carga económica de "Royalties" que, en muchos casos, hace inviable su desarrollo.

### **1.3.9 COMERCIALIZACIÓN DEL OSS <sup>17</sup>**

Debido a su propia naturaleza, resulta evidente que el software abierto tiene una distribución comercial cuando menos peculiar. La no limitación de la difusión del código fuente y su disponibilidad en la web, hacen que componentes concretos no sean susceptibles de ser vendidos por separado.

#### **Comercio directo con el OSS**

La comunidad OSS ha hecho un esfuerzo importante para desvincular la idea de software de fuente abierta de la idea de software por el que no se puede cobrar. En efecto, el OSS lleva implícita la connotación de su disponibilidad gratuita, pero esto no impide que ciertas formas de distribución del mismo no puedan producir un negocio rentable.

---

<sup>17</sup> <http://www.gnu.org>

Hasta el momento la actividad que ha demostrado ser más comercial, relacionada con el comercio directo con el OSS, ha sido la venta de las denominadas "distribuciones", que se basan en la comodidad que supone para el usuario el tener reunidos todos los componentes necesarios para poner en marcha su sistema informático.

Además, las distribuciones de OSS proporcionan una interfaz atractiva entre la muy dinámica comunidad de desarrolladores y un público cliente que no está dispuesto a actualizar su sistema al ritmo al que evoluciona realmente el software. Así, si bien las nuevas versiones del kernel de Linux aparecen muy a menudo, las distribuciones no se actualizan hasta que ha pasado un plazo mayor y se ha desarrollado un salto tecnológico sensible.

Como ejemplo de la importancia en el mercado de esta forma de comercialización, cabe reseñar que todas las grandes distribuciones GNU/Linux, están registrando niveles de ventas situados a la cabeza del sector, sobrepasando a multitud de aplicaciones comerciales.

En el 3º número de "Sólo Linux" aparece un artículo que registra un hito de ventas de SuSE y RedHat, al superar entre los dos en el parcial del mes de Octubre las ventas de Windows 98 en unos grandes almacenes parisinos.

### 1.3.10 SERVICIOS RELACIONADOS CON EL OSS <sup>18</sup>

Otra fuente de negocio en este campo, quizá de mayor interés, consiste en la oferta de servicios de mantenimiento y soporte técnico. Dada la relativamente baja confianza que puede inspirar un software en continuo desarrollo y distribuido sin garantía de ningún tipo, es comprensible que una empresa requiera los servicios de un operador especializado capaz de solucionar los problemas que ocasionalmente pudieran aparecer. En cierta forma, puede ser un negocio más rentable el basar el servicio proporcionado en OSS, que hacerlo sobre software propietario, ya que el usuario no tiene que

---

<sup>18</sup> <http://www.gnu.org>

pagar elevados precios de licencia al fabricante, y puede con mayor facilidad permitirse los honorarios de una empresa de estas características.

#### 1.4 TECNOLOGÍA INTRANET

Anteriormente en el mundo de los negocios se usaban herramientas de comunicación lentas y métodos ineficientes. Si querías hacer llegar un mensaje al personal de la empresa, tenías que distribuir un comunicado, repartir la información en los tabloncillos de anuncios o avisar a la gente por los pasillos o en la máquina de café.

Entonces llegaron las computadoras, el email e Internet. En un "clic" podías enviar mensajes y documentos a un grupo específico de personas o a cada una de ellas por separado. También se disponía la posibilidad de consultar noticias online e información al instante relacionadas con el sector o competidores.

Pero con el email y las webs llegaron nuevos retos. Importantes mensajes o documentos que se enviaban, se perdían o se borraban por accidente. El resultado, mucho de tiempo perdido re-enviando el mensaje o buscando mensajes ya archivados. O incluso peor, la gente con mucha frecuencia usaba versiones no actualizadas de documentos, lo que implicaba a no tomar la mejor decisión.

Si bien el email y las páginas Web son herramientas muy eficientes para obtener información publicada por fuentes externas, no lo son tanto cuando tratamos de buscar un acceso rápido a la información publicada de forma interna por nuestra compañía. Encontrar el último comunicado de prensa, aviso, lista de precios, periodos de vacaciones, documentación sobre proyectos concretos, esquemas y otra información crítica puede ser una tarea frustrante, repasando viejos emails, llamando a una y otra persona, tratando de dar con lo que necesitas.

Existe una solución para superar estos problemas, la intranet. Herramienta implantada habitualmente en grandes corporaciones, que ahora la nueva tecnología de Planet Intra pone a la disposición de cualquier empresa. Una solución de fácil instalación y mínimos requisitos de mantenimiento, que permite a las Pymes de desplegar nuevas estrategias de comunicación técnicas de trabajo interna.



#### 1.4.1 ¿QUÉ ES UNA INTRANET?



Una Intranet es un sistema de información dentro de la empresa basado en *tecnología web*, i.e., una red de servicios internos para la empresa basados en estándares Internet, por lo que un usuario (delegaciones, clientes, proveedores,

personal, red de distribución, red de transporte, etc) puede acceder a diferentes aplicaciones de la empresa.<sup>19</sup>

Con un servidor de intranet, cientos o miles de empleados pueden obtener información actualizada y confiable. Utilizando un manejador de bases de datos, tenemos información organizada y estructurada. Por medio del acceso a los sistemas de la empresa por medio de compuertas e interfaz podemos llevar a cabo una mayor disponibilidad de información requerida en nuestros días. Tomando las ventajas de los sistemas de programación en el cliente o el servidor contamos con aplicaciones extensibles y de rápida instalación basando nuestro desarrollo en torno a nuestro Explorador como la interfase universal.

Todo se traduce en una red empresarial de máquinas conectadas a un servidor o servidores Web. Los distintos empleados podrán acceder a cualquier tipo de información publicado en un "site" de esta web, de la misma manera que lo haría si conectase con la red global (esto es Internet).

También es posible que el trabajador acceda a sites tanto de la red empresarial (la Intranet) como la mundial (el WWW, FTP, etc.). En tal caso, la red interna de la compañía estará protegida de cualquier "visitante non grato" por medio de los cortafuegos o "firewalls",<sup>20</sup>

#### 1.4.2 INTERNET / INTRANET ¿CUÁL ES LA DIFERENCIA?

Básicamente, la localización de la información y quién accede a ella.

*Internet* es público, global, abierto a cualquiera que tenga una conexión a Internet.

Las *Intranets* están restringidas a aquellas personas que están conectadas a la red privada de la empresa.

---

<sup>19</sup> <http://www.intranet.com>

<sup>20</sup> <http://www.lavariante.com>

Además de eso, ambas funcionan esencialmente de la misma manera, con la misma tecnología TCP/IP para regular el tráfico de datos. Del mismo modo, desde el punto de vista de las aplicaciones (e-mail, newsgroups y transferencia de ficheros FTP, además del Web), no existe ninguna diferencia entre Internet e Intranet.

### 1.4.3 COMO FUNCIONA UNA INTRANET <sup>21</sup>

En la Intranet de una compañía, el acceso a la información es muy similar al acceso a Internet. Ello supone que, si conocemos el proceso para Internet, será casi automático empezar a utilizar una Intranet (con el ahorro en formación que eso supone para la empresa).

Se puede describir el proceso de la siguiente manera:

- Todas las máquinas cliente (PCs locales conectados a una máquina Servidor de Web), utilizan un software especial, un browser (por ejemplo, Netscape Navigator o Internet Explorer), además de software especial para el correo electrónico, FTP y demás servicios.
  
- El usuario hace click en un hipervínculo, que envía una petición de información al servidor. El hipervínculo contiene el nombre del servidor donde está localizada la información, y el nombre del fichero que ha de ser cargado en el browser.
  
- El servidor lee la petición, vuelca el fichero, y lo manda al cliente. El cliente recibe el fichero y lo deposita en la ventana del browser (siempre que dicho fichero sea compatible con el web, por ejemplo, con formato HTML y otros). Puede ocurrir también que el fichero se deposite en discos o en aplicaciones especiales.

### 1.4.4 ¿QUÉ NECESITAMOS PARA TENER UNA INTRANET? <sup>22</sup>

---

<sup>21</sup> <http://keinfos.com/web/intranet/>

<sup>22</sup> <http://infoperu.com/espanol/intranet.html>

Las Intranets son posibles gracias a la concepción de la informática distribuida, que surgió en el mundo de la empresa en la década de los ochenta. Uno de los avances en este campo, la llegada de lo que se denominan Redes de Área Extensa, o WAN, ha hecho de las Intranets un fenómeno posible en la realidad actual. Cuando hablamos de informática distribuida nos referimos a las redes de computadoras (las más conocidas son posiblemente las redes LAN), que han sustituido en parte a las tradicionales mainframes corporativas.

A la hora de construir cualquier red y, en especial, una Intranet, necesitaremos unos elementos básicos:

En primer lugar, es necesario disponer de un hardware o soporte físico adecuado para la Intranet, que incluye:

- PCs o Estaciones de Trabajo que actúan como *Servidores de Web*, bajo el modelo Cliente-Servidor.
- PCs de los trabajadores, que actúan como Clientes, y sus correspondientes periféricos.
- Un sistema de cableado que interconecta el Servidor o Servidores con los equipos Cliente (cable coaxial, de par trenzado, o fibra óptica)
- Elementos de hardware que configuran el concepto tradicional de red: tarjetas de conexión o NIC (*Network Interface Card*), transceptores, repetidores, concentradores o *hubs*, etc...
- Máquinas que actúan como firewalls, y su correspondiente software.
- En segundo lugar, necesitaremos una serie de elementos de software que hagan posible configurar la red como una Intranet. Destacan:

- Un sistema operativo de red, que soporta el intercambio de información y, que, como tal, reside tanto en clientes como en servidores. Hoy en día, existen varios sistemas operativos disponibles en el mercado: Unix, Linux, Windows NT, Novell Netware, y otros.
- Aplicaciones de red, que en este caso, se refieren a la utilización de browsers, residentes en los equipos servidor y clientes, así como de programas específicos de *correo electrónico*, FTP, etc.
- Un sistema de gestión de red, que permite el control de prestaciones, problemas, seguridad o configuración.
- Protocolos de comunicación Web estándares

El desarrollo de las Intranets y de Internet está propiciando actualmente el desarrollo de nuevas tecnologías hardware y de múltiples programas software que agilizan la comunicación corporativa (un ejemplo: el Catalog Server de Novell, que permite la creación de índices de Web sites como los buscadores actuales de Internet Yahoo, Lycos, y otros).

#### 1.4.5 ¿QUE BENEFICIOS APORTA UN INTRANET A LA EMPRESA? <sup>23</sup>

Antes de plantearse el desarrollo de un intranet, tanto si se trata de una gran empresa como si es un pequeño grupo de trabajo, debe tenerse claro un concepto fundamental, nunca debe tratarse como una plataforma de trabajo sino como una herramienta de trabajo.

Una herramienta que aportará múltiples beneficios:

- **Ahorro:** adiós a documentos, formularios, manuales, tarifas, notas internas, bases de datos y demás documentación que obliguen al uso de papel y a la utilización de impresoras para la distribución de la información diaria de la empresa. Esta infraestructura necesaria puede dejar

---

<sup>23</sup> <http://www.intranet.com>

paso a páginas de texto HTML, sonidos, e incluso imágenes digitalizadas que tan sólo requieren un navegador para su consulta.

- **Calidad:** Internet es la cantidad mientras que un intranet es la calidad. La Red está cargada de información importante, pero el tiempo que se debe emplear para su localización es a menudo demasiado para ofrecer soluciones rápidas a las dificultades diarias.
- En la intranet el usuario encuentra todo lo que necesita ya que la información se encuentra mucho más organizada y seleccionada.
- **Comunicación:** el sistema de mensajería implementado puede ofrecer comunicaciones internas y externas, en tiempo real y diferido, y completamente integradas.
- **Control:** los datos importantes de la organización no estarán tan sólo en la mesa de un directivo o en la de un mando intermedio. Se sabrá en cada momento cómo se encuentra un proyecto ya que la información en un intranet se modifica y consulta en tiempo real.
- **Colaboración:** permiten aprovechar la experiencia intelectual individual de todos los empleados y tenerla disponible para su utilización global. Los usuarios pueden crear y manejar sus propios contenidos.
- **Efectividad:** Permite que no se dejen asuntos pendientes y que se acaben las excusas de falta de información para llevar a cabo un proyecto cualquiera. Además cada departamento genera y mantiene su propia documentación así como los datos que le pertenecen.
- **Eficiencia:** un intranet elimina en porcentajes muy altos los métodos de trabajo tradicionales y en ocasiones poco eficientes como el teléfono, el fax o las pequeñas reuniones de pasillo..

- **Facilidad:** con el simple requerimiento de saber manejar un navegador, cualquier usuario corporativo podrá trabajar con un intranet. Con algo más de conocimientos, no excesivos, podrá dotarla de toda su información y compartirla con el resto de los empleados de la organización.
- **Flexibilidad:** el trazado de puesta en marcha de un intranet es algo que puede comenzar con un nivel relativamente simple e incrementar las prestaciones según las necesidades y las posibilidades de la empresa. El adecuado mantenimiento y la actualización posterior juegan un papel fundamental en el buen desarrollo del proyecto que se verá culminado con la adaptación al nuevo medio de todos los usuarios que componen la compañía.
- **Rendimiento:** todo el material que se puede tener de manera impresa en un catálogo, manual o libro, se puede implementar sin mucho esfuerzo en un intranet.
- **Participación:** un miembro cualquiera de la organización en la que se monta un intranet aporta la información que ha conseguido y la pone a disposición de todos los demás, que a su vez la pueden mejorar o implementar sus propios datos.

#### 1.4.6 ¿QUÉ SERVICIOS AÑADE UNA INTRANET A SU EMPRESA? <sup>24</sup>

Considerando que las intranets se desarrollan por y para el usuario y que las necesidades de ésta varían, hay diferentes servicios básicas que, como componentes muy elementales, todo intranet debe implementar.

- **Mensajería y correo electrónico:** comunicaciones internas y externas, en tiempo real y diferido, integradas. Todas las empresas precisan de un canal de información que puede sustituir al teléfono y que posibilite la comunicación y, sobre todo, la colaboración.

---

<sup>24,25</sup> <http://keinfos.com/web/intranet/>

- **Contenido e información compartida:** la información de la empresa puede ser referenciada en página HTML que incluyan textos, hojas de cálculo, etc. Hiperenlaces, multimedia y objetos incrustados, permiten integrar y personalizar contenidos en línea enriquecidos e interactivos que pueden ser consultados por toda la organización.
- **Acceso a aplicaciones:** debe permitir, sencillamente y bajo un solo interfaz, el acceso a bases de datos, información y aplicaciones ya existentes. Puede surtir de versiones actualizadas de software accesibles por parte de varios usuarios.
- **Administración:** debe proporcionar una interfaz de administración común, sencilla y basada en HTML que permita gestionar con seguridad los servidores y recursos desde cualquier lugar de la intranet.

#### 1.4.7 SERVICIOS ADICIONALES QUE APORTA UNA INTRANET A LA EMPRESA <sup>25</sup>

##### Beneficios de Comunicación

- Es posible recuperar cualquier tipo de documento en línea.
- Correo electrónico.
- Servicio de chat, conferencia de voz, videoconferencia entre usuarios.
- Unifica las operaciones de una empresa, incrementando de forma fácil y frecuente la comunicación interdepartamental con una mayor calidad.
- Contiene indicaciones para los empleados de cómo usar las tecnologías requeridas.

- Es posible acceder a diferentes bases de datos las cuales tienen información necesaria para el desarrollo de cualquier actividad, incrementando la eficiencia de los procesos y tiempos de respuesta.
- Aplicaciones multiplataforma. Las aplicaciones Intranet se entregan como páginas Web (HTML) y son multiplataforma por defecto.

### **Beneficios de Seguridad**

- Protege la información frente a accesos no autorizados.
- Puesto que una Intranet está basada en la tecnología de Internet, puede tener miles de usuarios ubicados en muchos lugares diferentes y seguir siendo privada.
- Beneficios Operacionales y Técnicos
- Consta de una arquitectura cliente-servidor.
- Su uso no requiere de un entrenamiento y solamente es cosa de colocar el mouse y dar un click a un tópico que nos interese.
- Soporta cualquier tipo de hipervínculo ya sea gráfico o textual (protocolo de transferencia de hipertextos)
- Trabaja con eficiencia igualmente en cualquier plataforma (UNIX, MAC, PC, etc.). Integrado en cualquier ambiente computacional operativo (WINDOWS NT, MAC, WINDOWS 95, 98).
- Trabaja en un soporte de ambiente de red comercial TCP/IP. Aplicaciones de banda estrecha. Dado que la mayor parte de procesamientos se hacen en el servidor y sólo se ven las páginas HTML hacia el cliente (algunas veces con componentes pequeños de software incrustados e información de guiones), las aplicaciones Web son las más indicadas para las conexiones de banda estrecha.

- Esto es conveniente para organizaciones que tienen muchos trabajadores móviles, tales como agentes comerciales que necesitan acceso a la información corporativa desde el campo o trabajadores que necesitan acceso desde sus casas.
- Webnet ofrece los servicios de: Mantenimiento, actualización (en caso de contar con una) y creación de Intranet (si no cuenta con ella su empresa) .

#### 1.4.8 ¿QUÉ PROCEDIMIENTOS NOS LLEVAN A LA REALIZACIÓN ÓPTIMA DE UNA INTRANET? <sup>26</sup>

- 1) Definir claramente cuáles son los objetivos y propósitos de su empresa.
- 2) Familiarización con los actividades que se llevan diariamente, quién las realiza y de qué manera se comunican.
- 3) Después de un análisis de los puntos anteriores se realiza un plan estratégico que contemple:

##### Planificación

- Cuál es la estructura que contendrá el Sitio Web.
- Cuál es la finalidad de la instalación de la Intranet.
- Quiénes estarán comunicados y con quién.
- Cuál será la información que podrá ser accesada por medio de la Intranet.

##### Diseño

- Realizar estratégicamente un diagrama de navegación de la red, incluyendo los niveles jerárquicos.

---

<sup>26</sup> <http://www.srt.com-ar/intranet.html>

- Diseño de cada una de las páginas y sus vinculaciones con otras.

#### 1.4.9 VENTAJAS GENÉRICAS DE LAS INTRANETS <sup>27</sup>

Las Intranets, como redes que utilizan la tecnología Internet, aportan una serie de ventajas básicas a la gestión empresarial, que pueden resumirse en:

- Sistema Universal
- Costo Aceptable
- Adaptación a las Necesidades Específicas
- Desencapsulamiento y Distribución de la Información
- Aumento de la Calidad de la Comunicación Corporativa
- Rápida formación de las personas en su uso
- Seguridad
- Internet

#### 1.4.10 APLICACIONES DE INTRANET <sup>28</sup>

Basándonos en las áreas en las que Intranet se puede desarrollar realizando aportaciones de gran relevancia, podemos diferenciar los siguientes rangos de aplicaciones:

- Aplicaciones genéricas.
- Aplicaciones a Recursos Humanos.
- Aplicaciones a Formación.

---

<sup>27,28</sup>

<http://www.intranet.com>

#### 1.4.11 ¿CÓMO MANTENER LA INTRANET SEGURA? <sup>29</sup>

A esta alturas seguramente estamos de acuerdo en que las intranets aportan un valor sin precedentes a la distribución de la información, la automatización de los grupos de trabajo y el acceso a la información corporativa. Pero no podían ser todo ventajas.

La facilidad de acceso a la información que permiten la intranets supone un aumento del riesgo de robo, pérdida o corrupción de esa información que ponemos a disposición en ella.

Y eso es precisamente la que desde la intranet de una empresa se debe evitar, para que la información no esté a disposición de quien no deba conocerla. Para ello existen mecanismos de seguridad que no se deben olvidar.

- Si la intranet de una empresa carece de los controles de acceso adecuados, supone una invitación al borrado accidental o a la modificación de documentos. Por lo cual **la política de seguridad es algo que nunca debe dejarse de lado.**
  
- Es necesaria **toda la seguridad de que se pueda disponer.** Debemos decidir quién tiene acceso a cada documento, cuándo y desde dónde.
  
- Primero debe considerarse **la seguridad física de cada computadora** (claves, cerrojos, cerraduras cifradas, etc.). **Después la seguridad lógica**, evitando poner los datos en disposición de personal no autorizado y protegiendo convenientemente las transmisiones.
  
- **Un cortafuegos (fire wall)** es una estrategia de defensa que se podemos considerar ideal. Este sistema permite restringir y controlar el tráfico de datos de la intranet pudiendo anular la acción de los curiosos.

---

<sup>29</sup> <http://www.intranet.com>

- No hay que olvidar la **encriptación**, asegura los datos en las transmisiones que se realizan de equipo a equipo. Existe un sistema denominado PGP(Pretty Good Privace), de utilización libre, especial para este tipo de métodos de salvaguarda de datos.
- La regla de oro es **mantener la intranet al día** en cuestiones de seguridad. Esto es un tema complejo que cambia tan rápido como la propia tecnología informática.

#### 1.4.12 PERSPECTIVAS DE FUTURO DE INTRANET <sup>30</sup>

Las predicciones para el crecimiento de Intranets se realizan sobre el estudio de las necesidades corporativas y de las demandas de mercado. Organizaciones de todos los tamaños se están enfrentando a retos importantes de comunicación y el entorno actual de cambios constantes y competitivo hace que el acceso rápido y fácil a la información departamental, corporativa y del consumidor sea esencial.

La influencia de las Intranets está siendo tan definitiva en el ámbito organizacional, que, probablemente, se hable de un antes y un después de la revolución de las Intranets.

Todo apunta a que el desarrollo de estas redes en los próximos años será más que significativo, puesto que miles de compañías se encuentran en estos momentos investigando en el fenómeno Intranets. Por otro lado, datos muy fiables dicen que el número de herramientas para su desarrollo de las Intranets se cuadruplicará en el próximo año, la cual puede aportar una idea de la importancia del futuro de las Intranets para la informática corporativa.

---

<sup>30</sup> <http://www.srt.com-ar/intranet.html>

- De manera más sofisticada un intranet deberá permitir **la conversación y el vídeo en tiempo real**. Algunos ejemplos son la videoconferencia, las conversaciones en tiempo real, la telefonía por Internet o la multiconferencia.
- Del mismo modo que ya es posible acceder a **Internet a través del cable**, en poco tiempo también será posible crear nuevas tecnologías multimedia de intranets que podrán llegar hasta nosotros de este modo, ejemplo de ellos son los Network Computing, la Televisión por cable que ya es una realidad, etc.

Las intranet actuales en muy poco tiempo deberán ampliar sus servicios y aplicaciones a la inmensa Internet, usando un medio que permita controlar el acceso para conceder privilegios a determinados **usuarios exteriores** seleccionados, sin necesidad de modificar contenidos o componentes de aplicaciones, y sin que ello suponga un riesgo importante para la adecuada seguridad de la intranet.

#### 1.4.13 NUEVAS TECNOLOGÍAS DE LA INFORMACIÓN <sup>31</sup>

Cabe destacar que la mayor parte de las grandes corporaciones están desarrollando ya o van a desarrollar *Intranets*, con la convicción de que las *INTRANET* serán el punto neurálgico utilizado para el acceso tanto a las nuevas aplicaciones como a las ya establecidas.

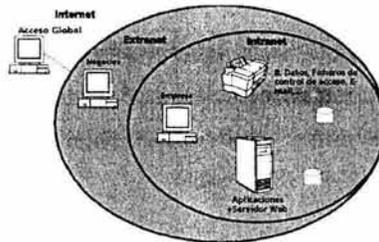
Dentro del campo de las nuevas tecnologías podemos encontrar tres movimientos importantes que marcan su evolución. El primero de ellos fue, sin duda alguna, el fenómeno Internet y su desarrollo definitivo a través del WWW. Más que un simple vehículo de difusión de información, Internet se ha convertido en la base común de trabajo y la fusión real de los entornos abiertos.

El segundo paso importante se produjo con la llegada del fenómeno *INTRANET*. Se consideran áreas seguras donde se utiliza la tecnología Internet (especialmente WWW) para desarrollar las comunicaciones internas y actividades de colaboración dentro de una empresa u organización.

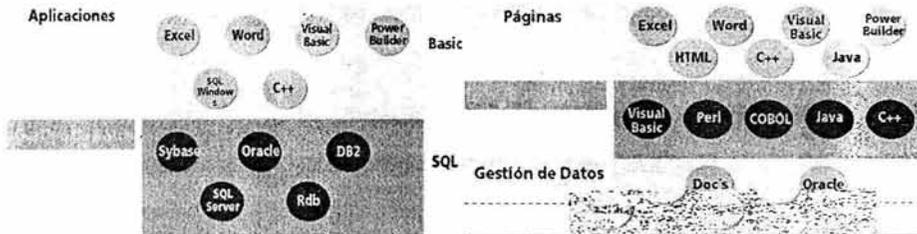
---

<sup>31</sup> <http://kcinfos.com/web/intranet/>

El tercer gran fenómeno, fruto de la natural evolución de los dos anteriores, fue el desarrollo de *Intranets* a un nivel aún más colaborativo: Son las denominadas Extranets. Representan realmente el puente entre la Internet (pública) y las *Intranets* corporativas (privadas).



También Extranet puede ser definida como una red colaborativa que utiliza tecnología Internet para el enlazar negocios con proveedores, clientes y otros negocios que compartan intereses comunes. Incluso el antiguo modelo cliente/servidor está evolucionando hacia la incorporación de páginas dinámicas y procesos tal como se muestra a continuación.



#### 1.4.14 RECURSOS INTRANET <sup>32</sup>

La principal cuestión que se plantea a la hora de desarrollar en *INTRANET* es saber exactamente qué tipo de tecnología será la más acertada. En una *INTRANET*, los programas pueden ejecutarse tanto sobre el servidor como sobre el cliente.

<sup>32</sup> <http://keinfos.com>

La tendencia es una ejecución desde el lado del cliente, lo que proporciona mejores resultados, permite eventos locales (por ejemplo, entrada de datos o el click de un ratón) y una interacción directa en la ejecución. El principal problema con las técnicas de ejecución desde el lado del cliente es la escasez de estándares.

La tecnología está siendo desarrollada por vendedores independientes, cada uno con un lenguaje propio. Resulta vital saber diferenciar con claridad entre todo el abanico de posibilidades y recursos que existen para dicho desarrollo. En los siguientes apartados haremos una breve referencia a cada uno de ellos.

Por un lado encontramos los primeros avances dentro de la dinámica del WWW a través de programas externos más conocidos como programas *gateway* (pasarela).

El protocolo estándar que gobierna este intercambio es conocido como **CGI** (*Common Gateway Interface*). Los programas *gateway* son llamados desde el navegador igual que cualquier otro recurso. CGI es un estándar especificado y mantenido por el National Center for Supercomputing Applications (NCSA).

**JavaScript** es un lenguaje que fue desarrollado por *Netscape Communications*. Originalmente *JavaScript* nació bajo el nombre de *LiveScript*, pero en noviembre de 1995, cuando *Sun Microsystems* y *Netscape* concretaron que *Java* era la rápida culminación de la tecnología más esperada del año, entonces *LiveScript* fue renombrado como *JavaScript*. Ambos lenguajes son ciertamente buenos para crear páginas dinámicas.

Los orígenes de **Java** se centran en un trabajo de *Sun Microsystems* encaminado a crear un lenguaje de programación en el que se pudiera escribir software ejecutable en dispositivos de muy distinto tipo. Después evolucionó hasta convertirse en un lenguaje apto para la distribución de contenidos ejecutables a través del WWW.

Para que el usuario pueda manejar contenidos *Java* debe disponer de la pieza clave de software: El intérprete *Java*. Para visualizar contenidos el navegador del usuario debe estar preparado para entender *Java*.



Lo más significativo es que *Java* no está sujeto a un tipo de plataforma de computadoras (*Java* es neutral a la arquitectura). La neutralidad en la arquitectura es una manera complicada para decir que los programas *Java* pueden ejecutarse sobre cualquier computadora o plataforma.

**Visual Basic Script** (VB Script) y **ActiveX** son dos nuevas tecnologías introducidas por Microsoft. Son similares a *JavaScript* y *Java*, respectivamente, en donde el primero es un lenguaje *script* para navegadores y el segundo es la forma de incrementar la funcionalidad a los mismos navegadores Web. Juntos, suministran una potencia y flexibilidad destacables. VB *Script* es un lenguaje *script* que permite incluir comandos en los documentos HTML.

Al igual que *JavaScript*, VB *Script* es un lenguaje interpretado. En el entorno *ActiveX* un cliente solicita una página HTML a través del navegador al servidor Web. Si la página HTML contiene controles *ActiveX*, los controles son transferidos hacia el cliente en caso de no estar presentes. Los controles de *ActiveX*, una vez transferidos, se ejecutan enteramente en el cliente.

#### 1.4.14.1 ACCESO A BASES DE DATOS <sup>33</sup>

A través de los diferentes recursos antes descritos será posible realizar una interacción con la base de datos. Para conseguir este objetivo, los desarrolladores de aplicaciones pueden seguir caminos diferentes, cada uno de ellos con sus propias herramientas asociadas y sus inconvenientes potenciales.

Una manera de integrar las bases de datos con los servidores y navegadores Web consiste en escribir sencillos programas de servidor basados en CGI, el estándar Web de acceso a programas externos. Una variante de este enfoque es crear conexiones CGI a la interfaz ODBC (Open Database Connectivity).

Otra opción consiste en utilizar ODBC junto con interfaces de servidor Web propietarios más flexibles y eficientes -como ISAPI (Information Server API) de Microsoft o NSAPI (*Netscape* API) de *Netscape* Communications- para acceder a las bases de datos remotas.

Los primeros pasos en este sentido partieron del denominado Internet Database Connector IDC de Microsoft, que no era más que una definición simple y poco potente de cómo interactuar desde una página HTML (en este caso con la combinación de una petición en una página con extensión .IDC y volcando los resultados en otra con formato .HTX) con la base de datos. Dicha solución aunque vigente, está dando paso a nuevas formas más avanzadas de interacción. Finalmente, los desarrolladores pueden eludir el servidor Web en su totalidad y usar un producto de middleware de terceros capaz de implementar algunas de esas interfaces.

#### 1.4.15 LOS NUEVOS USOS DE LA INTRANET <sup>34</sup>

La Intranet es una herramienta implantada habitualmente en grandes corporaciones, que ahora con la nueva tecnología se pone a la disposición de cualquier empresa. Permitiendo a las PYMES desplegar nuevas estrategias de comunicación dentro del trabajo interno.

---

<sup>33</sup> <http://www.keinfos.com>



La Intranet promete transformar el mundo corporativo de la misma forma. Tanto compañías como empresas pequeñas han implementado esta tecnología, mejorando la productividad mientras se reducen sus costos.

#### 1.4.15.1 UNA NUEVA FORMA DE TRABAJAR. <sup>35</sup>

Las organizaciones viven de la información, pero su acceso no es siempre fácil de tener. Las personas tienden a depender de la interconexión informal, la clásica rutina de "a quién conoces". Con una Intranet, existe un repositorio a lo largo de la empresa, un lugar al cual acudir para consultar un reporte anual, un manual de prestaciones, reservar la sala de conferencias, localizar una forma de reporte de gastos o revisar las políticas de vacaciones de la compañía.

Esta información está disponible las 24 horas del día, los siete días de la semana para cualquiera que tenga acceso, ya sea estando de viaje, trabajando en casa o en una oficina remota. No se tiene que esperar a que la respuesta a una pregunta viaje lentamente por los canales tradicionales de la compañía.

Características más sofisticadas de la Intranet permiten a los empleados llenar formularios en línea, consultar archivos visuales y ordenar materiales.

---

<sup>35</sup> <http://www.keinfos.com>

#### 1.4.15.2 APLICACIONES DE LA INTRANET PARA EMPRESAS:

##### Recursos Humanos

- Los Gerentes de la empresa pueden comunicarse con todos sus empleados, incluso los que se encuentran en sucursales remotas, a través de notas publicadas en la Intranet, e-mails, pizarrón de avisos, entre otros.

##### Ingeniería

- Pueden realizarse reservación de salas de juntas, minutas, semáforos, "comités" virtuales, para autorización de créditos, para evaluación de personal, incluso entre personas que están en distintas sucursales.

<sup>35</sup> <http://infoperu.com/espanol/intranet.html>

##### Atención al Cliente

- Soporte y ayuda en todas las áreas, proveídos por los propios miembros de la Intranet, gracias a la fluida comunicación entre ellos, a través de e-mails, chats, foros.
- Empleados que necesiten acceder, por ejemplo, al departamento de compras, podrán hacer órdenes de útiles o insumos on-line, y Usted podrá controlar la actividad diaria en tiempo real.

##### Contabilidad y Finanzas

- Podrá realizar consultas en línea de los indicadores financieros de procesos de ventas, compras, etc.

## Telecomuting

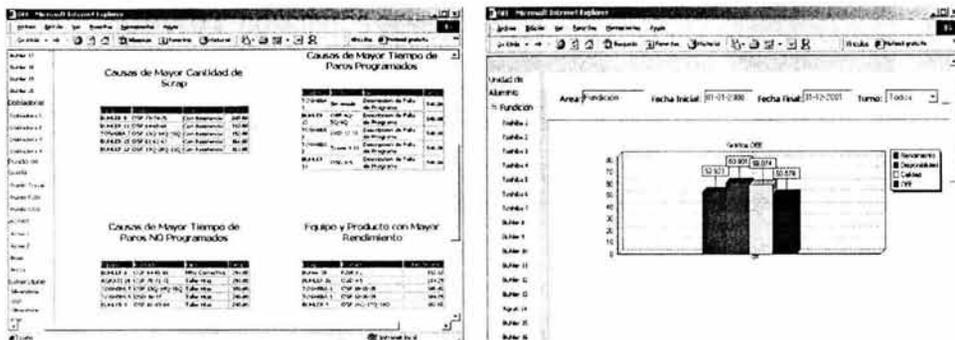
- Usted podrá realizar Telecomuting, es decir, ingresar desde su casa a la Intranet, ubicándose "virtualmente" en su escritorio.

The screenshot shows a window titled 'Modulo de Intranet Desktop'. The interface includes a menu bar (Archivo, Edición, Formato, Ayuda), a toolbar, and a main data area. The data area contains a table with the following columns: 'Cuenta', 'Debitos', 'Credito', 'Saldo', 'Debitos', 'Credito', 'Saldo', 'Debitos', 'Credito', 'Saldo', 'Debitos', 'Credito', 'Saldo'. The table lists various accounts and their corresponding financial values.

Aplicaciones de la Intranet para los clientes de la empresa:

## Mercadeo y Ventas

- Ventas, Marketing y Atención al cliente pueden ofrecer información de último momento siempre. Dando a sus clientes acceso limitado a zonas de su Intranet donde el cliente busca información, la empresa se ahorra gastos de atención al cliente.
- Dando acceso limitado a sus proveedores, la empresa mejora la comunicación y el intercambio de información con ellos.
- La empresa puede abrir sus aplicaciones a Internet, para tener acceso a más cantidad de clientes de todas partes del mundo.



Las Intranets ofrecen tanto ventajas estratégicas como tácticas además de hacer más eficientes los procesos de negocios, mejorar las comunicaciones, incrementar la colaboración personal y ayudar a amortizar los sistemas informáticos. Las Intranets permiten a las organizaciones proveer servicios que serían difíciles de prestar o demasiado costosos de desarrollar con los sistemas propietarios anteriores.

Las Intranets no solo son para compañías grandes, se están desarrollando por las empresas que desean incrementar su productividad y hacerse más ágiles a los cambios del entorno del mercado o tecnológico. Las Intranets también son una manera importante de aminorar las inversiones en cómputo, ya que utilizan la infraestructura de servidores, comunicaciones y saca más provecho a los computadores personales.

#### 1.4.15.3 CARACTERÍSTICAS FUNCIONALES 36

- Sencillez de funcionamiento para el usuario.
- Flexibilidad y permanente actualización.
- Hace interactiva a la empresa.
- Es una tecnología cliente/servidor.
- Compatibilidad con cualquier tipo de plataforma informática.
- Facilita la diseminación de información.

- Identifica procesos que pueden beneficiar a la empresa.

#### 1.4.15.4 REQUISITOS TÉCNICOS <sup>37</sup>

Diseñar y mejorar su red de comunicaciones con el servicio de Diseño e Implementación de Redes LAN/WAN para adaptar la carga de su Intranet.

- Servidor (Web Server)
- Sistema Operativo (Linux, Windows NT, Unix)
- Protocolo (Dependerá del Sistema Operativo)
- Browser (Internet Explorer, Netscape Navigator, etc.)
- Firewall ( Para prevenir los accesos no autorizados, o el ataque malicioso a sus sistemas tanto desde de fuera, como dentro de su organización).

#### 1.4.15.5 FACILITAR LA CAPACITACIÓN Y COMPARTIR EL CONOCIMIENTO. <sup>38</sup>



El sitio de Intranet favorito de los gerentes de J.C. Penney es el Sistema de Gestión del Conocimiento de la compañía, que se utiliza para capacitación interactiva y aprendizaje a distancia. Al ingresar, cada gerente ve una pantalla adaptada a su función y a su nivel, con programas de capacitación y consejos. "Ya no publicamos o enviamos material de entrenamiento", dice Deborah Masten, directora de Comunicación y desarrollo de Recursos Humanos en la sede central de la compañía, en Plano, Texas.

#### 1.4.15.6 AUTOMATIZAR TAREAS DE RUTINA. <sup>39</sup>

"En Microsoft hacemos todo en la Intranet", dice Mary Lee Kennedy, directora de servicios de información en la sede central de la compañía de software en Redmond, Washington. La Intranet de Microsoft alberga alrededor de 1,000 sitios, más de 2 millones de páginas y docenas de aplicaciones y funcionalidades. Todos los días, casi un tercio de los 30,000 empleados de la compañía ingresa a MSWeb, el principal sitio de propósitos generales. Otro favorito es MSMarket, donde los empleados pueden comprar suministros de oficina y reservar pasajes para viajes. El sitio ahorra tiempo a los empleados, y es el sueño de cualquier departamento de compras porque integra sin problemas a los proveedores.

#### 1.4.15.7 PROVEER FÁCIL ACCESO A LA INFORMACIÓN. <sup>40</sup>



Hace cuatro años, los ejecutivos de Kraft Foods advirtieron la necesidad de coordinar el uso de la Web entre diferentes divisiones de la compañía. "Todos los grupos desarrollaban páginas Web al mismo tiempo. Queríamos estar seguros de que no estuvieran 'reinventando la rueda' y duplicando esfuerzos", recuerda Karen Isaacson, directora de información y tecnología de Recursos Humanos de la compañía, con sede en Northfield, Illinois.

Kraft designó un comité de Intranet, compuesto por representantes de los departamentos de asuntos corporativos, recursos humanos, legales y TI. El resultado fue el Kraft Intranet Café, un asombroso sitio Web que hace las veces de home page para cada empleado. Brinda a los usuarios fácil acceso al resto de los sitios de Intranet de la compañía, como por ejemplo HR Online.

---

<sup>39,40</sup> <http://www.kcmfos.com>

## CAPITULO 2. ANÁLISIS Y DISEÑO DE SISTEMAS

### 2.1 INTRODUCCIÓN

Las metodologías y técnicas estructurales que han reinado en la década de los setenta y ochenta no han eliminado el problema de análisis y diseño de información, y de hecho la crisis del software continua, hoy en día pese a las muchas técnicas y métodos existentes, esto se debe a la complejidad de los problemas organizacionales que han crecido considerablemente.

Los cambios realizados en la evolución de un programa son el punto débil de los métodos tradicionales de desarrollo de software, siendo uno de los puntos fuertes de los métodos de desarrollo de software. Recientemente, las propuestas de reusabilidad o reutilización de componentes de software, se consideran como bloques iniciales para la construcción de programas, de modo similar a la construcción de cualquier objeto complejo que se construye ensamblando partes.

Dentro de las organizaciones, el Análisis de Sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados, por consiguiente, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

Los problemas que se intentan resolver con software implican elementos de ineludible complejidad, en los que se encuentra una gran cantidad de requisitos, en muchas ocasiones contradictorias. Esta complejidad se produce por las difíciles interacciones entre los usuarios de un sistema y sus desarrolladores: los usuarios encuentran generalmente muy difícil dar precisión sobre sus necesidades de forma que los desarrolladores puedan comprender. En casos extremos, los usuarios solo pueden tener ideas vagas de lo que se desea en un sistema de software.

Por otra parte los usuarios y desarrolladores tiene diferentes perspectivas de la naturaleza del problema y hacen suposiciones diferentes sobre la naturaleza de la solución.

Dentro de toda esta problemática el desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: **el análisis de sistemas y el diseño de sistemas**.

El diseño de sistemas es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. Pero antes de llevar a cabo esta planeación es necesario comprender, en su totalidad, el viejo sistema y determinar la mejor forma en que se pueden, si es posible, utilizar las computadoras para hacer la operación más eficiente.

El análisis de sistemas por consiguiente, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema. Este es en sí el trabajo del analista de sistemas.<sup>1</sup>

**“EL ANÁLISIS ESPECIFICA QUÉ ES LO QUE EL SISTEMA DEBE HACER. EL DISEÑO ESTABLECE COMO ALCANZAR EL OBJETIVO.”<sup>1</sup>**

## 2.2 ANÁLISIS DE INFORMACIÓN

La descripción dada hasta el momento del análisis de sistemas brinda un panorama de lo que hace el analista. Las responsabilidades de los analistas, sin embargo, así como su denominación dentro de una empresa, cambian de una organización a otra. A continuación se muestra una lista de las funciones más comunes asignadas a los analistas de sistemas.

1. **Análisis de Sistemas:** en este caso la única responsabilidad del analista es conducir estudios de sistemas para detectar hechos relevantes relacionados con la actividad de la empresa. La función más importante en este caso es reunir información y determinar los requerimientos. Los analistas no son responsables del diseño de sistemas.

---

<sup>1</sup> Análisis y Diseño de Sistemas de Información

2. **Análisis y Diseño de Sistemas:** además de llevar a cabo el estudio completo de los sistemas, el analista tiene la responsabilidad adicional de diseñar el nuevo sistema. Los que se responsabilizan tanto del análisis como del diseño trabajan en menos proyectos que los analistas de información, pero invierten más tiempo en ellos.
3. **Análisis, Diseño y Programación de Sistemas:** el analista conduce la investigación de sistemas, desarrolla las especificaciones de diseño y escribe el software necesario para implantar el diseño.

El análisis y diseño de sistemas se ha colocado en un lugar principal dentro de las organizaciones dado que los tomadores de decisiones comprenden que la información no es solo un subproducto de la conducción, sino que a la vez alimenta a los negocios y puede ser el factor crítico para la determinación del éxito o fracaso de estos.

Para maximizar la utilidad de la información una empresa la debe manejar correctamente tal como se manejan los demás recursos.

Los administradores necesitan comprender que hay costos asociados con la producción, distribución, seguridad, almacenamiento y recuperación de toda información. Aunque la información se encuentra a nuestro alrededor ésta no es gratis, y su uso es estratégico para posicionar la competitividad de un negocio.

Aunado a esto la fácil disponibilidad de las computadoras ha creado una explosión de información a través de la sociedad en general y de los negocios en particular. El manejo de la información generada por computadora difiere de en forma significativa del manejo de datos producidos manualmente.

Es por esta razón que los sistemas de información son desarrollados con propósitos diferentes, dependiendo de las necesidades del negocio. Los sistemas de procesamiento de transacciones por ejemplo.

El objetivo principal de esta tesis es llevar a cabo el Análisis, Diseño y programación de un Sistema de Procesamiento de Transacciones que tiene las siguientes características generales:

- Son sistemas de información computarizada desarrollados para procesar gran cantidad de datos para transacciones rutinarias de los negocios.
- Una transacción es cualquier suceso o actividad que afecta a toda la organización. el procesamiento de transacciones, que es el conjunto de procedimientos para el manejo de éstas incluye entre otras las siguientes actividades: Cálculo, clasificación, ordenamiento, almacenamiento y recuperación y generación de resúmenes.
- Estos sistemas eliminan en gran parte el tedio de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se utilizó para llevarlas a cabo manualmente, aunque el usuario aun debe alimentar datos a los sistemas computarizados.
- Estos sistemas brindan velocidad y exactitud; además se pueden programar para seguir rutinas sin ninguna variación
- Los sistemas de procesamiento de transacciones, son sistemas que traspasan fronteras y que permiten que la organización interactúe con ambientes externos.
- Debido a que los administradores consultan los datos generados al minuto acerca de lo que esta pasando en la organización, es esencial para las operaciones diarias que estos sistemas funcionen confiablemente y sin interrupción.

Hemos detallado la gran importancia que tienen el análisis y diseño de sistemas pero definitivamente nos ha faltado mencionar una parte imprescindible en este proceso y esta parte son los **"usuarios"**.

Los sistemas de información con mayor éxito – éxito en términos de beneficio para la empresa – se originan con los usuarios. Una razón para ello es que las solicitudes de estos sistemas se originan desde una necesidad de la organización que los usuarios perciben: por ejemplo, la necesidad de resolver un problema en particular, de manejar funciones rutinarias, o de monitorear la información para evitar ciertos problemas.

El hecho de que en las empresas los usuarios contribuyan con ideas que conduzcan hacia sistemas con éxito, tal como debe ser, demuestra que el propósito fundamental de un sistema de información, y el más importante, es mejorar la organización y no el de probar el valor de una tecnología sofisticada.

El desarrollo de sistemas con éxito, sin embargo es un esfuerzo conjunto. Las contribuciones de los usuarios son importantes y los analistas tienen un papel esencial: ***Extraer las mejores ideas de los usuarios para su análisis y Discusión.***<sup>2</sup>

Se ha mencionado repetidas veces la palabra "sistema", pero ¿Qué es un sistema?

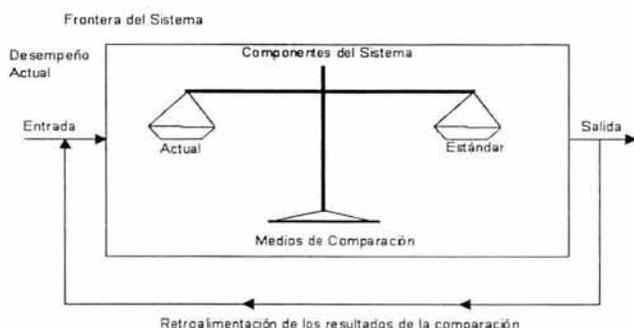
En le sentido más amplio un sistemas es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. Por ejemplo : Una organización es un sistema. sus componentes, mercadotecnia, manufactura, ventas investigación, embarques, contabilidad y personal, trabajan juntos para crear utilidades que beneficien tanto a los empleados como a los accionistas de la compañía. Cada una de estos es a su vez un sistema.

---

<sup>2</sup> Análisis y Diseño de Sistemas de Información. Senn James

Todo sistema organizacional depende, en mayor o menor medida, de una entidad abstracta denominada sistema de información. Este sistema es el medio por el cual los datos fluyen de una persona o departamento hacia otros.

Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que estos trabajen con eficiencia para alcanzar el mismo objetivo.



**Elementos básicos de control de un modelo de sistemas**

## 2.3 CLASIFICACIÓN DE ANÁLISIS DE INFORMACIÓN <sup>3</sup>

### 2.3.1 ANÁLISIS DEL MODELO FUNCIONAL

En el modelo funcional, el objetivo de los analistas es el de identificar y analizar las áreas funcionales, funciones y actividades que se realizan para llevar a cabo los objetivos de la empresa. Este tipo de información constituye el punto de partida para la agrupación adecuada de la información que se empleará para construcción de los modelos de datos.

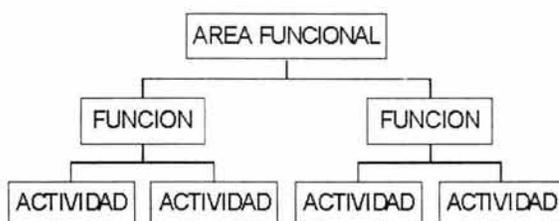
<sup>3</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

Las tareas principales para llevar a cabo el análisis del modelo funcional son:

- Identificar y analizar las áreas funcionales
- Identificar y analizar las funciones de cada área funcional
- Identificar y analizar las actividades de cada función
- Revisar y analizar el modelo funcional con los ejecutivos de alto nivel

El modelo funcional es representado por un documento analítico que muestra el conjunto de áreas funcionales de la organización con sus funciones y actividades el cual se puede representar de formas. La primera forma consta de una tabla que contiene el nombre del área funcional con sus funciones y su respectiva actividad por función

La segunda forma de representar a un Modelo Funcional puede llevar a cabo mediante un organigrama como se muestra en la figura.



El Modelo Funcional sirve como base para el análisis del Modelo de Datos, porque del Modelo Funcional se extraen datos de acuerdo a las actividades y funciones analizadas con el objetivo de obtener información de:

Las formas y los reportes que se emplean:

- El origen y destino del reporte y documento

- Las características de su uso
- Los catálogos que se emplean en la elaboración del reporte
- La descripción y el formato de cada dato

### 2.3.2 ANÁLISIS DEL MODELO DE DATOS

Este modelo de datos es la base para sistemas con propósito de toma de decisiones de la parte directiva y los mismos datos se aplican en usos múltiples. Para este método se deben tomar en cuenta conceptos asociados con las bases de datos y los sistemas para su manipulación.

Los diagramas de entidad-relación y los diagramas de estructuras de datos son herramientas de utilidad para el diseño y poder llevar a cabo la aplicación de este modelo.

Esta filosofía de diseño marca que los datos son el centro de procesamiento de datos, en donde los datos son almacenados y mantenidos con el apoyo de distintas transacciones, este almacenamiento y mantenimiento es efectuado por un sistema manejador de base de datos (DBMS: Data Base Manager System) auxiliándose de transacciones para la producción de información, es decir, existe una generación de datos así como su actualización, este modelo con los datos, el DBMS y las transacciones da como resultado generación de documentos, generación de gráficas, así como apoyo en la toma de decisiones, auditorías y búsqueda de información.

### 2.3.3 DIAGRAMA DE ESTRUCTURA DE DATOS <sup>4</sup>

Esta técnica considera a los datos independientes del procesamiento que transforma los datos, esto no indica que se omitan. Esta modelización de datos como su nombre lo indica modeliza datos sin preocuparse por el procesamiento que se deba aplicar para transformar los datos, siendo por tanto una

---

<sup>4</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

técnica complementaria en donde se puede combinar con otro enfoque de modelo que se haga cargo de la modelización que manipule aspectos de procesamiento para un análisis completo.

La modelización de datos es ampliamente en aplicaciones de bases de datos. Esto proporciona un gran panorama para analistas y diseñadores de la base de datos una amplia visión de los datos y sus relaciones. Los datos pueden ser entidades externas, cosas, concurrencias, sucesos, papeles, estructuras o lugares.

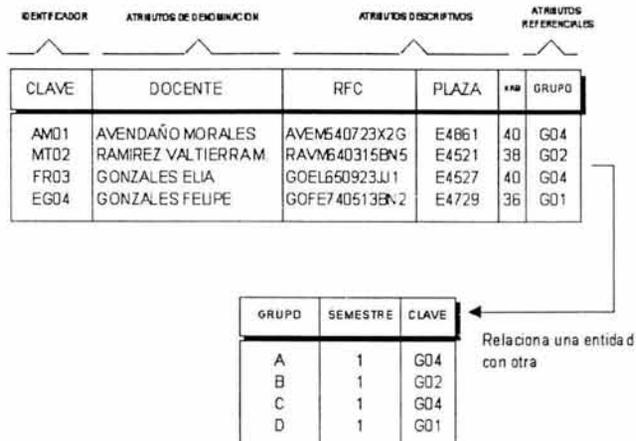
Por ejemplo: En un plantel se cuenta con docentes y alumnos, los cuales pueden ser tomados como datos puesto que estos contienen un conjunto de atributos.

Los atributos de las entidades u objetos de datos se caracterizan por 3 aspectos:

- Dar nombre a una creación de entidad
- Describir la entidad
- Hacer referencia a otra creación de otra tabla de atributos u/o otra entidad

Además de que se pueden utilizar uno o más atributos para relacionarse con otra entidad o entidades. Esto se puede llevar a cabo utilizando un modelo relacional de los datos siendo la aplicación sobre tablas (Figura 2.3) en donde se muestran los atributos y su relación tomando en cuenta la optimización de relaciones redundantes. Estas reglas de normalización son las siguientes:

- Una entidad tiene un valor y sólo un valor para cada atributo.
- Los atributos representan la unidad mínima, es decir, no contienen estructuras.
- Cuando se utiliza más de un atributo para identificar una entidad, hay que asegurarse que estos atributos representen una características completa de la entidad.
- Todos los atributos que no sean identificadores deben representar características de la entidad nombrada.



### 2.3.4 CARTA DE ENTIDADES. <sup>5</sup>

Para la modelización de datos se utiliza una notación denominada "Diagrama de Entidad-Relación", siendo su propósito principal representar a cada entidad y su relación mostrando la clase de información necesaria para satisfacer un requerimiento de información.

Su notación es sencilla, las entidades se representan con rectángulos etiquetados, las conexiones entre entidades se representan mediante varias líneas de conexión con un formato especial.

La modelización de datos y el diagrama de Entidad-Relación se convierten para el analista una notación que produce resultados concisos para ser examinados en el procesamiento de datos.

### 2.3.5 CARTA DE BURBUJAS. <sup>6</sup>

Debido a que los datos se mueven a través del sistema sufre constantes modificaciones, los diagramas son técnicas para representar el flujo de la información desde la entrada, proceso, hasta su salida.

<sup>5,6</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

El diagrama de Burbuja o diagrama de flujo de datos puede representar un sistema a cualquier nivel de abstracción representando así un mayor flujo de información y un mejor detalle funcional, el primer nivel o nivel 0 también denominado modelo fundamental del sistema del cual se puede elaborar en subniveles más diagramas partiendo del nivel 0.

En los diagramas de burbuja la notación que se usa para su creación consiste en un rectángulo el cual representa una entidad externa (hardware, persona, otro programa, etc..) u otro sistema que genere información a ser transformada por el software o sistema.

El círculo representa un proceso o transformación aplicado a los datos, las flechas utilizadas en el diagrama de burbuja indican el flujo y estas deben estar acompañadas por una etiqueta.

La doble línea representa almacenamiento de información que es utilizada por el sistema.

Cabe mencionar que el diagrama de burbuja no representa ninguna información explícita de la secuencia de procesamiento, estando implícitamente en el diagrama.

### **2.3.6 ESQUEMA GENERAL DEL SISTEMA.**

Este tipo de diagrama muestra la operación propuesta para el nuevo sistema, en el cual se representan las transacciones manuales y las transacciones automatizables, las cuales interactúan con la base de datos.

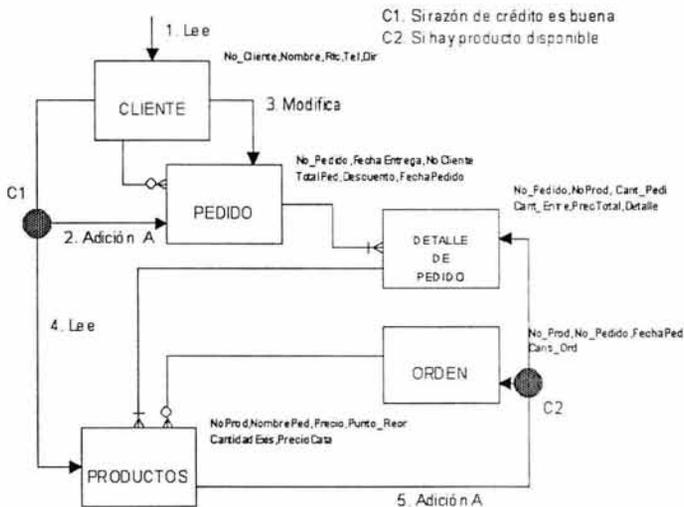
Su notación es simple utilizando rectángulos sencillos para indicar las transacciones manuales y rectángulos dobles para indicar las transacciones automatizables utilizando las flechas que indican la dirección de la transacción.

### 2.3.7 MAPA DE ACCESO LÓGICO

Este tipo de diagrama sirve para representar la secuencia del acceso lógico de una transacción a un modelo de datos.

siendo su notación básica similar a un Diagrama de Entidad-Relación con la diferencia de que en el Diagrama de Mapa de Acceso Lógico muestra unas líneas que indican la dirección etiquetadas por el tipo de operación que se efectúa, estas operaciones afectan a los datos como: Lectura, modificación, borrado o agregar datos.

Las entidades representadas por un rectángulo van acompañadas en su parte exterior por los atributos de cada entidad.



Cabe mencionar que dentro de la nomenclatura básica de un Mapa de Acceso Lógico existe la aplicación de condiciones representadas por un punto etiquetado el cual establece una conexión, en el

Mapa de Acceso Lógico se agregará la etiqueta de condición acompañada de la descripción de la condición.

### 2.3.8 MINI ESPECIFICACIÓN

La mini especificación es un listado de proposiciones en un lenguaje parecido a cualquier lenguaje natural como el Español o el Inglés. Cada una de estas proposiciones especifican un paso de transacción o Actividad en un Modelo Funcional o un proceso en un Diagrama de Flujo de Datos.<sup>6</sup>

Debido a que es una narrativa desarrollada con nuestras propias palabras no existe una nomenclatura estricta a seguir. Por lo cual se pueden aplicar palabras o frases comunes de nuestro lenguaje cotidiano. Por ejemplo:

INICIO	RESTA
FIN	SUMA
SI/ENTONCES/FIN DEL SI	MULTIPLICA
SI/DE LO CONTRARIO	DIVIDE
ESCRIBE	ELIMINA
LEE	BORRA
MODIFICA	ACTUALIZA
REPITE HASTA / FIN DEL REPITE	SALIR
ENVÍA MENSAJE	COMENTARIO
CALCULA	
+,*,-,/	

<sup>6</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

La mini especificación únicamente lista actividades o transacciones de una parte del sistema en donde se omite la especificación de la presentación de la información visual en pantalla o impresora (color, tipo de letra, posición, cuadros, etc...) ya que se presenta a detalle la transacción, más no aspectos secundarios, los cuales se describen en formatos especiales como son: Formatos de Pantallas y Reportes

### 2.3.9 FORMATO DE PANTALLA Y REPORTE

La mayoría de los sistemas en un análisis requieren de una planeación de la forma en que se obtendrán los datos, así como la forma en que el sistema presentará los resultados, es decir se requiere de una planeación de la distribución de los atributos de las entidades presentadas en pantalla y en papel. <sup>7</sup>

Los formatos de pantalla van acompañados de una hoja de especificación de colores, su función es la de establecer una estandarización en los colores que se aplicaran en las pantallas, en donde además cada color debe estar acompañado por una descripción que representa una acción en el sistema. Por ejemplo:

HOJA DE ESPECIFICACION DE COLORES	
□ □ □	Se utiliza para pantallas de captura, modificaciones, consultas y bajas
□ □ □	Indica un mensaje de error enviado por el sistema
	Pantallas de ayuda (Cualquier tipo de

<sup>7</sup> Diseño de Sistemas de Información

	□□□	ayuda)	
		Para aquellas pantallas donde existan preguntas y menús de barra iluminada	
	□□□	Para indicar la barra iluminada de cada menú	
	□□□		

Los formatos de pantalla deben contener una escala para columnas y filas que representen la cantidad máxima de filas y columnas (Por ejemplo: si se está presentando el sistema en el modo texto, la escala en los formatos debe ser de 80 columnas y 24 filas).

El formato de pantalla también debe contener una descripción de los campos que se aplican en la pantalla con su nombre, tipo y longitud representando en la pantalla el espacio físico que ocuparía realmente.

Cabe mencionar que los campos de tipo carácter o alfanuméricos se representan por medio de una equis ("x") y los campos de tipo numérico se representan por un nueve ("9").

Los formatos de reporte sirven para indicar como se representará la información producida por el sistema en hoja.

Estos deben contener una escala que representa la cantidad máxima de columnas de acuerdo a la capacidad de la impresora que se utilizará o al tipo de letra (Por ejemplo: podemos contar con una impresora de 10" de matriz de puntos y en letra normal el espacio máximo ocupado será de 80 columnas, pero si utilizamos letra comprimida en la misma impresora puede ocupar un máximo de 130 columnas). Los formatos de reporte también van acompañados de una descripción de campos que indique su nombre, tipo y longitud.

La información contenida en cada campo al igual que los formatos de pantalla también se deben representar físicamente en el formato de reporte indicando con equis ("x") la longitud de los campos



**Mapa de transacciones.** Es relevante a una transacción, a través de una tabla de referencias donde muestra las rutas utilizadas en particular por dicha transacción, en donde su objetivo principal es hacer referencia al número de veces que los datos sufren una transacción o pasan por un proceso.

**Mapa de uso combinado.** Este documento se encarga de mostrar las transacciones concurrentes contra el modelo de datos, y la carga con la cual contribuye cada transacción, a través de cada ruta utilizada (asociación entre dos entidades), durante un período de procesamiento seleccionado.

**Mapa de carga compuesto.** Muestra las transacciones concurrentes contra el Modelo de Datos, y la carga con la cual contribuyen todas las transacciones a través de cada ruta utilizada (asociación entre dos entidades), durante un período de procesamiento seleccionado.

**Matriz de Carga.** Este documento de diseño es importante para ser utilizado durante el diseño físico de la base de datos; este resume la carga total del sistema sobre la base de datos para aquellas transacciones que se procesan concurrentemente durante su período definido; totaliza el número de referencias lógicas contra la base de datos para una transacción sencilla y el número total para aquellas transacciones procesadas durante un período.

**Modelo Relacional.** Diagrama que muestra el Modelo de Datos Lógico convertido en Modelo Físico, siendo su dependencia del tipo de manejador de base de datos utilizado.

### 2.3.11 ANÁLISIS DE TRANSACCIONES

En muchos sistemas un elemento de datos es el motivo para determinar uno o más flujos de información, que afectan a la función de acuerdo con el que determine dicho elemento de datos, a este elemento se le denomina transacción que desencadena otro flujo de datos a través de uno o varios caminos.<sup>9</sup>

---

<sup>9</sup> Análisis y Diseño de Sistemas de Información

Los pasos a seguir para manipular el flujo de transacción tienen el objetivo de transformar un diagrama de flujo de datos en la estructura de un programa.

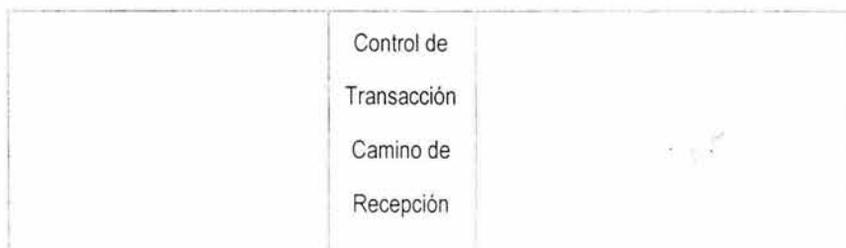
Revisión del modelo fundamental del sistema, que consiste en una revisión del diagrama de flujo de datos iniciando del nivel 0 y el nivel 1 que describen la especificación del sistema y la especificación de requisitos del sistema.

Refinar y revisar los diagramas de flujo de datos para el sistema, siendo una revisión a detalle de los niveles 1 y 2 para detectar con mayor posibilidad de éxito un diagrama de flujo de transacción.

‘Determinar si el diagrama de flujo de datos tiene características de transformación o de transacción, en donde el diseñador se encarga de determinar si la característica natural del diagrama de flujo de datos pertenece a un flujo de transacción.

Identificar el centro de transacción y las características de cada camino de acción, para detectar un flujo de transacción cuando es el origen de varios flujos de información que surgen de él, tratando de aislar las burbujas.

Transformar el diagrama de flujo de datos en una estructura adecuada al procesamiento de transacciones, llevándose a cabo creando subgrupos de burbujas a lo largo de un camino, así el flujo de cada camino de acción del diagrama de flujo de datos se convierte en una estructura con las características específicas de flujo.



**Correspondencias en la transacción**

Factorizar o Refinar la estructura de transacciones y la estructura de cada camino aplicando reglas de diseño para mejorar la calidad. Para factorizar se busca la mejor correspondencia entre los caminos, tratando de no omitir ninguno de ellos puesto que ocasionaría pérdida de una transacción en el sistema conllevando a que este no cumpla sus objetivos en su totalidad.

En el análisis de transacciones la estructura del programa, en sus módulos pueden expandirse o reducirse con el objetivo de mejorar su independencia, tomando en cuenta que la expansión de un módulo se divide en 2 o más módulos en la estructura de un programa final. Un módulo reducido es el resultado de la combinación del procesamiento involucrado en 2 o más módulos. Es necesario lograr una estructura en donde los módulos contengan varios niveles para una mejor entrada/salida ya que esto indica varias capas de control y gran aprovechamiento de recursos en los módulos de niveles inferiores.

Se debe tomar en cuenta que cuando hablamos de módulos, no solo nos referimos a los niveles iniciales 0, 1 ó 2 sino que también hablamos de aquellos que conforman al sistema los cuales nos muestran las transacciones en forma superficial, es decir, sin llegar al código ó al tipo de estructura de datos, uso de archivos, etc.

Es por ello que se recomienda evaluar las interfaces de los módulos para reducir la complejidad y la redundancia y así mejorar la consistencia de los módulos.

Una aplicación correcta del análisis de transacciones se complementa documentando al análisis realizando los siguientes pasos:

- Desarrollar para cada módulo, un texto explicativo del procesamiento
- Dar para cada módulo una descripción de la interfaz
- Definir las estructuras de datos globales y locales
- Anotar todas las restricciones/limitaciones de diseño
- Llevar a cabo una revisión del diseño preliminar registrando las actualizaciones.

Algunos aspectos típicos en las restricciones/limitaciones son el tipo de formato de datos, limitaciones de memoria, valores o cantidades límites para las estructuras de datos así como características especiales de un módulo individual.

Todo esto con el fin de reducir el número de errores antes de llegar al diseño o con ello justificar que no es viable su desarrollo.

Las actividades de revisión se centran en el seguimiento de los requisitos del sistema, la estructura del programa, en las descripciones de las interfaces, descripción de las estructuras de datos, la descripción de restricciones/limitaciones, en la facilidad de implementación y desarrollo de pruebas, y en la facilidad de mantenimiento.

### 2.3.12 ANÁLISIS DEL USO DE LOS DATOS

A medida que la información se mueve a través del software, está es modificada por una serie de transformaciones. El diagrama de flujo de datos (DFD) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.<sup>10</sup>

Se puede utilizar un DFD para representar cualquier software a cualquier nivel de abstracción. El rectángulo se utiliza para representar una unidad externa, es decir, un elemento del sistema (Ejemplo: Hardware, personas, conjunto de datos u otro programa) u otro sistema que produzca información a ser transformada por el software o que reciba información producida por el software.

---

<sup>10</sup> Diseño de Sistemas de Información

Un círculo representa un proceso o transformación que se aplica a los datos y los cambia de alguna manera. Todas las flechas que aparezcan en un diagrama de flujo de datos deben ser etiquetadas. La línea doble representa un almacén de información, Información almacenada que se utiliza por el software. La sencillez de la notación de un DFD es una de las razones por las que las técnicas de análisis estructurado son ampliamente utilizadas.

El Diagrama de Flujo de Datos es una herramienta gráfica que puede ser valiosa durante el análisis de requisitos del sistema. Cabe mencionar que el diagrama no proporciona ninguna secuencia explícita de los procesos

## 2.4 CARACTERÍSTICAS Y REQUERIMIENTOS DEL SISTEMA

La finalidad de un sistema es la razón de su existencia. Para alcanzar sus objetivos, los sistemas interactúan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas.<sup>11</sup>

Los sistemas que interactúan con su medio ambiente (reciben entradas y producen salidas) se denominan sistemas abiertos. En contraste aquellos que no interactúan con su medio ambiente se conocen como sistemas cerrados.

Los sistemas emplean un modelo de control básico consistente en :<sup>11</sup>

1. Un estándar para lograr un desempeño aceptable.
2. Un método para medir el desempeño actual.
3. Un medio para comparar el desempeño actual contra el estándar.
4. Un método de retroalimentación.

---

<sup>11</sup> Análisis y Diseño de Sistemas de Información.

El concepto de interacción con el medio ambiente que es lo que caracteriza a los sistemas abiertos, es esencial para el control. Recibir y evaluar la retroalimentación, permite al sistema determinar que tan bien está operando.

En contraste, los sistemas cerrados sostienen su nivel de operación siempre y cuando posean información de control adecuada y no necesite nada de su medio ambiente.

Dado que esta condición no puede sostenerse por mucho tiempo, la realidad es que no existen sistemas cerrados.

Los componentes que forman un sistema pueden ser a su vez sistemas más pequeños; es decir, los sistemas puede estar formados por varios niveles de sistemas o subsistemas.

**A continuación se enlistan los detalles más importantes para el análisis de sistemas:**

1. **Canales informales.** ¿ Que interacciones existen entre las personas y los departamentos que no aparecen en el organigrama o no están descritos en los procedimientos de operación ?
2. **Interdependencias.** ¿ De que otros departamentos y componentes de la organización depende un elemento en particular ?
3. **Personas y funciones clave.** ¿ Cuales son las personas y elementos más importantes en el sistema para que este tenga éxito ?
4. **Enlaces críticos de información.** ¿ Como es el flujo de información e instrucciones entre los distintos componentes de la organización ? ¿ Cómo se comunican las áreas entre sí ?

La anterior no es una lista exhaustiva de preguntas pero recalca la importancia de investigar y analizar la manera en que operan las organizaciones.

Durante el diseño los analistas tienen la responsabilidad de identificar las características importantes y necesarias que deben tener los nuevos sistemas. El analista especifica la forma en que se va a operar el sistema y sus subsistemas, las entradas requeridas, las salidas que se deben producir y los trabajos que se efectuarán tanto por las computadoras como en forma manual.

Por otro lado, los analistas también participan en el control de los sistemas básicamente en dos formas: La primera cuando describen los elementos de control, tales como estándares y métodos para evaluar el desempeño en relación con los demás estándares para los sistemas de información que diseñen.

Al mismo tiempo los sistemas que especifican proporcionan información a los directivos y usuarios que permite a estos determinar si los sistemas que administran operan correctamente.

Incorporar mecanismos de retroalimentación es un paso esencial en el diseño ya que su inclusión permite sostener las actividades de ambos sistemas. Ningún sistema perdurara si falta un control adecuado.

## 2.5 METODOLOGÍA <sup>12</sup>

Como se ha explicado anteriormente los sistemas de información, sirven para diversas finalidades que van desde el procesamiento de las transacciones de una empresa, hasta proveer de la información necesaria para decidir sobre asuntos que se presentan con frecuencia, asistencia a los altos funcionarios con la formulación de estrategias difíciles y la vinculación entre la información de las oficinas y los datos de toda corporación.

---

<sup>12</sup> Análisis y desarrollo de sistemas

En algunos casos los factores que deben considerarse en un proyecto de sistemas de información, tales como el aspecto más apropiado de la computadora o la tecnología de comunicaciones que se va a utilizar, el impacto del nuevo sistema sobre los empleados de la empresa y las características específicas que el sistema debe tener, se pueden determinar de manera secuencial.

En otros casos, debe ganarse experiencia por medio de la experimentación conforme el sistema evoluciona en etapas.

A medida que las computadoras son empleadas cada vez más por personas que no son especialistas en computación, el rostro del desarrollo de sistemas adquiere una nueva magnitud. Los propios usuarios emprenden ya el desarrollo de algunos de los sistemas que ellos emplean.

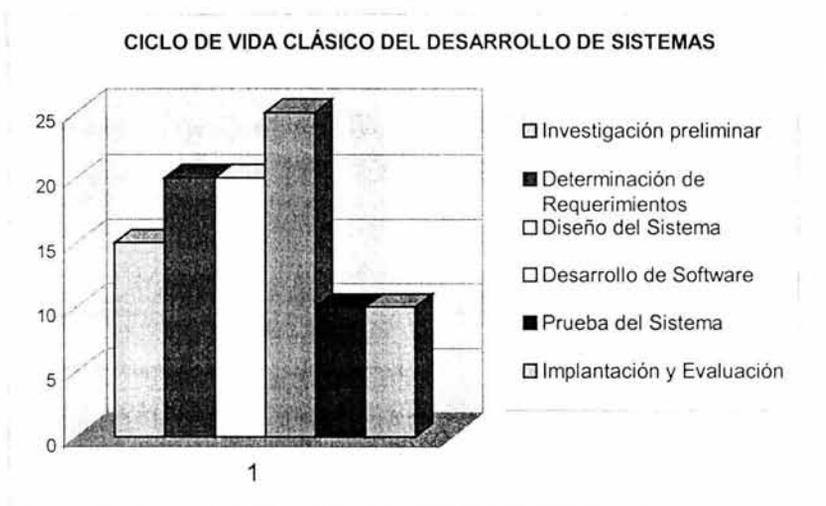
Todas estas situaciones están representadas por tres distintos enfoques al desarrollo de sistemas de información basados en computadora:

- Método del ciclo de vida para el desarrollo de sistemas.
- Método del desarrollo del análisis estructurado.
- Método del prototipo de sistemas.

### 2.5.1 CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS

El desarrollo de sistemas, un proceso formado por etapas del análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un sistema de la empresa que necesita mejoras.

El método del ciclo de vida para desarrollo de sistemas, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. El método del ciclo de vida para desarrollo de sistemas consta de las siguientes actividades:



### 2.5.1.1 INVESTIGACIÓN PRELIMINAR

La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones; sin importar cuáles sean éstas. El proceso se inicia siempre con la petición de una persona – administrador, empleado o especialista en sistemas.

Cuando se formula la solicitud comienza la primera actividad de sistemas: *la investigación preliminar*. Esta actividad tiene tres partes; aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

*Aclaración de la solicitud.* Muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara. Por consiguiente, antes de considerar cualquier investigación de sistemas, la solicitud del proyecto debe examinarse para determinar con precisión lo que el solicitante desea.

**Estudio de Factibilidad.** Un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible, en la investigación preliminar existen tres aspectos relacionados con el estudio de factibilidad.

- **Factibilidad técnica.** El trabajo para el proyecto, ¿Puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible? Si se necesita de nueva tecnología, ¿Cuál es la posibilidad de desarrollarla?
- **Factibilidad económica.** Al crear el sistema, ¿Los beneficios que se obtienen son suficientes para aceptar los costos?, ¿Los costos asociados con la decisión de NO crear el sistema son tan grandes que se debe aceptar el proyecto?
- **Factibilidad operacional.** Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿Existirá cierta resistencia al cambio por parte de los usuarios que dé como resultado una disminución de los posibles beneficios de la aplicación?

El estudio de factibilidad lo lleva a cabo un pequeño equipo de personas (en ocasiones una o dos) que está familiarizado con técnicas de sistemas de información; dicho equipo comprende la parte de la organización que se verá afectada por el proyecto, y es gente experta en los procesos de análisis y diseño de sistemas.

**Aprobación de la solicitud.** No todos los proyectos solicitados son deseables o factibles. Algunas organizaciones reciben tantas solicitudes de sus empleados que solo es posible atender unas cuantas. Sin embargo, aquellos proyectos que son deseables y factibles deben incorporarse en los planes.

#### 2.5.1.2 DETERMINACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. (Es por esta razón que el proceso de adquirir

información se denomina, con frecuencia, *investigación detallada*), los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

- ¿Qué es lo que se hace?
- ¿Cómo se hace?
- ¿Con qué frecuencia se presenta?
- ¿Qué tan grande es el volumen de transacciones o de decisiones?
- ¿Cuál es el grado de eficiencia con que se efectúan las tareas?
- ¿Existe algún problema?
- Si existe algún problema ¿Qué tan serio es?
- Si existe un problema. ¿Cuál es la causa que lo origina?

Para dar respuesta a todas estas preguntas es necesario conversar con varias personas para reunir detalles relacionados con los procesos de la empresa, sus opiniones, sobre porque ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso.

Asimismo la investigación detallada requieren del estudio de manuales y reportes, la observación en condiciones reales de las actividades del trabajo y, en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad.

Conforme se reúnen los detalles, los analistas estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles del procesamiento, tiempos de respuesta y métodos de entrada y salida.

### 2.5.1.3 DISEÑO DEL SISTEMA

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase del análisis. Los especialistas en

sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la de desarrollo de software, a la que denominan diseño físico.

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con gran precisión los datos específicos para cada reporte y salida. (Bosquejo del formato o pantalla que se espera).

El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Asimismo se escriben con todo detalle los procedimientos de cálculo y los datos individuales. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento, tales como discos y cintas magnéticas o incluso archivos en papel.

**Los procedimientos que se escriben indican como procesar los datos y producir las salidas.**

Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas y símbolos especiales). La información detallada del diseño se proporciona al equipo de programación para comenzar la fase del desarrollo del software.

Los diseñadores son los responsables de dar a los programadores las especificaciones de software completas y claramente delineadas. Una vez comenzada la fase de programación, los diseñadores contestan preguntas, aclaran dudas y manejan los problemas que enfrentan los programadores cuando utilizan las especificaciones de diseño.<sup>13</sup>

La etapa del Diseño del Sistema encierra cuatro etapas:

- **El diseño de los datos.** Transforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el Software.

---

<sup>13</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

- **El Diseño Arquitectónico.** Define la relación entre cada uno de los elementos estructurales del programa.
- **El Diseño de la Interfaz.** Describe como se comunica el Software consigo mismo, con los sistemas que operan junto con el y con los operadores y usuarios que lo emplean.
- **El Diseño de procedimientos.** Transforma elementos estructurales de la arquitectura del programa. La importancia del Diseño del Software se puede definir en una sola palabra **Calidad**, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El Diseño del Software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el Software.

El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

#### 2.5.1.3.1 CRITERIOS TÉCNICOS PARA EVALUAR LA CALIDAD DE UNA PRESENTACIÓN DEL DISEÑO

se deben establecer criterios técnicos para un buen diseño como son:

- Un diseño debe presentar una organización jerárquica que haga un uso inteligente del control entre los componentes del software.

- El diseño debe ser modular, es decir, se debe hacer una partición lógica del Software en elementos que realicen funciones y subfunciones específicas.
- Un diseño debe contener abstracciones de datos y procedimientos.
- Debe producir módulos que presenten características de funcionamiento independiente.
- Debe conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.
- Debe producir un diseño usando un método que pudiera repetirse según la información obtenida durante el análisis de requisitos de Software.

Estos criterios no se consiguen por casualidad. El proceso de Diseño del Software exige buena calidad a través de la aplicación de principios fundamentales de Diseño, Metodología sistemática y una revisión exhaustiva.

Cuando se va a diseñar un Sistema de Computadoras se debe tener presente que el proceso de un diseño incluye, concebir y planear algo en la mente, así como hacer un dibujo o modelo o croquis.

#### 2.5.1.3.2 DISEÑO DE LA SALIDA.

En este caso salida se refiere a los resultados e informaciones generadas por el Sistema, Para la mayoría de los usuarios la salida es la única razón para el desarrollo de un Sistema y la base de evaluación de su utilidad. Sin embargo cuando se realiza un sistema, como analistas deben realizar lo siguiente:

- Determine que información presentar. Decidir si la información será presentada en forma visual, verbal o impresora y seleccionar el medio de salida.
- Disponga la presentación de la información en un formato aceptable.
- Decida como distribuir la salida entre los posibles destinatarios.

### 2.5.1.3.3 DISEÑO DE ARCHIVOS.

Incluye decisiones con respecto a la naturaleza y contenido del propio archivo, como si se fuera a emplear para guardar detalles de las transacciones, datos históricos, o información de referencia. Entre las decisiones que se toman durante el diseño de archivos, se encuentran las siguientes:

- Los datos que deben incluirse en el formato de registros contenidos en el archivo.
- La longitud de cada registro, con base en las características de los datos que contenga.
- La secuencia a disposición de los registros dentro del archivo (La estructura de almacenamiento que puede ser secuencial, indexada o relativa).

No todos los sistemas requieren del diseño de todos los archivos, ya que la mayoría de ellos pueden utilizar los del viejo Sistema y solo tenga que enlazarse el nuevo Sistema al Archivo maestro donde se encuentran los registros.

### 2.5.1.3.4 DISEÑO DE INTERACCIONES CON LA BASE DE DATOS.

La mayoría de los sistemas de información ya sean implantado en sistemas de cómputos grandes o pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas utilizan u administrador de base de datos, en este caso el diseñador no construye la base de datos sino que consulta a su administrador para ponerse de acuerdo en el uso de esta en el sistema.

### 2.5.1.4 DESARROLLO DE SOFTWARE

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante.

La elección depende en gran medida del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores.

Los programadores también son los responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma.

La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

#### **2.5.1.5 PRUEBA DE SISTEMAS**

Durante la fase de pruebas del sistema, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga.

Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

En muchas organizaciones, las pruebas son conducidas por personas ajenas al grupo que escribió los programas originales: con esto se persigue asegurar, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable.

#### **2.5.1.6 ESTRATEGIAS DE PRUEBAS DEL SOFTWARE <sup>15</sup>**

La prueba es un proceso individualista y el número de tipos diferentes de pruebas varía tanto como los

diferentes enfoques de desarrollo, una prueba es un conjunto de actividades que se puede planificar por adelantado y llevar a cabo sistemáticamente.<sup>15</sup>

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en un conjunto de pasos bien planeados que dan como resultado la correcta construcción del software.

Una estrategia de prueba de software proporciona una guía o plano para los desarrolladores del software (sistema), para la organización de control de calidad y para el cliente -un plano que describe los pasos a llevar a cabo como parte de la prueba, cuando se deben planificar y llevar a cabo su realización, y cuánto tiempo, esfuerzo y recursos se van a utilizar.

Por tanto una estrategia de software debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de pruebas y la agrupación y evaluación de los datos resultantes.

#### 2.5.1.6.1 PRUEBA DE UNIDAD

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software. Aquí se prueban los caminos de control importantes, con el fin de descubrir errores dentro del ámbito de un módulo.

La complejidad relativa de las pruebas y errores descubiertos se encuentra limitada por los lineamientos establecidos por la prueba de software.

Se prueba la Interfaz del módulo para asegurar que la información fluye en forma adecuada hacia y desde la unidad del programa que está siendo probada. Se analizan las estructuras de datos para asegurar que los datos mantienen su integridad temporal durante todos los pasos de ejecución del algoritmo.

---

<sup>15</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

Se prueba las condiciones límite para asegurar que el módulo funciona correctamente dentro de los límites establecidos por el procesamiento.

Se activan los caminos básicos de la estructura de control con el fin de asegurar que las sentencias del módulo se ejecutan por lo menos una sola vez, y finalmente, se prueban todos los caminos de manejo de errores.

Antes de iniciar cualquier otra prueba es preciso probar el flujo de los datos del módulo, ya que si los datos no entran correctamente, todas las demás pruebas no tienen sentido. Myers, propone una lista de comprobaciones para la prueba de interfaces.

- ¿Es igual el número de parámetros de entrada al número de argumentos?
  - ¿Coinciden las características (atributos) de los parámetros con los argumentos?
- ¿Coinciden el sistema de unidades de los parámetros con el de los argumentos?
- ¿Son correctos el número, atributos y el orden de los argumentos de las funciones incorporadas?
- ¿Existen referencias o parámetros que no estén asociados con el punto de entrada actual?
- ¿Entran sólo argumentos alterados?
- ¿Son consistentes las definiciones de variables globales entre módulos?
- ¿Se pasan las restricciones como argumentos?
- Cuando un módulo tenga operaciones básicas de Entrada/Salida externa, se deben llevar a cabo pruebas adicionales :
  - ¿Son correctos los atributos de los archivos?
  - ¿Son correctas las sentencias de apertura?
  - ¿Coinciden las especificaciones de formato con las sentencias de Entrada/Salida?
  - ¿Coincide el tamaño del buffer con el tamaño del registro?
  - ¿Se abren los archivos antes de usarlos?
  - ¿Se tienen en cuenta las condiciones de fin de archivo?
  - ¿Se manejan errores de Entrada/Salida?
  - ¿Hay algún error textual en la información de salida?

Las estructuras de datos locales de cada módulo son una gran fuente de errores. Se deben diseñar casos de prueba para descubrir errores:

- Tipificación impropia o inconsistente
- Inicialización o valores implícitos erróneos
- Nombres de variables incorrectos (mal escritos o truncados)
- Tipos de datos inconsistentes
- Desbordamiento o errores en el direccionamiento de memoria

Se deben diseñar casos de prueba para detectar errores causados por cálculos incorrectos o flujos de control inapropiados. Las pruebas de camino básico o de ciclos son técnicas más efectivas para descubrir una gran cantidad de errores en los caminos. Entre los errores más comunes en los cálculos están:

- Procedencia aritmética incorrecta mal aplicada
- Operaciones de modo mixto
- Inicializaciones incorrectas
- Falta de precisión
- Representación incorrecta de una expresión
- Las comparaciones y el flujo de control están muy relacionados, es decir, el flujo de control cambia tras una comparación

Los casos de prueba deben descubrir errores como:

- Comparaciones entre tipos de datos distintos
- Operadores lógicos o de procedencia incorrecta
- Terminación de ciclos inapropiada o inexistente
- Falta de salida cuando se encuentra una iteración mal aplicada (Ciclos infinitos)
- Variables internas a un ciclo modificadas en forma inadecuada.

Entre los errores que se deben comprobar se evalúa en la manipulación de errores lo siguiente:

- La condición de error hace que intervenga el sistema antes que el mecanismo de errores
- Descripción ilegible del error
- El error señalado no corresponde con el error encontrado
- La descripción del error no proporciona suficiente información para ayudar en la localización de la causa del error.

La prueba de los límites es la última etapa de la prueba de unidad y quizá la más importante. El software falla en sus condiciones límite, o sea, que frecuentemente aparece un error cuando se procesa el elemento  $n$ -ésimo de un arreglo  $n$ -dimensional, cuando se hace la  $i$ -ésima repetición de un ciclo de  $x$  pasos o cuando se llega a los valores máximo ó mínimo permitidos.

Los casos de prueba que ejerciten las estructuras de datos por debajo o encima de los mínimos y máximos permitidos son apropiados para descubrir estos errores.

Normalmente, se considera a la prueba de unidad como algo adyacente a al paso de la codificación. En donde los casos de prueba de unidad comienzan una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente.

#### 2.5.1.6.2 PRUEBA DE INTEGRACIÓN

Una vez que los módulos funcionen por separado y hayan pasado la prueba de unidad es necesario cuestionar lo siguiente: "El módulo funciona bien sólo", ¿Por qué dudar que funcionen juntos?. Aquí pueden surgir los problemas en la unión de los módulos.

Los datos se pueden perder en una interfaz, un módulo puede tener un efecto adverso sobre otro módulo; las subfunciones, cuando se combinan, pueden no producir el objetivo principal deseado, las estructuras de datos globales pueden presentar problemas.

La prueba de Integración es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y estructurar un programa que esté de acuerdo con lo que dicta el diseño.

Existen 2 tipos de integración, la primera es **no incremental** en donde se intenta elaborar software en módulos grandes, en otros casos un sólo módulo, pero en ellos es más difícil aislar los errores y cuando alguno de ellos es corregido produce otros errores.

El segundo tipo de integración es **incremental** en donde se desarrollan módulos pequeños y funcionales que hacen que los errores sean más fácil de aislar y corregir, es más probable que se puedan probar completamente las interfaces y aplicar un enfoque de prueba sistemático.

Integración descendente, es una estrategia de integración incremental a la construcción de la estructura de programas, en cual se integran los módulos moviéndose en dirección hacia abajo por la jerarquía de control comenzando con el módulo principal (Programa principal).

#### **El proceso de integración se lleva a cabo en 5 pasos:**

Se usa el módulo de control principal como conductor de prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.

Dependiendo del enfoque de integración elegido (primero-en profundidad o primero-en-anchura) se van sustituyendo los resguardos subordinados cada uno por los módulos reales.

Las pruebas se llevan a cabo cada vez que se integra un módulo.

Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real.

Se hace una prueba de regresión, es decir, todas las pruebas anteriores para asegurar que no se han introducido errores.

La estrategia descendente puede ocasionar algunos problemas. Uno de ellos puede ser cuando se requiere un procesamiento de los niveles más bajos de la jerarquía para probar los niveles superiores. El encargado de la prueba tiene 3 opciones:

- Retrasar muchas de las pruebas hasta que los resguardos sean reemplazados por módulos reales.
- Desarrollar resguardos que realicen funciones limitadas que simulen los módulos reales.
- Integrar el software desde el fondo de la jerarquía hacia arriba.

**Integración Ascendente**, es en donde la construcción del diseño empieza de los módulos más bajos hacia arriba (Módulo principal), el procesamiento requerido de los módulos subordinados siempre está disponible y elimina la necesidad de resguardos. Una estrategia de integración ascendente puede ser implementada mediante los siguientes pasos:

- Se combinan los módulos de bajo nivel en grupos que realicen una subfunción en el software.
- Se escribe un programa de control de prueba para coordinar la Entrada y Salida de los casos de prueba
- Se prueba el grupo
- Se eliminan los programas de control y se combinan los grupos moviéndose hacia arriba por la estructura del programa

### 2.5.1.6.3 PRUEBA DE VALIDACIÓN Y VERIFICACIÓN

Al conjunto de actividades que aseguran que el software implementa correctamente una función específica se denomina Verificación. La Validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos y necesidades del cliente. Boehm lo establece de otra forma:

Verificación : "¿estamos construyendo el software correctamente?"

Validación : "¿estamos construyendo el software correcto?"

La definición de Verificación y Validación envuelve lo que se conoce como calidad del software. Los métodos de análisis, de diseño y de implementación (codificación) actúan para mejorar la calidad al proporcionar técnicas continuas y resultados predecibles. Las revisiones técnicas formales ayudan (inspección) ayudan a asegurar la calidad de los productos, a lo largo del proceso la medición y el control se aplican sobre cada elemento de una configuración del software.

La prueba constituye un elemento importante desde el que se puede evaluar la calidad y, de forma más práctica, descubrir los errores. Cabe mencionar que la prueba no se debe contemplar como una red de seguridad.

La aplicación adecuada de los métodos y de las herramientas, las revisiones técnicas formales efectivas y una sólida gestión y medida, conducen a la calidad, que se confirma durante la prueba.

Miller relaciona la prueba del software con la garantía de calidad al establecerse que "la motivación subyacente de la prueba de programas es confirmar la calidad del software con métodos que se puedan aplicar de forma económica y efectiva, tanto en grandes como pequeños sistemas".<sup>1</sup>

Es importante mencionar que la validación y verificación abarcan un amplio rango de la calidad del software que incluyen revisiones técnicas formales, auditorías de configuración y calidad, supervisión de

rendimiento, simulación, estudio de viabilidad, revisión de la documentación, revisión de la base de datos, análisis de algoritmos, pruebas de desarrollo, prueba de calificación y prueba de instalación.

Una vez que se culminó la etapa de integración se puede decir que el software está completamente ensamblado, se han encontrado y corregido errores de la interfaz y se puede comenzar una serie final de pruebas del software. La prueba de validación se logra cuando las expectativas razonables del cliente se cumplen en donde incluyen la especificación de requisitos, documentos en donde se describen los atributos del software que son visibles para el usuario, esta información forma la base del enfoque a la prueba de validación.

El procedimiento de prueba estará diseñado para asegurar que se satisfacen los requisitos funcionales, que se alcanzan todos los requisitos de rendimiento, que la documentación es correcta y que se alcanzan otros requisitos tales como portabilidad, compatibilidad, recuperación de errores, facilidad de mantenimiento, etc...

Una vez que se procede con la prueba de validación, puede darse una de las dos condiciones:

- Las características de funcionamiento o rendimiento están de acuerdo con las especificaciones y son aceptables ó
- Se encuentra una desviación de las especificaciones y se crea una lista de deficiencias.

Cuando se construye el software para llevar a cabo la prueba de validación es casi imposible que el desarrollador pueda prever como un cliente usará realmente el programa es por ello que se hace una serie de pruebas de aceptación que puede permitir que un cliente valide todos los requisitos, se puede dar el caso de las pruebas alfa y beta.

**La prueba alfa consiste** en una prueba del software ejecutado por el cliente estando presente el desarrollador para hacer las anotaciones necesarias cuando los errores o las observaciones del cliente sucedan.

**Las pruebas beta** son versiones del software que los desarrolladores lanzan antes de la versión final, esto es que se realicen las pruebas si la presencia del desarrollador ni el equipo de desarrollo para efectos de compatibilidad, portabilidad, mantenimiento, etc. , Así la prueba beta es una aplicación en vivo del software en un entorno diferente.

En las técnicas anteriores se mencionan algunas técnicas de prueba que se efectúan durante el diseño del software, cabe mencionar que en cada una de esas técnicas serán el éxito o fracaso para la integración del software en el sistema total. Un clásico problema de la prueba del sistema es la delegación de culpabilidad. Esto ocurre cuando se descubre un error y cada uno de los creadores de cada elemento del sistema echa la culpa del problema a los otros, para evitar esto se debe prever y tomar medidas correctivas tomando en cuenta lo siguiente:

- Diseñar caminos de manejo de errores que prueben toda la información procedente de otros elementos del sistema.
- Llevar a cabo una serie de pruebas que simulen la presencia de datos en mal estado o de otros posibles errores en la interfaz del software.
- Registrar los resultados de las pruebas como "evidencia" en el caso de señalamientos informales.
- Participar en la planificación y diseño de las pruebas del sistema para asegurarse de que el software es probado en forma adecuada.

La prueba del sistema se basa en otras técnicas de prueba que hacen ejercitar profundamente el sistema basado en computadora, aunque la finalidad de cada prueba es distinta, sirven para verificar que se hayan integrado correctamente cada uno de los elementos del sistema:

#### 2.5.1.6.4 PRUEBA DE RECUPERACIÓN.

Muchos sistemas basados en computadora deben recuperarse de los fallos y resumir el procesamiento en tiempos previamente especificados. La prueba de recuperación es una prueba que se hace al sistema forzando a que produzca fallas de software de muchas maneras y verificando que la

recuperación se lleve a cabo, ya sea automáticamente o manual, tomando en cuenta los recursos que se requieran para efectuar la recuperación.

#### 2.5.1.6.5 PRUEBA DE SEGURIDAD.

Cualquier sistema basado en la computadora que manipule información sensible o efectúe acciones que pueden perjudicar o beneficiar a los individuos es objeto de penetraciones impropias o ilegales. La penetración incluye una serie de actividades como:

- Penetrar sistemas por joby
- personas disgustadas que intentan penetrar por venganza
- personas deshonestas que intentan penetrar para obtener ganancias personales ilícitas
- usuarios que intentan penetrar para solucionar problemas, desconociendo las consecuencias que esto pueda traer a la integridad del sistema.

La prueba de seguridad intenta verificar la aplicación de los mecanismos de protección incorporados en el sistema.

Durante la prueba el encargado de la prueba desempeña el papel de intruso tratando de violar la seguridad del sistema, intentando obtener las claves de acceso por cualquier medio externo; debe bloquear el sistema, negando así el servicio a otras personas además de producir errores a propósito en el sistema; o debe curiosear los datos públicos intentando encontrar una clave de acceso al sistema.

#### 2.5.1.6.6 PRUEBA DE RESISTENCIA.

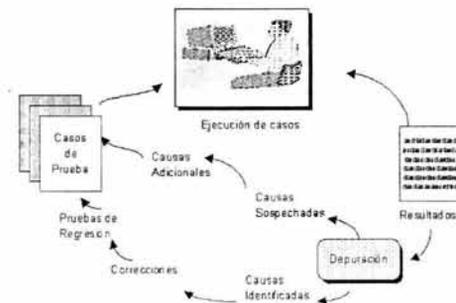
La prueba de resistencia está diseñada para enfrentar a los programas en situaciones anormales, es decir ejecutar el sistema en forma que demande recursos en cantidad, frecuencia o volúmenes

anormales. El encargado de la prueba debe intentar tirar el sistema. Para lograr esto se puede tomar en consideración lo siguiente:

- Diseñar pruebas especiales que generen 10 o más interrupciones por segundo.
- Incrementar las frecuencias de datos de entrada en un orden de magnitud con el fin de comprobar como responden las funciones de entrada.
- Ejecutar casos de prueba que requieran al máximo de memoria o de espacio en disco.
- Diseñar casos de prueba que produzcan excesivas búsquedas de datos almacenados en disco.

### 2.5.1.6.7 DEPURACIÓN

La depuración aparece como resultado de una prueba efectiva, es decir, cuando en un caso de prueba se encuentra un error, la depuración es el proceso que resulta en la eliminación de un error. Se debe tomar en cuenta que la depuración no es una técnica de prueba, aunque siempre se da como consecuencia de la prueba. De acuerdo con la figura el proceso de depuración comienza en los casos de prueba, se evalúan los resultados y resulta una falta de correspondencia entre los esperados y los reales, el proceso de depuración intenta hacer corresponder el sistema con una causa, llevando así a la corrección del error.



El proceso de depuración siempre tiene uno de los dos resultados:

- Se encuentra el error, se corrige y elimina
- no se encuentra la causa del error

En el último caso es cuando la persona encargada de la depuración debe sospechar la causa, en donde debe diseñar casos de prueba que ayuden a confirmar sus sospechas y el trabajo se regresa a la corrección del error de una forma interactiva.

La depuración tiene como objetivo principal encontrar y corregir la causa de un error en el software, existen 3 enfoques que se pueden proponer en la depuración.

- Eliminación de causas
- Fuerza Bruta
- Vuelta atrás

La categoría de depuración "eliminación de causas" se manifiesta mediante inducción o deducción. Los datos relacionados con la ocurrencia del error se organizan para llegar a las posibles causas; se desarrolla una lista de las causas y se llevan a cabo las pruebas para eliminar cada una.

La categoría de depuración por "fuerza bruta" es la categoría probablemente la más común y la menos eficiente en el momento de aislar la causa del error del software en donde se confía que en algún lugar de la gran cantidad de información producida nos puede llevar finalmente al éxito, lo más frecuente es que se desperdicie tiempo y esfuerzo.

La vuelta atrás es el enfoque más normal para la depuración, que se puede usar con gran éxito en programas pequeños. Partiendo de donde se detecta el error hacia atrás en el código fuente hasta llegar a la posición del error.

Cada uno de los enfoques anteriores puede complementarse con herramientas de depuración, ayudas dinámicas de depuración, generadores automáticos de casos de prueba, etc.

En general a partir de una indicación de un problema, la actividad de depuración debe rastrear la causa del error.

### 2.5.1.7 IMPLANTACIÓN Y EVALUACIÓN <sup>16</sup>

La implantación es el proceso de verificar e instalar el nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarlas. <sup>16</sup>

Dependiendo del tamaño de la organización se empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa (prueba piloto), por ejemplo en un departamento con una o dos personas, algunas veces se deja que los dos sistemas funcionen en forma paralela con la finalidad de comparar los resultados.

En otras ocasiones se deja de utilizar el viejo sistema determinado día para comenzar a utilizar el nuevo al día siguiente.

Cada estrategia de implantación tiene sus propios méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cuál sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema este libre de problemas.

Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones; realizando cambio y modificaciones en el software, archivos o procedimientos para satisfacer las necesidades de los usuarios.

Dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios de manera continua, los sistemas de información deben mantenerse siempre al día. En este sentido, la implantación es un proceso de constante evolución.

---

<sup>16</sup> Análisis y Diseño de sistemas de Información

La evaluación de un sistema se lleva a cabo para verificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- Evaluación Operacional. Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.
- Impacto Organizacional. Identificación y medición de los beneficios para la organización en áreas tales como finanzas (costos, ingresos, ganancias), eficiencia operacional e impacto competitivo, también se incluye el impacto sobre el flujo de información Interno y Externo.
- Opinión de los Administradores. Evaluación de las actitudes de directivos y administradores dentro de la organización así como de los usuarios finales.
- Desempeño del Desarrollo. La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo.

Desgraciadamente la evaluación de sistemas no siempre recibe la atención que merece. Sin embargo cuando se conduce de forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.

#### **2.5.1.7.1 OTROS MÉTODOS PARA LA IMPLANTACIÓN DE UN SISTEMA 17**

Existen cuatro métodos para efectuar la implantación de un sistema, en donde cada método tiene ventajas y desventajas entre los mismos. Es conveniente mencionar que algunos métodos de implantación se ajustan perfectamente a problemas planteados, en donde nos es conveniente aplicar

otro método debido a que los posibles resultados sean ampliar los periodos de implantación, situación que puede crear problemas con los usuarios y los analistas.

**Sistemas paralelos.** Este método es un modelo seguro, ya que el sistema nuevo funciona al mismo tiempo que el sistema anterior; debido a que si encuentran fallas en el sistema nuevo, el anterior permanece en constante funcionamiento sin pérdidas de tiempo, ni de información.

Una desventaja muy notable es que este método de conversión es muy costoso ya que se duplican los recursos humanos como materiales para trabajar en forma paralela ambos sistemas duplicando así los costos de operación.

**Conversión directa.** El sistema anterior será reemplazado por el nuevo sistema, ya que la organización confía plenamente en el nuevo sistema. Obligando así a los usuarios a que hagan funcionar al nuevo sistema encontrando en él nuevos métodos y controles. En caso de encontrar errores no existe un sistema que lleve un respaldo de seguridad en las operaciones, este método ocasiona una planeación más cuidadosa.

**Enfoque piloto.** En este método se implanta el nuevo sistema en una parte de la organización. Con base a retroalimentación se hacen cambios y el sistema se instala en el resto de la organización mediante algunos otros métodos, esto en gran medida proporciona experiencia y prueba directa antes de la implantación. Ocasionando la desventaja de que el nuevo sistema puede dar la impresión de que el nuevo sistema no es confiable, ni está libre de errores.

**Conversión por etapas.** Se implanta el nuevo sistema de manera gradual, reemplazando partes del sistema anterior, permitiendo a los primeros usuarios aprovechar las ventajas del nuevo sistema, así como la capacitación de los mismos, llevando a cabo la instalación de una nueva etapa una vez que fue aceptada sin hacer uso necesario de recursos extras.

**Plan De Conversión.** El plan de conversión incluye una lista de actividades para llevar a cabo la implantación del nuevo sistema y ponerlo en operación. En él se identifica a las personas responsables

de cada actividad, así como un programa de actividades para dar seguimiento a cada una de estas etapas.

El plan de conversión debe prevenir los posibles problemas y la forma de enfrentarlos implementando planes de emergencia adecuados. Algunos problemas que se presentan comúnmente son:

- Documentos perdidos
- Variación entre datos de los formatos nuevos y anteriores
- Errores en la conversión de datos
- Extravío de datos o pérdida de archivos
- Omisión en algunos pasos de la conversión

Durante la conversión se deben considerar diversos aspectos, desde la instalación del equipo hasta el orden de las formas y los suministros. Algunas actividades de conversión comienzan realmente desde que un sistema se está desarrollando, así como contactar proveedores.

**Acondicionamiento De Las Instalaciones.** Frecuentemente los analistas trabajan con el personal de los proveedores para definir el acondicionamiento de las instalaciones. El cliente o el ingeniero de sistemas presentará una lista de especificaciones para el cableado eléctrico y los contactos, necesidades de aire acondicionado, controles de humedad y exigencias de espacio, lo más recomendable es tener el local antes de la llegada del equipo.

**Capacitación De Operadores Y Usuarios Para Operar El Sistema.** En algunos casos los sistemas de las compañías de más prestigio pueden tener éxito o fracasar debido a la forma en que se operan y se usan. Como consecuencia la calidad de capacitación recibida por el personal manipulará el sistema ayuda u obstruye, y puede llegar a impedir, la implantación exitosa de un sistema de información.

Aquellas personas que se encuentren asociadas o afectadas por el sistema deben conocer cuál es su papel, cómo pueden usar el sistema y qué hará o no hará el sistema. Tanto operadores como los usuarios del sistema requieren de capacitación.

**Capacitación de operadores.** La mayoría de los sistemas dependen del personal de Centro de Cómputo, el cual es el encargado de mantener los equipos de cómputo funcionando, así como de proporcionar el apoyo necesario. La capacitación que el personal reciba debe de asegurar que puedan manejar todas las operaciones posibles, tanto rutinarias como extraordinarias. La capacitación que reciba el operador también debe contemplar al personal de captura de datos.

Si el sistema requiere de la instalación de equipo nuevo, el personal debe ser capacitado en lo necesario para encender y apagar el equipo, así como conocimiento de lo que es su operación y su uso normal. Como parte de la capacitación se le debe dar al personal una lista de formas de resolver los problemas y que identifique los posibles problemas y solución, así como datos personales de las personas a quién buscar cuando surjan problemas inesperados.

**Capacitación de usuarios.** La capacitación de usuarios incluye la identificación de problemas, determinando si el problema que surge es ocasionado por hardware o por software, o por algo realizado por los mismos usuarios que ocasione la falla del sistema.

No hay nada más desesperante que trabajar con un sistema y que suceda un problema y no ser capaz de determinar si es falla del propio sistema, del equipo o de los usuarios. El lugar para prevenir estos casos es durante la capacitación. La mayor parte de la capacitación de los usuarios tiene que ver con la operación del sistema en sí.

La capacitación en la codificación de los datos marca la pauta en el proceso de captura a partir de las transacciones, o en la preparación de datos necesarios para las actividades de apoyo a las decisiones.

Las actividades de manipulación de los datos que recibe la mayor atención en la capacitación de usuarios son la captura de datos (Cómo modificar datos previamente almacenados), la formulación de consultas (Cómo localizar información específica u obtener respuestas a preguntas) y el borrados de los

registros de datos. La parte fundamental de la capacitación cae sobre esta actividad dedicándose la mayor parte del tiempo a esta área.

En algunos casos los usuarios tienen que tener conocimiento de la operación básica de algunos periféricos tales como una impresora, en donde deben tener el conocimiento de como colocar el papel ó cambiar la cinta de la impresora, así como otros procesos rutinarios como mantener limpio el equipo, y además del conocimiento para la manipulación de los discos como es formatear y probar discos.

En la capacitación de usuarios se debe tomar en cuenta dos aspectos importantes mencionados anteriormente los cuales consisten en la familiarización con el sistema de procesamiento en sí (es decir, el equipo usado para la captura y procesamiento de datos) y la capacitación en el uso de la aplicación (es decir, el software que acepta los datos, los procesa y produce los resultados). La debilidad de cualquier aspecto de la capacitación puede ser una posibilidad para la generación de errores. Una buena documentación, aunque es importante, no reemplaza la capacitación.

**Métodos de capacitación.** La capacitación a operadores y usuarios se puede manifestar de diferentes formas. Las actividades de capacitación se pueden llevar a cabo en las instalaciones de los proveedores, locales rentados como podría ser el caso de hoteles y Universidades, "en casa", o en las instalaciones de la empresa. Los métodos y contenido de la capacitación varía de acuerdo a la fuente y del lugar de la capacitación

**Instalación Y Pruebas Del Usuario Para Aceptación Del Sistema.** En la instalación y prueba del usuario, son algunas actividades que realiza el usuario con el fin de verificar el sistema si funciona con el hardware diferente al hardware de desarrollo, así como su instalación y funcionamiento en general, en esta etapa se abarcan cuatro aspectos importantes:

La prueba de entrada que implica ver si las formas electrónicas y las de papel así como la estructura de codificación cumplen con las guías y especificaciones de diseño, también se hacen una auto prueba los usuarios finales del sistema para saber si están llenando correctamente las formas. Al efectuar estas pruebas se supone que el usuario ya recibió capacitación y en ella se les enseñó como llenar los formatos. Las pruebas en esta etapa consisten en determinar si fueron capacitados correctamente.

Muchas organizaciones contratan personas cuya única función es capturar datos siendo de gran importancia una capacitación adecuada para la captura de datos.

La prueba de salida, en la mayoría de estas pruebas son un subproducto de la prueba de otros componentes estructurales. Por ejemplo, mientras se prueban la entrada y las bases de datos, se revisa la utilidad y exactitud de la salida resultante.

Una prueba de salida no es otra cosa que la generación de un reporte, su entrega al usuario y determinar si satisface las necesidades de información. Una buena forma de llevar a cabo la prueba de salida consiste en entregar un reporte a una persona no familiarizada con el sistema, si esta persona puede explicar el reporte, entonces probablemente el formato puede ser entendido por los usuarios.

**Prueba de la base de datos.** Las pruebas importantes para determinar si las bases de datos satisfacen las necesidades de los usuarios, en gran medida, es la prueba de salida. Sin embargo se pueden llevar a cabo pruebas adicionales para asegurarse que las bases de datos contengan la información que satisfaga todas las demandas que se le plantean. Incluyendo pruebas tales como la creación de un nuevo registro antes del primer registro en el archivo maestro, la creación de un nuevo registro después del último, la creación de un registro de un departamento que no existe, intentar leer o escribir en un archivo con el disco fuera de la unidad (en el caso de sistemas en unidades flexibles).

**Prueba de los controles.** El propósito primordial de la prueba de controles es asegurar que estos se encuentren en su lugar y trabajen según se planeó. Las transacciones de prueba ayudan a asegurar que los controles de procesamiento, como las verificaciones de límite y racionalidad, la prueba aritmética y la identificación están en su lugar funcionando correctamente. Algunas pruebas de control que normalmente se realizan son:

- Tratar de leer o escribir en un archivo incorrecto
- Introducir varios campos con datos incompletos o faltantes
- Tratar de procesar una transacción delicada sin la autorización apropiada y ver si el sistema lo rechaza

- Hacer verificaciones de caracteres numéricos, alfabéticos y especiales, por ejemplo en la captura de una clave se deberán introducir únicamente valores numéricos, en donde se deben introducir valores alfabéticos. Una verificación que funcione correctamente detectará el error antes de realizar el procesamiento.
- Hacer verificaciones de límite y racionalidad, por ejemplo : Un docente no puede tener más de 40 horas en una jornada al día, se probará introduciendo valores que excedan la jornada de un día
- Realizar verificaciones de validez de campos clave, por ejemplo : introducir una clave inválida y ver como responde el sistema
- Introducir en un campo numérico valores negativos y ver si lo acepta como tal, o los efectos que ocasiona.

Las pruebas de los controles incluyen también una revisión de la documentación que aparece en los reportes del desarrollo de sistemas. Después de que el usuario considera que el software paso las pruebas, deberá llevarse a cabo una reunión de aceptación, a la que asista el gerente de operaciones de sistemas, el analista de sistemas y el personal usuario.

En este momento se le da terminación oficial del proyecto de desarrollo obteniendo así la "clausura" final de sistemas, es entonces cuando el equipo de desarrollo queda libre para otra tarea o para diseñar o implementar algunas de las solicitudes y sugerencias de mejora del sistema que fueron suspendidas anteriormente.

#### 2.5.1.8 MANTENIMIENTO DEL SOFTWARE <sup>18</sup>

El mantenimiento de software, es mucho más que una corrección de errores; el mantenimiento se puede describir en tres actividades. <sup>18</sup>

- Mantenimiento correctivo
- Mantenimiento adaptativo

---

<sup>18</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

➤ Mantenimiento perfectivo

**Mantenimiento correctivo**, esta actividad del mantenimiento es debido a que no es razonable que en la prueba de software se haya descubierto los errores de un gran sistema de software. Durante el uso de cualquier programa se encuentran errores y estos son informados al personal de desarrollo. Este proceso incluye el diagnóstico y corrección de uno o más errores.

La evolución rápida tanto de hardware como de software genera cambios en periféricos, sistemas operativos o nuevas versiones de los anteriores en otros casos nuevas disposiciones nacionales (por ejemplo en nuestro país el cambio de moneda) hacen que algunos sistemas no se adopten a la nueva tecnología o las nuevas disposiciones, por lo tanto una actividad que modifica al software para que interaccione adecuadamente con su entorno cambiante, es la del **mantenimiento adaptativo**.

La tercera actividad del mantenimiento se da cuando existe software que tiene un gran éxito. A medida que se utiliza el software algunos usuarios hacen observaciones sobre recomendaciones para nuevas posibilidades ó sobre modificaciones a las funciones ya existentes ó sobre mejoras en general. Para satisfacer estas peticiones se lleva a cabo el mantenimiento perfectivo siendo una actividad que genera un mayor esfuerzo empleado en el **mantenimiento de software**.

**Características del mantenimiento.** Las actividades requeridas para cubrir la fase de mantenimiento abarcan dos aspectos importantes: **Mantenimiento estructurado y mantenimiento no estructurado**.

El mantenimiento no estructurado, es cuando sólo se dispone del código fuente como un elemento de configuración empezándolo a evaluar a menudo complicada por la pobre documentación interna. Las delicadas características, tales como las estructuras de datos, variables globales, las interfaces del sistema, el rendimiento y/o limitaciones del diseño, son difíciles de descubrir y frecuentemente mal interpretados.

Si existe una completa configuración del software, la tarea del mantenimiento comienza con la evaluación de la documentación del diseño, se determinan las importantes características estructurales, de rendimiento y de interfaz del software.

Se estudian las consecuencias de las correcciones o modificaciones requeridas y se traza un plan de acciones, se modifica el diseño y revisa. Se desarrolla el nuevo código fuente sometiéndolo a las pruebas del software haciendo las respectivas correcciones y se vuelve a lanzar el software, a este suceso de actividades corresponde al mantenimiento estructurado.

Con respecto al costo del mantenimiento del software algunas veces genera insatisfacción al cliente cuando una petición o modificación aparentemente legítima no se puede atender en un tiempo razonable.

Disminución de la calidad global del software debido a los errores permanentes que se generan cuando se introducen cambios al software en mantenimiento, utilización de muchos recursos humanos del área de mantenimiento para efectuar el mantenimiento.

La falta de control y disciplina en las actividades de desarrollo, casi siempre se traduce en problemas para el mantenimiento del software como los siguientes:

- Frecuentemente es difícil seguir la evolución del software a través de varias versiones ya que los cambios no están adecuadamente documentados.
- Frecuentemente es imposible o difícil seguir el proceso por el que se construyó el software.
- Generalmente es completamente difícil comprender un programa "ajeno".
- Ese personaje "ajeno", por lo regular, no se encuentra cerca para que pueda explicar lo que hizo.
- No existe una documentación apropiada o está mal preparada.
- La mayoría del software no ha sido diseñado previendo el cambio.

Estos problemas que se acaban de mencionar, en parte, se pueden atribuir al gran número de programas actualmente existentes que han sido desarrollados sin tener en cuenta una metodología de desarrollo.

#### 2.5.1.8.1 EFECTOS SECUNDARIOS DEL MANTENIMIENTO

La modificación al software en los códigos fuente es peligrosa. Algunas veces hemos escuchado.. "Pero si todo lo que realicé fue cambiarle una sentencia", lamentablemente cada vez que se introduce un cambio en un proceso lógico muy complejo, la posibilidad del error aumenta.

FREEDMAN y WEINBERG definen tres grandes categorías de efectos secundarios del mantenimiento:

**Efectos secundarios sobre el código.** Con el hecho de realizar un sencillo cambio sobre una sola sentencia puede a veces tener resultados desastrosos.

El cambio inadvertido de algún símbolo "." ó ";" Muchas veces pueden acarrear problemas trágicos. En una computadora nos comunicamos mediante el código fuente en algún lenguaje de programación.

Las posibilidades de efectos secundarios abundan, aunque cada modificación que se realice en el código puede regularmente inducir al error. Los siguientes cambios tienen mayor probabilidad de inducir a error que otros:

- Un subprograma eliminado o cambiado
- Eliminación ó modificación de una sentencia o etiqueta
- Eliminación o modificación de un identificador
- Cambios para mejorar el rendimiento en ejecución
- Modificación de apertura o cierre de archivos
- Modificación de operadores lógicos
- Cambios sobre las pruebas límite

**Efectos secundarios sobre las bases de datos.** Durante el mantenimiento frecuentemente se hacen cambios sobre determinados elementos de una estructura de datos o sobre la propia estructura de datos. Los efectos secundarios sobre las bases de datos regularmente suceden por las modificaciones sobre la estructura de la información del software. Los siguientes cambios en los datos producen efectos secundarios como:

- **Redefinición de constantes locales o globales**
- Redefinición de registros o archivos
- Aumento o disminución del tamaño de los arreglos o de las estructuras de datos de mayor orden
- Modificación de datos globales
- Reiniciación de indicadores de control o de punteros
- Reorganización de argumentos de Entrada/Salida o de subprogramas

Los efectos secundarios se pueden limitar mediante una profunda documentación de diseño que describa las estructuras de datos y dé una referencia que asocie los elementos de datos, los registros, los archivos y otras estructuras a los módulos del software.

**Efectos secundarios sobre la documentación.** El mantenimiento que se efectúa en el software se debe centrar en la completa configuración del software, no solo las modificaciones que se efectúan en el código fuente. Los efectos secundarios en la documentación es debido a que no se reflejan las modificaciones hechas en el código fuente sobre la documentación respectiva, no llevando a cabo un registro u/o actualización de los códigos fuente modificados.

Siempre que se realice una modificación sobre el flujo de datos, sobre la arquitectura de diseño sobre los procedimientos (módulos) o sobre cualquier otra característica asociada, se debe actualizar la información técnica soporte, la documentación actual no se reflejará en su totalidad en el software pero puede ser peor la ausencia total de este tipo de documentación.

## 2.5.2 MÉTODO DE DESARROLLO POR ANÁLISIS ESTRUCTURADO <sup>19</sup>

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo por análisis estructurado tiene como finalidad superar esta dificultad por medio de: <sup>19</sup>

- La división del sistema en componentes.
- La construcción de un modelo del sistema.

Pero ¿Qué es el análisis estructurado?, el análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece como se cumplirán los requerimientos o la forma en que implantarán la aplicación. Más bien permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadoras, terminales, sistemas de almacenamiento, etc.) Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado.

### 2.5.2.1 ELEMENTOS DEL ANÁLISIS ESTRUCTURADO

Los elementos esenciales del análisis estructurado son símbolos gráficos, diagramas de flujos de datos y el diccionario centralizado de datos.

*Descripción Gráfica.* Una de las formas de describir un sistema es preparar un bosquejo que señale sus características, identifique la función para la que sirve e indique como éste interactúa con otros elementos, entre otras cosas.

---

<sup>19</sup> Diseño de Sistemas de Información

Sin embargo, describir de esta manera un sistema grande es un proceso tedioso propenso a errores, ya que es fácil omitir algún detalle o dar una explicación que quizá los demás no entiendan.

En lugar de las palabras el análisis estructurado utiliza símbolos, o iconos, para crear un modelo gráfico del sistema. Los modelos de este tipo muestran los detalles del sistema pero sin introducir procesos manuales o computarizados, archivos en cinta o disco magnético, o procedimientos operativos y de programas.

Si se seleccionan los símbolos y notación correctos entonces casi cualquier persona puede seguir la forma en que los componentes se acomodarán entre sí para formar el sistema. El diagrama lógico de flujo de datos muestra las fuentes y los destinos de los datos, identifica y da nombre a los grupos de datos que relacionan una función con otra y señala los almacenes de datos a los que se tiene acceso.

**Diagrama de Flujo de Datos.** El modelo del sistema recibe el nombre de diagrama de Flujo de Datos (DFD). La descripción completa de un sistema está formada por un conjunto de diagramas de flujos de datos.

Para desarrollar una descripción del sistema por el método del análisis estructurado. Se sigue un proceso descendente (Top - Down). El modelo original se detalla en diagramas de bajo nivel que muestran características adicionales del sistema.

Cada proceso puede desglosarse en Diagramas de Flujo de Datos cada vez más detallados. Esta secuencia se repite hasta que se obtienen suficientes detalles que permiten al analista comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

**Diccionario de datos.** Todas las definiciones de los elementos en el sistema – Flujo de Datos, procesos y almacenes de datos – están descritos en forma detallada en el diccionario de datos. Si algún miembro del equipo encargado del proyecto desea saber alguna definición del nombre o dato o el contenido en particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos.

Ahora veamos que es el **Diseño estructurado**. Es otro elemento del análisis estructurado que emplea la descripción gráfica. Se enfoca en el desarrollo de especificaciones del software.

La meta del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional. Este enfoque no solo conduce a mejores programas sino que facilita el mantenimiento de los mismos cuando surja la necesidad de hacerlo.

El diseño estructurado es una técnica específica para el diseño de programas y no un método de diseño de comprensión. Es decir, no indica nada relacionado con el diseño de archivos o bases de datos, la presentación de entradas y salidas, la secuencia del procesamiento o del hardware que dará soporte a la aplicación.

Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes.

La herramienta fundamental del diseño estructurado es el diagrama estructurado. Al igual que los diagramas de flujo de datos, los diagramas estructurados son de naturaleza gráfica y evitan cualquier referencia relacionada con el hardware o detalles físicos.

Su finalidad no es de mostrar la lógica de los programas (que es la tarea de los diagramas de flujo). Los diagramas estructurados describen la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interacciona con él.

Estas especificaciones funcionales para los módulos se proporcionan a los programadores antes que de comienzo la fase de escritura de código.

### 2.5.2.2 ANÁLISIS ESTRUCTURADO CON OTROS MÉTODOS DE DESARROLLO

El análisis estructurado se combina, con bastante frecuencia, con el método antes descrito "Ciclo de vida Clásico de desarrollo de sistemas". Por ejemplo podemos optar por desarrollar diagramas de flujos de datos como una forma para documentar las relaciones entre componentes durante la investigación

detallada de algún sistema existente. Asimismo, se pueden definir los archivos y datos en un diccionario centralizado de datos de acuerdo con las reglas del análisis estructurado.<sup>20</sup>

### 2.5.3 MÉTODO DEL PROTOTIPO DE SISTEMAS

Este método hace que el usuario participe de manera más directa en la experiencia del análisis y diseño. El prototipo es un sistema que funciona – no sólo una idea en el papel -, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el sistema, al igual que cualquier sistema basado en computadora, está constituido por software que acepta entradas, salidas, realiza cálculos, produce información ya sea impresa o presentada en una pantalla, o que lleva a cabo otras actividades significativas. Es la primera versión, o iteración, de un sistema de información; es el modelo original.

Los usuarios evalúan el diseño y la información generada por el sistema. Lo anterior sólo puede hacerse con efectividad si los datos utilizados, al igual que las situaciones, son reales.

Por otra parte, se esperan cambios a medida que el sistema es utilizado.

Este método en realidad es un modelo piloto o prueba; el diseño evoluciona con el uso, si el empleo del prototipo revela que se cometen muchos errores, los diseñadores modificarán en lo posible el proceso para que disminuyan estos, automatizando lo más posible al sistema.

Aunque el prototipo es un sistema que funciona, está diseñado para ser modificado con facilidad. La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que revela más información valiosa sobre el diseño. El proceso se repite las veces que sea necesario para revelar los requerimientos esenciales del diseño.

En general los prototipos tienen mayor utilidad bajo las siguientes condiciones:

---

<sup>20</sup> Análisis y Diseño de Sistemas de Información

- Los encargados de diseñar e implantar sistemas nunca han desarrollado uno con las características del sistema propuesto.
- Se conoce solo una parte de las características esenciales del sistema; las demás no son identificables a pesar de un cuidadoso análisis de requerimientos.
- La experiencia con el uso del sistema añadirá una lista significativa de requerimientos que el sistema debe satisfacer (más que la que puede obtenerse con cualquier otro método de desarrollo).
- Las diferentes versiones del sistema evolucionan con la experiencia al igual que el desarrollo adicional y el refinamiento de sus características.
- Los usuarios del sistema que participan en el proceso de desarrollo.

El principio fundamental del desarrollo de prototipos es el siguiente:

Los usuarios pueden señalar las características que les agrada o no tener, junto con los problemas que presenta un sistema que existe y funciona, con mayor facilidad que si se les pidiese que las describieran en forma teórica o por escrito. El uso y la experiencia producen comentarios más significativos que el análisis de diagramas y las propuestas por escrito.

El desarrollo de prototipos de sistemas es un proceso interactivo, comienza con unas cuantas funciones y crece al incluir otras que son identificadas con posterioridad. También pueden comenzar con un conjunto de funciones que tanto el analista como los usuarios consideran completo y que puede aumentar o disminuir con el uso y la experiencia.

En general, los pasos a seguir en el proceso de desarrollo de prototipos son los siguientes:

1. Identificar los requerimientos de información que el usuario conoce junto con las características necesarias del sistema.

2. Desarrollar un prototipo que funcione.
3. Utilizar el prototipo anotando las necesidades de cambios y mejoras. Esto expande la lista de los requerimientos de sistemas conocidos.
4. Revisar el prototipo con base en la información obtenida a través de la experiencia del usuario.
5. Repetir los pasos anteriores las veces que sea necesario. Hasta obtener un sistema satisfactorio.

La construcción de prototipos no es un proceso de desarrollo por prueba y error. Antes que de inicio cualquier actividad de diseño o programación, el analista se reúne con los usuarios una o dos veces con la finalidad de identificar los requerimientos. El resultado de estas reuniones forma la base para la construcción del prototipo.

El desarrollo de un prototipo que funcione es responsabilidad del analista de sistemas. El diálogo de interfase permite a los usuarios actuar recíprocamente con el sistema, las rutinas de procesamiento y las salidas deben ser adecuadas (aunque no necesariamente completas) para que las personas puedan comprender como utilizar el sistema para realizar estas funciones.

Los mensajes y pantallas no incluidos en el prototipo de añaden más tarde, cuando se conoce un conjunto más completo de requerimientos.

## 2.6 HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS <sup>21</sup>

En general, una herramienta es cualquier dispositivo que, cuando se emplea de forma adecuada, mejora el desempeño de una tarea, tal como el desarrollo de sistemas de información basados en computadora. Las herramientas se agrupan en las siguientes categorías: análisis, diseño y desarrollo. <sup>21</sup>

---

<sup>21</sup> <http://www.monografias.com/trabajos6/sista/sista2.shtml#ana>

## 2.6.1 HERRAMIENTAS PARA ANÁLISIS

Estas herramientas ayudan a los especialistas en sistemas a documentar un sistema existente, ya sea éste manual o automatizado, y a determinar los requerimientos de una nueva aplicación. Estas herramientas incluyen:

- **Herramientas para Recolección de Datos.** Capturan detalles que describen sistemas y procedimientos en uso. Documentan procesos y actividades de decisión. Se utilizan para apoyar la tarea de identificar requerimientos.
- **Herramientas para Diagramación.** Crean representaciones gráficas de sistemas y actividades. Apoyan el dibujo y la revisión de diagramas de flujo de datos e iconos asociados con el análisis estructurado. Asimismo incluyen programas para representación de diagramas de flujo
- **Herramientas para el Diccionario.** Registran y mantienen descripciones de los elementos del sistema, tales como grupos de datos, procesos y almacenamiento de datos. Con frecuencia proporcionan la capacidad de examinar las descripciones del sistema para decidir si son incompletas o inconsistentes. Muchas incluyen la facilidad de reportar donde se utilizan los elementos del sistema.

Las herramientas con mayor utilidad, en cualquier categoría, están siendo ya automatizadas tanto para mejorar la eficiencia de analista, como para permitir obtener del esfuerzo de análisis, resultados más completos y exactos.

## 2.6.2 HERRAMIENTAS PARA DISEÑO

Las herramientas de diseño apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos detectados durante las actividades del análisis:

- **Herramientas de Especificación.** Apoyan el proceso de formular las características que debe tener una aplicación, tales como entradas, salidas, procesamiento y especificaciones de control. Muchas incluyen herramientas para crear especificaciones de datos.
- **Herramientas para Presentación.** Se utilizan para describir la posición de datos, mensajes y encabezados sobre las pantallas de las terminales, reportes y otros medios de entrada y salida.

### 2.6.3 HERRAMIENTAS PARA DESARROLLO

Estas herramientas ayudan al analista a trasladar los diseños en aplicaciones funcionales:

- **Herramientas para Ingeniería de Software.** Ayudan al proceso de formular diseños de software, incluyendo procedimientos y controles, así como la documentación correspondiente.
- **Generadores de código.** Producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas.
- **Herramientas para Pruebas.** Apoyan la fase de evaluación de un sistema o de partes del mismo contra las especificaciones. Incluyen facilidades para examinar la correcta operación del sistema así como el grado de perfección alcanzado en comparación con las expectativas.

La infusión de procesamiento computarizado, aunado con prácticas de diseño sofisticadas, está cambiando en forma dramática la manera en que se trasladan las especificaciones de diseño en sistemas de información funcionales.

La revolución del procesamiento de datos de manera computarizada, junto con las prácticas de Diseño sofisticadas están cambiando de forma dramática la manera en que se trasladan las especificaciones de Diseño d Sistemas de Información funcionales.

**En Conclusiones Generales.** En una organización o Empresa, el análisis y Diseño de Sistemas, es el proceso de estudiar su Situación con la finalidad de observar como trabaja y decidir si es necesario realizar una mejora; el encargado de llevar a cabo estas tareas es el analista de sistemas.

Antes de comenzar con el desarrollo de cualquier proyecto, se conduce un estudio de Sistemas para detectar todos los detalles de la situación actual de la empresa. La información reunida con este estudio sirve como base para crear varias estrategias de Diseño. Los administradores deciden que estrategias seguir. Los Gerentes, empleados y otros usuarios finales que se familiarizan cada vez mas con el uso de computadoras están teniendo un papel muy importante en el desarrollo de sistemas.

Todas las organizaciones son Sistemas que actúan de manera reciproca con su medio ambiente recibiendo entradas y produciendo salidas. Los Sistemas que pueden estar formados por otros Sistemas de denominan Sub-sistemas y funcionan para alcanzar los fines de su Implantación.

## CAPITULO 3. HERRAMIENTAS DE LIBRE DISTRIBUCION (GUI)

### 3.1 ¿ QUÉ ES EL SOFTWARE LIBRE?

"Software Libre" es cuestión de libertad no de Precio. "Free" en "Free software" , es una palabra que debe ser traducida como "libre" tal como "Libertad de Expresión " y no como "Gratis".

"Software Libre" se refiere a la libertad de los usuarios de correr, copiar, distribuir, estudiar, cambiar y mejorar el Software. Esto se ejemplifica de una forma más precisa en las cuatro libertades definidas para los usuarios de "Software Libre".<sup>1</sup>

- a) La libertad de correr el programa, con cualquier propósito (Libertad 0).

---

<sup>1</sup> <http://www.gnu.org/philosophy/free-sw.es.html>

- b) La libertad de estudiar como funciona el programa, y adaptarlo a sus necesidades (Libertad 1). El acceso al código fuente es una precondition para esto.
- c) La libertad de distribuir copias de manera que se pueda ayudar al vecino (Libertad 2).
- d) La libertad de mejorar el programa, y liberar las mejoras al público de tal manera que toda la comunidad se beneficie (Libertad 3). El acceso al código fuente es una precondition para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Por esto, eres libre de redistribuir copias, ya sea con o sin modificaciones, ya sea gratis o cobrando una cuota por la distribución a cualquiera y a cualquier lugar.

El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También se tiene la libertad de hacer modificaciones y utilizarlas de manera privada para el trabajo o juego, sin siquiera mencionar que dichas modificaciones existen. Si se publican los cambios, no es necesario avisar a alguien en particular.

Se pudo haber pagado por obtener copias del software, o tal vez se obtuvieron las copias sin ningún cargo. Pero no importa como se obtuvieron las copias, se tiene la libertad de cambiar y copiar el Software.

Por supuesto que para que estas libertades sean reales, deben ser irrevocables conforme no se haga algo malo, el desarrollador de software tiene el poder de revocar la licencia, inclusive si no se ha dado causa, el software no es libre.

Sin embargo, algunos tipos de reglas acerca de la manera de distribuir software libre son aceptables, cuando no entran en conflicto con las libertades centrales.

Por ejemplo **copyleft** (expresado muy simplemente) es la regla que implica que cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla o esta en contra de las libertades centrales sino que las protege.

Las reglas acerca de cómo empaquetar una versión modificada son aceptables, si ellas no bloquean efectivamente la libertad de liberar las versiones modificadas. Reglas que indican "Si tu haces

disponible el programa de esta manera, debes también hacerlo en esta otra", pueden ser aceptables también, bajo la misma condición. (Esta regla deja la elección de hacer o no disponible el software).

Algunas veces las reglas de control de exportación del gobierno y las sanciones del mercado puede reprimir la libertad de distribuir copias de los programas a escala internacional. Los desarrolladores de software no tiene el poder para eliminar o sobrepasar estas restricciones, pero lo que pueden y deben hacer es rehusar el imponerlas como condiciones de uso del programa. De esta manera, las restricciones no afectan actividades y gente fuera de la jurisdicción de estos gobiernos.

Cuando se habla de software libre, es mejor evitar términos como: "regalar" , o "gratis", porque estos términos implican que lo importante es el precio y no la libertad. Algunos términos comunes como "pirataje" contraen opiniones que espero no se apoyen.

## 3.2 CÓMO OBTENER EL SOFTWARE LIBRE

El software libre está disponible de varias formas:

- Compras copias en CD-ROM de la fundación para software libre (FSL).
- Bajarlo a través de FTP (File Transfer Protocol).
- Obtener una copia de un amigo
- Compra una computadora con un sistema GNU/LINUX completamente libre preinstalado de una de las compañías que pueden ofrecerlos.

### 3.2.1 CATEGORÍAS DEL SOFTWARE LIBRE Y NO LIBRE

Aquí mostramos un glosario de varias categorías de software que son mencionadas frecuentemente en discusiones de Software Libre. Explicando que categorías se traslapan o son parte de otras categorías.<sup>2</sup>

---

<sup>2</sup> <http://www.gnu.org/philosophy/categories.es.html/>

1. **Software Libre.** Este software viene con autorización para que cualquiera pueda usarlo, copiarlo y distribuirlo, ya sea literal o con modificaciones, gratis o mediante una gratificación.

Esto significa que el código fuente debe estar disponible. "Si no es fuente, no es software". Si un programa es libre, entonces puede ser potencialmente incluido en un sistema operativo libre tal como GNU, o sistemas GNU/LINUX libres.

2. **Software de Fuente Abierta.** El término software de fuente abierta es usado por algunas personas para dar a entender más o menos lo mismo que software libres, se prefiere el término software libre.
3. **Software de dominio público.** El software de dominio público es software que no está protegido con copyright. Es un caso especial de software libre no protegido con copyleft, que significa que algunas copias o versiones modificadas no pueden ser libres completamente.

Algunas veces la gente utiliza el término "dominio público" de una manera imprecisa para decir "libre" o "disponible gratis". Sin embargo, "Dominio público" es un término legal y significa de manera precisa "sin copyright". Por claridad, se recomienda que el uso de "dominio público" sea para este significado solamente y el uso de otros términos sean utilizados para transmitir los demás significados.

4. **Software protegido con Copyleft.** El software protegido con copyleft es software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando estos redistribuyen o modifican el software. Esto significa que cada copia del software, aún si ha sido modificado, debe ser software libre.

En el proyecto GNU, se protege mediante copyleft casi todo el software que se desarrolla, por que el objetivo es dar a cada usuario las libertades que el termino "Software libre" implica.

Copyleft es un concepto general; para proteger actualmente un programa con copyleft, necesita usar un conjunto específico de términos de distribución .

5. **Software libre no protegido con copyleft.** Este software viene desde el autor con autorización para redistribuir y modificar así como para añadirle restricciones adicionales.

Si un programa es libre pero no protegido por copyleft, entonces algunas copias o versiones modificadas pueden no ser libres completamente. Una compañía de software puede modificar y compilar el programa y posteriormente distribuirlo como propietario del software.

6. **Software abarcado por GPL.** La GPL (General Public License) de GNU es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El proyecto GNU la utiliza como los términos de distribución para la mayoría del Software GNU.
7. **EL sistema GNU.** Es un sistema operativo libre completo estilo UNÍX. El cual consiste en muchos programas. El sistema GNU incluye todo el software GNU, así como muchos otros paquetes tales como el sistema X Window y TeX que no son software GNU.

Debido a que el propósito de GNU es ser libre, cada componente individual en el sistema GNU tiene que ser software libre. No todos tienen que estar protegidos con copyleft, sin embargo; cualquier tipo de software libre es legalmente apto de incluirse si ayuda a alcanzar metas técnicas. Podemos hacer uso de software libre no protegido con copyleft como el Sistema X Window.

8. **Software GNU.** Este software es liberado bajo el auspicio del Proyecto GNU. La mayoría del Software GNU está protegido por Copyleft, pero no todos; sin embargo todo el Software GNU, debe ser libre.

Algo de software GNU es escrito por el personal de la fundación para el software libre (Free Software Foundation), pero la mayoría del software GNU es aportada por voluntarios. Parte del software aportado esta protegido con Copyright por la Fundación para el Software Libre; otra parte está protegido con copyright por los aportadores que lo escribieron.

9. **Software Semilibre.** El software semilibre no es libre, pero viene con autorización para particulares de usar, copiar, distribuir y modificar (incluyendo la distribución de versiones modificadas) sin fines de lucro.

Cabe mencionar que sería imposible incluir software semilibre en un sistema operativo libre. Esto obedece a que los términos de distribución para el sistema operativo libre como un todo es la conjunción de los términos de distribución de todos los programas en él. Agregando un programa semilibre al sistema haría al sistema semilibre.

Y de echo existen algunas razones por las que no se quiere que esto sucede. Definitivamente creo que el software libre debe ser para todos tal como lo indica la GNU. Si se incluyera software semilibre se perdería por completo la finalidad para la que fue creado.

Este software debe ser libre para todos incluyendo empresas, no solamente para escuelas o para quienes gusten de matar el tiempo.

La distribución comercial de sistemas operativos libres, incluyendo sistemas GNU basados en Linux, es muy importante y los usuarios aprecian poder comprar distribuciones comerciales en CD-ROM. Incluyendo un programa semilibre en un sistema operativo cortaría su distribución comercial en CD- ROM.

10. **Software propietario.** Este software no es libre ni semilibre, su uso, redistribución o modificación está prohibida, o requiere que se solicite autorización o está tan restringida que no pueda hacerla libre de un modo efectivo.

La fundación para el software libre sigue la regla de no instalar ningún programa propietario en nuestras computadoras excepto temporalmente para el propósito específico de escribir un reemplazo libre para ese programa. Con excepción de lo anterior no hay excusa posible para instalar un programa propietario.

11. **Freeware.** Este término no tiene una definición clara aceptada, pero es usada comúnmente para paquetes que permiten la redistribución pero no la modificación (y su código fuente no está disponible). Estos paquetes no son software libre.
12. **Shareware.** El Shareware es software que viene con autorización para la gente de redistribuir copias, pero dice que quien continúe haciendo uso de una copia deberá pagar un cargo por licencia.
- 13.

**El Shareware no es software libre, ni siquiera semilibre. Existen dos razones por las que no lo es: para la mayoría del Shareware, el código fuente no está disponible; de esta manera, no puede modificar el programa en absoluto.**

El Shareware no viene con autorización para hacer una copia e instalarlo sin pagar una cantidad por licencia, ni aún para particulares involucrados en actividades sin ánimo de lucro.

14. **Software Comercial.** Este software es desarrollado por una entidad que tiene la intención de hacer dinero del uso del software. "Comercial" y "propietario". la mayoría del software comercial es propietario. Pero hay software libre comercial y hay software no libre no comercial.

### 3.3 SISTEMA OPERATIVO LINUX <sup>3</sup>

La mayor parte de las computadoras que existen en la actualidad están diseñados de forma que puedan ejecutar diversas tareas o programas, estos programas pueden ir desde un procesador de textos, a un programa para la animación de gráficos tridimensionales o distintos tipos de juegos.

Para su correcto funcionamiento deben ser además capaces de acceder a los recursos de que dispone la computadora, como por ejemplo leer o escribir datos en un disco duro, mostrar un gráfico por pantalla, etc.

Es evidente que si cada programa actuase de una forma independiente, existirían graves problemas y conflictos, puesto que, por ejemplo tendrían libertad para escribir sus datos sobre los de otro etc.

Para solucionar este tipo de problemas se desarrollaron los sistemas operativos, los cuales aportan unos mecanismos y reglas básicas de funcionamiento, de forma que los programas puedan acceder a los recursos de la PC de una forma adecuada.

Aunque esta fue la funcionalidad inicial de los sistemas operativos, con el tiempo se han añadido muchas otras, como la ejecución de programas, el control de la memoria de la máquina, la creación y control de interfaces gráficas de usuario etc.

En la actualidad existe una gran cantidad de sistemas operativos dependiendo del tipo de computadora en el que se va a ejecutar. Por ejemplo para los PC uno de los sistemas operativos más difundidos es Microsoft Windows en sus diferentes versiones (95 , 98 , 2000 , NT 4.0), por mencionar algunas.

Otros posibles sistemas operativos para este tipo de PC's son Solaris, OS/2, BeOS, Microsoft DOS, o uno de los sistemas operativos más poderosos y en rápida expansión para PC, LINUX.

### 3.3.1 HISTORIA DE LINUX

Unix es uno de los sistemas más populares del mundo debido a su extenso soporte y distribución. Originalmente fue desarrollado como sistema multitarea con tiempo compartido para mini computadoras y mainframes a mediados de los 70, y desde entonces se ha convertido en uno de los sistemas más utilizados a pesar de su, ocasionalmente, confusa interfaz con el usuario y el problema de su estandarización.<sup>4</sup>

Pero, ¿cuál es la verdadera razón de la popularidad de Unix? Muchos Hackers consideran que UNIX es el auténtico y único sistema operativo. El desarrollo de Linux parte de un grupo en expansión de

---

<sup>4</sup> <http://linux.galeon.com/>

Hackers de Unix que quisieron hacer un sistema con sus propias manos. Existen numerosas versiones de UNIX para muchos sistemas, desde computadoras personales hasta supercomputadoras.

Linux como ya se ha mencionado en repetidas ocasiones es una versión de UNIX de libre distribución, inicialmente desarrollada por Linus Torvalds en la universidad de Helsinki, en Finlandia. Fue desarrollado con la ayuda de muchos programadores y expertos de Unix a lo largo y ancho del mundo, gracias a la presencia de Internet. Cualquier habitante del planeta puede acceder a Linux y desarrollar nuevos módulos o cambiarlo a su antojo.

El núcleo de Linux no utiliza ni una sola línea del código AT&T o de cualquier otra fuente de propiedad comercial, y buena parte del software para Linux se desarrolla bajo las reglas de GNU de Free Software Foundation, Cambridge, Massachusetts.

Inicialmente solo fue un proyecto de aficionado de Linus Torvalds, se inspiraba en Minix, un pequeño Unix desarrollado por Andy Tanenbaum, y las primeras discusiones sobre Linux surgieron del grupo de News compo Minix. Estas discusiones giraban en torno al desarrollo de un pequeño sistema UNIX de carácter académico dirigido a aquellos usuarios de Minix que querían algo más.

El desarrollo inicial de Linux ya aprovechaba las características de conmutación de tareas en modo protegido del 386, y se escribió todo en ensamblador, al final de agosto de 1991 Linus tenía ya la versión 0.01, un driver de discos con numerosos bugs, así como un pequeño sistema de ficheros pero que parecían funcionar en una PC.

No se anunció nada sobre esta versión puesto que las fuentes del 0.01 jamás fueron ejecutables : contenían solo rudimentos de lo que quería ser el núcleo, y se asumía que se tenía acceso a un Minix para poderlo compilar y jugar con él.

El 5 de octubre de 1991, Linus anunció la primera versión oficial de Linux , 0.02. ya podía ejecutar bash (el shell de GNU) y gcc (el compilador de GNU), pero no hacía mucho más, la intención era ser un juguete para Hackers, no había nada sobre soporte a usuarios, distribuciones , documentación ni nada parecido.

Tras la versión 0.03 Linus salto de la versión 0.10, al tiempo que más gente empezaba a participar en su desarrollo. Tras numerosas revisiones se alcanzó la versión 0.95, reflejando la esperanza de tener

muy pronto una versión "Oficial". (Generalmente , la versión 1.0 de los programas de corresponde con la primera teóricamente completa y sin errores ). Esto sucedía en marzo de 1992. año y medio después, en Diciembre del 93, el núcleo estaba en la revisión 0.99 pl14, en una aproximación asintótica al 1.0.

El día de hoy Linux es ya un clónico de UNIX completo, capaz de ejecutar X window, TCP/IP, Emacs,UUCP y software de correo y News. Mucho software de libre distribución ha sido ya portado a Linux, y están empezando a aparecer aplicaciones comerciales . el hardware soportado es mucho mayor que en las primeras versiones del núcleo.

Hemos hablado sobre la libre distribución de este software pero , ¿Qué son las distribuciones ?, definitivamente debemos destacar que uno de los primeros conceptos que aparecen al iniciarse Linux es el concepto de la distribución, una distribución es un agrupamiento del núcleo del sistema operativo Linux (la parte desarrollada por Linus Torvalds) y otra serie de aplicaciones de uso general o no tan general.

En principio las empresas que desarrollan las distribuciones de Linux están en su derecho al cobrar cierta cantidad por el software que ofrecen, aunque la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde Internet, de revistas o de amigos, siendo todas estas formas gratuitas y legales.

Las distribuciones más conocidas son RedHat, Debian, Slackware, Suse, Corel Linux y Mandrake todas ellas incluyen el software más reciente y empleado lo cual incluye compiladores de C, C++, editores de texto, juegos, programas para el acceso a internet, así como el entorno gráfico de Linux: X Window.

Para la presentación de esta Tesis, el sistema se desarrollo bajo Linux Mandrake 8.0

### 3.3.2 CARACTERÍSTICAS DEL SISTEMA LINUX

Linux toma la mayor parte de las características que se encuentran en otras implementaciones de UNIX, más algunas otras que no son habituales.

Linux es un sistema operativo completo con multitarea y multiusuario (como cualquier otra versión de UNIX). Esto significa que pueden trabajar varios usuarios simultáneamente en él, y que cada uno de ellos puede tener varios programas en ejecución.<sup>5</sup>

El sistema Linux es compatible con ciertos estándares de UNIX a nivel de código fuente, incluyendo el IEEE.POSIX.1, System V y BSD. Fue desarrollado buscando la portabilidad de los fuentes: encontrará que casi todo el software gratuito desarrollado para UNIX se compila en Linux sin problemas. Y todo lo que se hace para Linux (código del núcleo, drivers, librerías y programas de usuario) es de libre distribución.

En Linux también se implementa el control de trabajos POSIX(que se usa en los shells csh y bash), las pseudoterminals (dispositivos pty), y teclados nacionales mediante manejadores de teclado cargables dinámicamente. Además soporta consolas virtuales, lo que permite tener más de una sesión abierta en la consola de texto y conmutar entre ellas fácilmente.

Linux soporta diversos sistemas de ficheros para guardar los datos. Algunos de ellos como el ext2fs, han sido desarrollados específicamente para linux, otros sistemas de ficheros como el Minix-1 o el de Xenix también están soportados. Y con el MS-DOS se podrán acceder desde Linux a los disquetes y particiones en discos duros formateados en MS-DOS, además también soporta el ISO-9660, que es el estándar seguido en el formato de los CD-Roms.

Linux implementa todo lo necesario para trabajar en la red con TCP/IP, desde manejadores para las tarjetas de red más populares hasta SLIP/PPP, que permiten acceder a una red TCP/IP por el puerto serial. También se implementan PLIP (para comunicarse por el puerto de la impresora) y NFS (para acceso remoto a ficheros).

El núcleo de Linux ha sido desarrollado para utilizar las características del modo protegido de los microprocesadores 80386 y 80486. en concreto hace uso de la gestión de memoria avanzada del modo protegido y otras características avanzadas.

El núcleo soporta ejecutables con paginación por demanda. Esto significa que solo los segmentos del programa que se necesiten se cargan en memoria desde el disco. Las páginas de los ejecutables son

---

<sup>5</sup> <http://www.linux-gpl.com/>

compartidas mediante la técnica *copy-on-write*, contribuyendo todo ello a reducir la cantidad de memoria requerida para las aplicaciones. Con el fin de incrementar la memoria disponible, Linux implementa la paginación con el disco; puede tener hasta 256 megabytes de espacio de intercambio o swap en el disco duro.

Cuando el sistema necesita más memoria, expulsará páginas inactivas al disco, permitiendo la ejecución de programas más grandes o aumentando el número de usuarios que pueden atender a la vez. Sin embargo, el espacio de intercambio no puede suplir totalmente la memoria RAM, ya que el primero es mucho más lento que esta.

La memoria dedicada a los programas y a la cache del disco está unificada. Por ello, si en cierto momento hay mucha memoria libre, el tamaño de la cache de disco aumentará acelerando así los accesos. Los ejecutables hacen uso de las librerías de enlace dinámico. Esto significa que los ejecutables comparten el código común de las librerías en un único fichero, como sucede en SunOS. Así los ejecutables serán más cortos a la hora de guardarlos en el disco, incluyendo aquellos que hagan uso de muchas funciones de librería.

También pueden enlazarse estáticamente cuando se deseen ejecutables que no requieran la presencia de las librerías dinámicas en el sistema. El enlace dinámico se hace en tiempo de ejecución, con lo que el programador puede cambiar las librerías sin necesidad de recompilación de los ejecutables.

Para facilitar la depuración de los programas, el núcleo de Linux puede generar volcados de la imagen de memoria de los programas (ficheros core). Entre esto y la posibilidad de compilar ejecutables con soporte de depuración, el programador podrá averiguar la causa de los fallos de su programa.<sup>6</sup>

### 3.3.3 LICENCIA 7

Linux no es del dominio público, pero bajo la Licencia Pública de GNU, el código fuente para Linux debe siempre permanecer libremente disponible. La gente puede cobrar dinero por Linux si lo desea, en tanto que no intente limitar la redistribución de Linux.

---

<sup>6</sup> <http://linux-mandrake.com/en/> y <http://www.linux.com>

Mucha gente alrededor del mundo ha trabajado en equipo para crear Linux, bajo los lineamientos de Linus Torvald, el autor original, y cada uno mantiene los derechos de autor del código que el o ella haya escrito.

Linux contiene todas las características de un sistema operativo moderno, como por ejemplo: Multitarea Real, Memoria Virtual, Multiusuario, TCP/IP nativo, librerías compartidas, kernel modulable, funciona en modo protegido, soporte para multiprocesamiento simétrico y soporte para hardware de 32 y 64 bits.

Existe una variedad de software disponible para Linux, la mayoría de el puede conseguirse de manera gratuita bajo la licencia GNU GPL, permitiendo que Linux sea configurado para distintas aplicaciones: Servidor de archivos, Servidor Web, Base de Datos Relacional, Servidor de correo, FTP, etc. (o todos ellos al mismo tiempo); así como también puede actuar como Firewall, Servidor de Terminales, Router, etc.

Linux tiene una implementación libre de X Windows, basado en el estándar X/OPEN que le permite ejecutar entornos gráficos de alta calidad. Sobre estos pueden ejecutarse una variedad de manejadores de escritorio (Desktop Managers) según las necesidades o el gusto del usuario.

Es importante destacar la capacidad que tiene Linux para conectarse a redes basadas en protocolos diversos como TCP/IP, AppleTalk, IPX/SPX, Netbios, etc. Incluso puede reemplazar funciones de otros servidores en estas redes de manera transparente y funcional.

Debido al éxito de Linux, respaldado por su funcionalidad y disponibilidad en todo el mundo, las empresas más importantes de software han lanzado versiones de sus productos para Linux, así, por ejemplo, tenemos a Oracle, Informix, Corel, Netscape, IBM, HP, etc.

También las empresas de hardware, están lanzando equipos preconfigurados con Linux y prestan soporte linux para sus máquinas, como Compaq, Dell, IBM, SUN, HP, etc. lo que asegura un futuro prometedor.

El desarrollo del kernel del Linux, que es el corazón del sistema operativo, está en constante desarrollo y se encuentra soportado por una multitud de programadores comunicados a través de Internet bajo la coordinación de Linus Torvald; sin embargo, como el mismo lo reconoce, las versiones actuales de kernel, sólo tienen de un 5 a 10 por ciento de su propiedad.

Esto garantiza que el futuro del Linux no dependa de una sola persona, sino más bien de un grupo abierto de desarrolladores a nivel mundial.

GNU/Linux® no es *Freeware*. Es *software libre*. Cuando hablamos de software libre, nos referimos a la libertad y no al precio. Las Licencias Públicas Generales (GPL), a la cual Linus Torvalds incorporó a Linux, están pensadas para asegurar que el usuario tiene la libertad de distribuir copias del software libre (y cobrar por el servicio si así lo desea).

El software de Código Abierto, también conocido por el nombre genérico de Software Libre, nació con el advenimiento de Unix en los años 70.

Este sistema operativo fué distribuido con su código fuente, por lo que los usuarios, la mayor parte de las veces en las universidades y en los laboratorios de investigación, podían encargarse del mantenimiento del mismo ellos mismos. Ya que la asistencia técnica estaba faltando, las comunidades de especialistas en computadoras estaban organizados para mantener, mejorar y distribuir el sistema operativo.

En 1983, Richard Stallman del MIT fundó la Fundación de Software Libre - Free Software Foundation, una organización que abogaba por el uso del software libre. Este proyecto, llamado GNU (Gnu no es Unix) rápidamente tomó forma, y la GPL, Licencia Pública General (General Public License), que proporcionaba una base legal para el uso y distribución del software libre, fué escrita también.

### 3.3.4 LAS FUERZAS DEL SISTEMA OPERATIVO LINUX

El proceso de desarrollo del software de código abierto reside en los equipos internos de editores de software así como en comunidades independientes asociadas con el proyecto, incluyendo a usuarios y expertos en computadoras. El desarrollo de Internet ha disparado la cantidad de colaboradores de parte de las comunidades externas y ha mejorado drásticamente la eficiencia del proceso de desarrollo del código abierto.

Por otro lado, se han diseñado para la web muchos programas de código abierto de bajo costo y consecuentemente intensificado el crecimiento de Internet en sí mismo.

El acceso a su código fuente permite a usuarios experimentados el adaptar y optimizar el sistema basado en sus propias especificidades. Proporciona la flexibilidad necesaria para adaptarse a un gran número de entornos técnicos.

Además, es compatible con todo tipo de hardware, incluso en aquel que ya se considera obsoleto. Ayuda a que las inversiones hechas tanto por compañías como por particulares tengan una vida más larga: estos usuarios ya nunca más tienen que tirar sus viejas computadoras a la basura.

Obviamente, el rendimiento de Linux se puede ver mejor con el hardware de última generación, donde las aplicaciones standard como Oracle, corren de un 20 a un 30% más rápido. Otra ventaja es la disponibilidad de Linux para plataformas cada vez más frecuentes: Apple's PowerPC, i386, Pentium I, II y III de Intel y compatibles, Ultra Sparc de Sun, Digital's Alpha.

En otras palabras, Linux está diseñado para ayudar a que las plataformas más heterogéneas del un cada vez más amplio parque de computadoras se comuniquen unas con otras.

Como Linux permite que los desarrolladores adapten el código fuente de los programas acorde con sus necesidades, es una opción muy atractiva para corporaciones que típicamente y de manera profusa invierten en sistemas operativos y software de propiedad.

En el año pasado, Linux ha ganado un gran apoyo por parte de gran parte de la industria, y ha llevado a cabo alianzas con compañías tales como IBM, Hewlett-Packard, Compaq, Dell, Oracle, Sun, SGI, Netscape y Apple.

### 3.3.5 ¿QUIÉN USA LINUX?

Linux es el único sistema operativo Unix cuya participación de mercado aumenta cada año que pasa. Sus usuarios pueden ser divididos en varias categorías:

La primera categoría está compuesta por **aficionados a la informática**, los cuales generalmente saben mucho de sus computadoras, y son perfectamente capaces de apreciar la estabilidad y funcionamiento del sistema.

Un número creciente de nuevos usuarios está descubriendo Linux porque este proporciona un solución completa (sistema operativo y aplicaciones), lo cual asegura el funcionamiento, estabilidad, simplicidad y amigabilidad a un costo muy bajo. El desarrollo y multiplicación de las apariciones de Linux en la Prensa son un signo del interés que el Público General tiene por el sistema.

La tercera categoría cubre el **pequeño negocio**. Además de la importancia económica del fenómeno del software de código abierto (costes de licencia menores), Linux corre en cualquier plataforma, incluso en aquellas que tecnológicamente se consideran anticuadas. Además, la fiabilidad del sistema asegura que no hay necesidad de una presencia constante de un administrador del sistema, mientras que esto sí es indispensable en sistemas tales como Netware, SCO Unix ó Windows NT.

**Usuarios clave**, tales como operadores Telecom y proveedores de acceso a Internet (el servidor de web Apache representa el 50% del mercado de servidores Web (fuente:Netcraft)). La fiabilidad y eficiencia de este sistema operativo ha convencido a estos clientes exigentes.

**Universidades y centros de investigación** se benefician de las plataformas no sólo para desarrollo sino también para entrenamiento y enseñanza. Y lo que es más, Linux les proporciona agrupamientos aritméticos científicos y automatización científica de oficina. Finalmente, los industriales reconocen la calidad de Linux y lo están integrando en su oferta de productos. El libre acceso al código fuente, asegurado por la licencia GPL, hace de Linux la elección líder para aplicaciones embutidas en todo tipo de productos (PABX, monitores, set-top, boxes, Imóviles, PDAs, etc.)

### 3.3.6 ¿PORQUÉ DEBERÍAMOS USAR LINUX?

GNU/Linux® es un sistema operativo estable, confiable y robusto. Está hecho para poder trabajar las 24 horas del día los 365 días del año, e incluso durante varios años SIN sufrir colapso. <sup>8</sup>

Y como el resto de los sistemas operativos, tiene ventajas y desventajas:

#### A Favor:

- No hay (aún) virus que realmente representen una amenaza para Linux. Esto significa que usted no necesitará comprar una licencia de algún antivirus. Además, los virus, en su mayoría creados para Windows® y programas que trabajan en este, no pueden ser ejecutados en GNU/Linux® y, por lo tanto, sus archivos e información se encontrarán a salvo.
- Los *Crackers* están más interesados en perjudicar a todo lo que tenga que ver con instituciones gubernamentales, Microsoft® y/o Windows®. Debe saber que Linux® le permitirá, utilizando las herramientas apropiadas, controlar el acceso a los puertos y bloquear intrusiones.

Si se presenta un "error" o algo se "cae", no es necesario reinicializar todo el sistema, bastará con "matar" y reiniciar la aplicación, programa o servicio. Usted no perderá tiempo y productividad. En GNU/Linux®

- los servicios como Sendmail, Servidores web, demonios en general, aplicaciones, etc, se desempeñan de forma independiente.
- Se puede actualizar el sistema operativo (el mismo kernel) sin necesidad de reinicializar. Esto significa que se ahorran muchas horas/hombre.
- GNU/Linux® y los programas para éste, a diferencia de otros sistemas operativos, son hechos desinteresadamente por personas que aman lo que hacen. podemos estar seguros de que los programadores han puesto realmente su máximo empeño y alma en su trabajo.

---

<sup>8</sup> <http://www.proyectolinux.org/>

- GNU/Linux® es software libre, es decir, la licencia del mismo le permite instalarlo en la cantidad de equipos que se desee. Esto quiere decir que si usted ha comprado un CD de instalación, o bajo gratuitamente los archivos de instalación, y posee dos o más computadoras, podrá instalar GNU/Linux® en todas estas. Cada distribución de GNU/Linux incluye el código fuente, el cual usted es libre de modificar a sus necesidades.
- Los métodos de seguridad de Linux son mejores que los de los "otros" sistemas operativos, por lo que es menos probable que seamos víctima de un "Hacker" o que se filtre información fuera de la PC sin su autorización. En Linux el acceso a los directorios y los archivos, así como la capacidad de borrado o modificación, depende de los permisos de usuario que estos tengan.
- Linux no nos sorprende con las "pantallas azules" que indican que algo salió mal y que debe reiniciar su equipo.

#### En contra:

- Se necesita algo de experiencia y algunos conocimientos básicos de Unix® para poder configurarlo adecuadamente, sobre todo lo relacionado con multimedia y redes. El usuario está obligado por las circunstancias a capacitarse adecuadamente.
- No hay tanto software para GNU/Linux® como lo hay para Windows®, y definitivamente no encontraremos una versión de Internet Explorer® para GNU/Linux®, hay otras mejores formas de navegar de forma insegura por Internet y, al mismo tiempo, delatar su identidad y contenido del disco duro a cualquiera.
- Sin embargo un gran número de empresas, como Corel, Netscape® y Sun Microsystems®, han puesto sus ojos en GNU/Linux®, dado que es un campo virgen y sin la competencia de Microsoft®. Actualmente encontraremos desde aplicaciones solitarias como Corel WordPerfect 8, hasta Suites completas para satisfacer sus necesidades, como lo es Corel WordPerfect Office 2000 o StarOffice. Que nos van a permitir abrir y editar documentos de MS-Office.
- Algunos importantes fabricantes de hardware se rehusan a proporcionar controladores para el funcionamiento de sus dispositivos bajo GNU/Linux®. Algunos argumentan que la cantidad de usuarios de GNU/Linux® no amerita su atención, otros no lo han hecho por falta de personal capacitado.

- Por fortuna esto tiene una fuerte tendencia a cambiar debido el incremento en la demanda por Linux® en el sector empresarial y doméstico.

### 3.4 BASE DE DATOS MYSQL

#### ¿Qué es SQL?

SQL es un método basado en un potente lenguaje, para organizar, administrar y consultar datos almacenados en una computadora. El modelo SQL de sus siglas en inglés (Structured Query Language) (Estructura de lenguaje estructurado), está definido en torno al modelo de bases de datos relacionales, basado en el álgebra relacional, esto le da al SQL las ventajas que lo imponen como el sistema de mayor aceptación.

Algunas de las ventajas con las que cuenta SQL son las siguientes:

- Marco teórico sólido, fundamentado en el álgebra relacional.
- Simplicidad de conceptos (modelo de bases de datos: tablas=líneas x columnas)
- Definición de vínculos en la consulta, esto le da a SQL una gran flexibilidad
- Fácil y rápido aprendizaje.
- Arquitectura cliente- servidor
- Integración con cualquier lenguaje de programación
- Estandarización.

#### 3.4.1 BREVE HISTORIA DE MYSQL

IBM comenzó a comercializar en 1981 el SQL y desde entonces este producto ha tenido un papel importante en el desarrollo de las bases de datos relacionales. IBM propuso y fue aceptada una versión de SQL al instituto de Estándares Nacional Americano (ANSI) y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

En 1983 nació DB2 la más popular (por lo menos de las grandes computadoras) de las bases de datos de este tipo hasta estos mismos momentos.

Alrededor de la década de los 90s, Michael Widenis ([monty@analytikerna.se](mailto:monty@analytikerna.se)) comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM), sin embargo después de muchas pruebas llegó a la conclusión de que mSQL no era lo suficientemente rápido ni flexible para sus necesidades. De todo esto surgió la nueva interfaz SQL (claro que con código más portable) con algunas apariencias similares en la API de C y en los nombres y funciones de muchos de sus programas.

Esto había sido hecho para lograr con relativa facilidad portar aplicaciones y utilidades de MiniSQL a MySQL.

El origen del nombre MySQL no está perfectamente claro. Algunos lo atribuyen al hecho de un gran número de nuestras librerías y herramientas le asignamos el prefijo "My" por costumbre. Sin embargo la hija de "Monty" es además llamada My así que ¿cual de las dos razones da el nombre a MySQL?, eso es aún un misterio.

Bien después de haber hecho esta breve revisión historia entremos más a detalle de lo que es MySQL.

MySQL es un sistema de gestión de base de datos. Y ¿qué es una base de datos?, es una colección estructurada de datos. Puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa.

MySQL es un gestor de bases de Datos SQL con implementaciones cliente-servidor que consta precisamente de un servidor y diferentes clientes (programas y librerías), podemos agregar, acceder y procesar datos grabados en una base de datos.

Actualmente el gestor de una base de datos juega un rol muy importante en la informática, como única unidad o como parte de una aplicación.

Es un sistema de gestión de base de datos relacional. El modelo relacional se caracteriza a muy grandes rasgos para disponer que toda la información debe estar contenida en tablas, y las relaciones

entre datos deben estar representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

MySQL es un software de código abierto, esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades.

MySQL usa el GPL (Licencia Pública General) para definir que podemos y que no podemos hacer con el software en diferentes situaciones. Entre otras cuestiones, esta licencia aclara que no cuesta dinero a menos que lo incluyamos en un software comercial y tenemos el código fuente.

### 3.4.2 ¿POR QUÉ UTILIZA MYSQL?

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría). Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace que MySQL sea altamente conveniente para acceder a bases de datos en internet.

En este proyecto de tesis se utilizó la versión 3.23.41

### 3.4.3 PRINCIPALES CARACTERÍSTICAS

- El principal objetivo es dar velocidad y robustez
- Escrito en C y C++, testado con GCC 2.7.2.1. usa GNU autoconfort para portabilidad.
- Clientes C, c++, Java, Perl, Tcl, PHP.
- Multiproceso, es decir se puede usar en varios CPUs, si está disponibles.
- Puede trabajar en distintas plataformas y S.O distintos.
- Sistema de contraseñas y privilegios, muy flexible y segura.

- Todas las palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.
- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad (Isamchk) para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859\_1.
- Los clientes usan TCP o UNIX socket para conectarse al servidor.
- El servidor soporta mensajes de error de distintas lenguas.
- Todos los comandos tienen help o ?.....para las ayudas.
- Diversos tipo del columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, caracter, fechas, enumerados etc.
- ODBC para windows 95 (con fuentes), se puede utilizar ACCES para conectar con el servidor.

## 3.5 LENGUAJES DE PROGRAMACIÓN

### 3.5.1 INTRODUCCIÓN

En informática, cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por una computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción.<sup>10</sup>

### 3.5.2 BREVE HISTORIA DE LOS LENGUAJES

En los años 50 se realizaron varios compiladores primitivos y fue en 1957 cuando apareció el primer compilador de FORTRAN. El compilador de FORTRAN (FORMula TRANslator) estaba diseñado para

---

<sup>10</sup> <http://civilizacionweb.com/ijh>

traducir a lenguaje máquina expresiones y operaciones matemática, e incluso permitía la manipulación de matrices. <sup>11</sup>

La aparición del FORTRAN fue un gran cambio para los programadores que no todos aceptaron de buen grado. No les gustaba que sus programas fueran tratados como meros datos, y argumentaban que el código máquina generado por el compilador nunca podría ser tan eficiente como el escrito por ellos directamente.

Esto no era generalmente así, puesto que el FORTRAN no fue diseñado pensando en crear un lenguaje bien estructurado sino pensando en crear un traductor de expresiones aritméticas a código máquina muy eficiente. Por ello, el diseño lógico de la Computadora IBM 704 para el que fue creado casi puede deducirse del lenguaje FORTRAN.

En diferentes versiones, cada vez más estructuradas, el lenguaje FORTRAN se ha utilizado extensivamente desde que apareció hasta hoy en día, y puede considerarse el lenguaje estándar del cálculo científico. Unos años después de aparecer el FORTRAN apareció el lenguaje ALGOL 60 (Algorithm Language), que fue diseñado para ser independiente de la computadora con una gramática bien definida.

También de aquella época es el COBOL (Common Business Oriented Language) que se diseñó para las manipulaciones de datos normales en aplicaciones de negocios y con un uso mayor del lenguaje inglés en sus frases. Las versiones modernas del COBOL siguen usándose en la actualidad y es el lenguaje estándar en aplicaciones informáticas bancarias.

Desde entonces han aparecido diversos lenguajes de alto nivel entre los que podemos mencionar el BASIC (Beginners All-purpose Symbolic Instructional Code), PL/I , APL, PASCAL, ADA, MODULA , C , RPG, PROLOG, LISP, ... etc.

---

<sup>11</sup> <http://www.ii.uam.es/vicente/apuntes/node11.html>

Alguno de estos lenguajes han sido diseñados para un tipo concreto de aplicaciones. Por ejemplo, el ADA para aplicaciones relacionadas con defensa, o el RPG para transacciones usuales en los bancos.

La evolución de los lenguajes de programación ha estado guiada por la evolución de:

- Las Computadoras y sus sistemas operativos.
- Las aplicaciones.
- Los métodos de programación.
- Los fundamentos teóricos.
- La importancia dada a la estandarización.

Podemos resumir la evolución de los lenguajes de programación en la siguiente tabla:

periodo	Influencias	Lenguajes
1950 - 55	Computadoras primitivas	Lenguajes ensamblador
		Lenguajes experimentales
		de alto nivel
1956 - 60	Computadoras pequeñas,	FORTRAN
	caros y lentos	ALGOL 58 y 60
	Cintas magnéticas	COBOL
	Compiladores e interpretes	LISP
	Optimización del código	
1961 - 65	Ord. grandes y caros	FORTRAN IV
	Discos Magnéticos	COBOL 61 Extendido
	Sistemas operativos	ALGOL 60 Revisado
	Leng. de propósito general	SNOBOL

		APL ( como notación sólo)
1966 - 70	Computadoras de diferentes	PL/I
	tamaños, velocidades, costes	FORTRAN 66 (estandard)
	Sistemas de almacenamiento	COBOL 65 (estandard)
	masivo de datos (caros)	ALGOL 68
	S.O. multitarea e	SNOBOL4
	interactivos	SIMULA 67
	Compil. con optimización	BASIC
	Leng. estandard ,	APL/360
	flexibles y generales	
1971 - 75	Micro Computadoras	
	Sistemas de almacenamiento	PASCAL
	masivo de datos pequeños	COBOL 74
	y baratos	PL /I
	Progr. estructurada	
	Ingeniería del software	
	Leng. sencillos	
1976 - 80	Ord. baratos y potentes	ADA
	Sistemas distribuidos	FORTRAN 77
	Prog. tiempo-real	PROLOG
	Prog. interactiva	C
	Abstracción de datos	
	Prog. con fiabilidad	

y fácil mantenimiento
-----------------------

### 3.5.3 TIPOS DE LENGUAJES

Desde un punto de vista más general los lenguajes se pueden clasificar en **lenguajes de procedimiento o declarativos**. En los primeros, con el lenguaje se especifica paso a paso el procedimiento que ha de realizar la PC para procesar la información, mientras que en los segundos se declaran hechos que han de dirigir las respuestas de la PC.

Estableciendo relaciones lógicas que determinan una base de verdades con las que han de ser coherentes las respuestas del programa. Existe una diferencia grande en la programación con un lenguaje u otro según sea **interpretado** o **compilado**, si bien esta distinción puede no ser inherente al lenguaje sino a su puesta en práctica en un determinado computador.<sup>12</sup>

Un lenguaje es **interpretado** cuando la transformación de las instrucciones de alto nivel a lenguaje máquina se realiza sentencia a sentencia según se van ejecutando.

Un lenguaje es **compilado** cuando esta transformación se realiza en bloque antes de que ninguna instrucción sea ejecutada. En un lenguaje interpretado la puesta a punto de un programa ha de realizarse secuencialmente puesto que las distintas partes del programa no se pueden verificar hasta que entran en ejecución. En el caso más común de lenguajes compilados, son varios los subprocesos implicados en la transformación del código de alto nivel en instrucciones ejecutables por el CPU.

A su vez el compilador realiza varias tareas:

- verificación de la sintaxis

<sup>12</sup> <http://www.ii.uam.es/vicente/apuntes/node11.html>

- transformación a código máquina
- optimización

Si un programa es incorrecto sintácticamente, el compilador detectará el error y lo comunicará con un mensaje relacionado con la regla del lenguaje que se ha incumplido.

Normalmente, estos son los errores más comunes y más fáciles de detectar. El montador (*Linker*) unifica el código con el proveniente de otros subprogramas con el que se intercambian datos. Para ello realiza una lista de los datos que comparten todos los programas y asigna las direcciones comunes donde cada uno deberá procesar esos datos.

Esta unificación de direcciones será imposible si algún procedimiento supuestamente existente en otro subprograma no aparece o aparece de un modo no unívoco. También será causa de error que algún dato compartido por subprogramas esté declarado de modo distinto en cada programa.

Estos errores son poco comunes y fácilmente detectable con los mensajes de error proporcionados por el montador. Sin embargo, los errores más comunes y más tediosos de eliminar son aquellos de programación que dan lugar a sentencias sintácticamente correctas pero que corresponde a acciones distintas a las deseadas. Desgraciadamente sólo se detectan en la ejecución del programa.

En la actualidad existe la posibilidad de utilizar depuradores de programas (llamados en inglés *debuggers* por el origen de los errores en las computadoras primitivos ) que permiten seguir la ejecución paso a paso de un programa aunque se obtenga el código máquina por compilación.

Esta herramienta facilita enormemente la depuración de los programas pues permite conocer o modificar el valor de los datos manipulados en el programa durante la ejecución.

### 3.5.3.1 LENGUAJES DE BAJO NIVEL

Vistos a muy bajo nivel, los microprocesadores procesan exclusivamente señales electrónicas binarias. Dar una instrucción a un microprocesador supone en realidad enviar series de unos y ceros espaciadas en el tiempo de una forma determinada.<sup>13</sup>

Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números e instrucciones para manipularlos. Un modo más fácil de comprender el código máquina es dando a cada instrucción un mnemónico, como por ejemplo STORE, ADD o JUMP. Esta abstracción da como resultado el ensamblador, un lenguaje de muy bajo nivel que es específico de cada microprocesador.

Los lenguajes de bajo nivel permiten crear programas muy rápidos, pero que son a menudo difíciles de aprender. Más importante es el hecho de que los programas escritos en un bajo nivel sean altamente específicos de cada procesador. Si se lleva el programa a otra máquina se debe reescribir el programa desde el principio.

### 3.5.3.2 LENGUAJES DE ALTO NIVEL

Por lo general se piensa que las PC's son máquinas que realizan tareas de cálculos o procesamiento de textos. La descripción anterior es sólo una forma muy esquemática de ver una computadora. Hay un alto nivel de abstracción entre lo que se pide a la computadora y lo que realmente comprende. Existe también una relación compleja entre los lenguajes de alto nivel y el código máquina.

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés. En BASIC, el lenguaje de alto nivel más conocido, los comandos como "IF CONTADOR = 10 THEN STOP" pueden utilizarse para pedir a la computadora que pare si

---

<sup>13</sup> <http://www.ii.uam.es/vicente/apuntes/node11.html>

CONTADOR es igual a 10. Por desgracia para muchas personas esta forma de trabajar es un poco frustrante, dado que a pesar de que las computadoras parecen comprender un lenguaje natural, lo hacen en realidad de una forma rígida y sistemática.

### 3.5.3.3 INTÉRPRETES Y COMPILADORES

La traducción de una serie de instrucciones en lenguaje ensamblador (el código fuente) a un código máquina (o código objeto) no es un proceso muy complicado y se realiza normalmente por un programa especial llamado compilador.<sup>14</sup>

La traducción de un código fuente de alto nivel a un código máquina también se realiza con un compilador, en este caso más complejo, o mediante un intérprete. Un compilador crea una lista de instrucciones de código máquina, el código objeto, basándose en un código fuente.

El código objeto resultante es un programa rápido y listo para funcionar, pero que puede hacer que falle la PC si no está bien diseñado. Los intérpretes, por otro lado, son más lentos que los compiladores ya que no producen un código objeto, sino que recorren el código fuente una línea cada vez. Cada línea se traduce a código máquina y se ejecuta. Cuando la línea se lee por segunda vez, como en el caso de los programas en que se reutilizan partes del código, debe compilarse de nuevo. Aunque este proceso es más lento, es menos susceptible de provocar fallos en la computadora

### 3.5.4 BREVE HISTORIA DE HTML

El lenguaje HTML nace en 1991 de manos de Tim Berners-Lee de CERN como un sistema hipertexto con el único objetivo de servir como medio de transmisión entre físicos como parte de la iniciativa WWW.<sup>15</sup>

---

<sup>14</sup> <http://www.ii.uam.es/vicente/apuntes/node11.html>

En 1993 Dan Connelly escribe el primer el primer DTD (Document Type Definition) de SGML describiendo el lenguaje. En 1994 el sistema había tenido tal aceptación que la especificación se había quedado ya obsoleta.

Por aquel entonces WWW y Mosaic eran casi sinónimos debido a que el browser Mosaic del NCSA (National Center for Supercomputing Applications) era el más extendido por las mejoras que incorporaba. Es cuando nace HTML 2.0 en un draft realizado también por Dan Connelly.

El crecimiento exponencial que comienza a sufrir el sistema lleva a organizar la *Primera Conferencia Internacional del WWW* en Mayo de 1994.

El principal avance de la versión 2.0 de HTML es la incorporación de los llamados forms, formularios que permiten que el usuario cliente envíe información al servidor y ésta sea recogida y procesada allí: Precisamente con este fin, NCSA presenta la especificación del CGI, Common Gateway Interface, versión 1.0 que define un interfaz entre programas ejecutables y el sistema WWW.

Con la incorporación de los forms, aparece por primera vez campos donde el usuario puede escribir, menús desplegables y los denominados radio-buttons (botones de radio) integrados en páginas WWW.

Desde entonces el Lenguaje ha seguido creciendo como algo dinámico como una lengua humana, algo vivo, siendo modificado sobre todo por las personas que lo utilizan. Así, una evolución con el lenguaje suele seguir de una propuesta que es adoptada por algunos clientes (browsers). Con el uso se ve si es eficiente y es adoptada y si es así, finalmente se incorpora al standar.

Con la version 3.0 de HTML surgió un nuevo estándar inducido por los rápidos cambios que ofrecía el líder indiscutible de los navegadores de la época: Netscape. Dado que sus nuevas etiquetas no eran

---

<sup>15</sup> <http://www.cneq.unam.mx/cursos/paginas/06html.html>

incluidas en los estándares de HTML 2 al final fue creado ésta nueva versión de HTML. Pero dado que era muy extenso, los demás navegadores sólo incorporaban pequeñas partes de él, lo que produjo su rechazo.

El HTML 3.2 nació en enero de 1997 de la unión de las anteriores etiquetas con las más nuevas implementadas por los navegadores Netscape y Microsoft Explorer, como tablas, applets de Java y texto alrededor de las imágenes.

El HTML 4.0 fue presentado en julio de 1997 como borrador el cual incluía nuevas etiquetas de tablas, hojas de estilo, frames, JavaScript, VBScript, accesibilidad e internacionalización. Y en Diciembre de 1997 es liberada como recomendación de la World Wide Web Consortium o W3C.

La última revisión por el W3C de HTML es la versión 4.01 y fue liberada el 24 de Diciembre de 1999. El W3C es el organismo de realizar las recomendaciones de los estándares de SGML y entre ellos del HTML.

### 3.5.5 BREVE HISTORIA DE PHP

PHP fue concebido en otoño de 1994 por Rasmus Lerdorf. Las primeras versiones no distribuidas al público fueron usadas en sus páginas web para mantener un control sobre quién consultaba su currículo.<sup>16</sup>

La primera versión disponible para el público a principios de 1995 fue conocida como "Herramientas para páginas web personales" (Personal home page tools). Consistían en un analizador sintáctico muy simple que solo entendía unas cuantas macros y una serie de utilidades comunes en las páginas web de entonces, un libro de visitas, un contador y otras pequeñas cosas.

---

<sup>16</sup> <http://www.php.net>

El analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado como PHP/FI versión 2. FI viene de otro programa que Rasmus había escrito y que procesaba los datos de formularios. Así que combinó las herramientas para páginas web personales, "el interprete de formularios", añadió soporte para mSQL y PFP/FI vio la luz. Creció a una gran velocidad y la gente comenzó a contribuir en el código.

Es difícil dar estadísticas exactas, pero se estima que a finales de 1996 PHP/FI se estaba usando al menos en 15000 páginas web alrededor del mundo. A mediados de 1997 este número había crecido a más de 50 000 y a mediados de 1997 el desarrollo del proyecto sufrió un profundo cambio, dejó de ser un proyecto personal de Rasmus, al cual habían ayudado un grupo de usuarios y se convirtió en un proyecto de grupo mucho más organizado.

El analizador sintético se rescribió desde el principio por Zeev Suraski y Andi Gutmans y este nuevo analizador estableció las bases para PHP versión 3.

Gran cantidad de código de PHP/FI fue portado a PHP3 y otra gran cantidad fue escrito completamente de nuevo.

A finales de 1999, tanto PHP/FI como PHP3 se distribuyeron en un gran número de productos comerciales tales como el servidor web "C2's StrongHold" y Redhat Linux. Una estimación conservativa basada en estadísticas de Netcraft, es que más de 1 000 000 de servidores alrededor del mundo usan PHP.

Para hacernos una idea, este número es mayor que el número de servidores que utilizan el "Netscape's Enterprise server" en internet. El trabajo de la actual generación de PHP, utiliza el potente motor de scripts zend para proporcionar altas prestaciones, así como soporta otros servidores web, además de apache, que corren PHP como módulo nativo.

### 3.5.6 ¿QUÉ ES PHP?

PHP (acrónimo de "PHP": Hypertext Preprocessor) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

PHP no es lo mismo que un script CGI escrito en otro lenguaje de programación como Perl o C. En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP introducido en el mismo, que producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP<sup>17</sup>.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones.... No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.



<sup>17</sup> <http://www.php.net>

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

### 3.5.7 QUÉ SE PUEDE HACER CON PHP

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos o mandar o recibir cookies. Quizás la característica más potente y destacada de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP.

PHP soporta las siguientes bases de datos: Adabas D, Ingres, Oracle Dbase, Interbase, Postgre Sql, Empress, Frontbase, Solid, Filepro, Msql, Sybase, IBM DB2, MySql, Velocis, Informix, ODBC, Unis DBM.

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos.

### 3.5.8 APACHE WEB SERVER

Uno de los servicios más demandados actualmente es el de internet especialmente el de poder despachar páginas web, este servicio es conocido como servidor web. Este debe proveer además de poder mandar la información, poder generar páginas dinámicas para que el servidor que puedan arrojar datos procesados que fueron el resultado de datos introducidos por el usuario<sup>18</sup>.

El servidor web se encarga de publicar la información hacia el internet o hacia una Intranet. El http es el medio de acceso usado por los navegadores para proveer el acceso a un sitio web, el trabajo del

---

<sup>18</sup> <http://www.apache.org> y <http://www.apache-ssl.org>

servidor consiste en andar el contenido en respuesta a la peticiones de un URL, el contenido de un sitio web puede ser estático o generado dinámicamente, aunque las páginas estáticas son creadas por el HTML, las interfaces comunes de programación o más comúnmente llamadas CGI's proveen contenidos dinámicos en el servidor web.

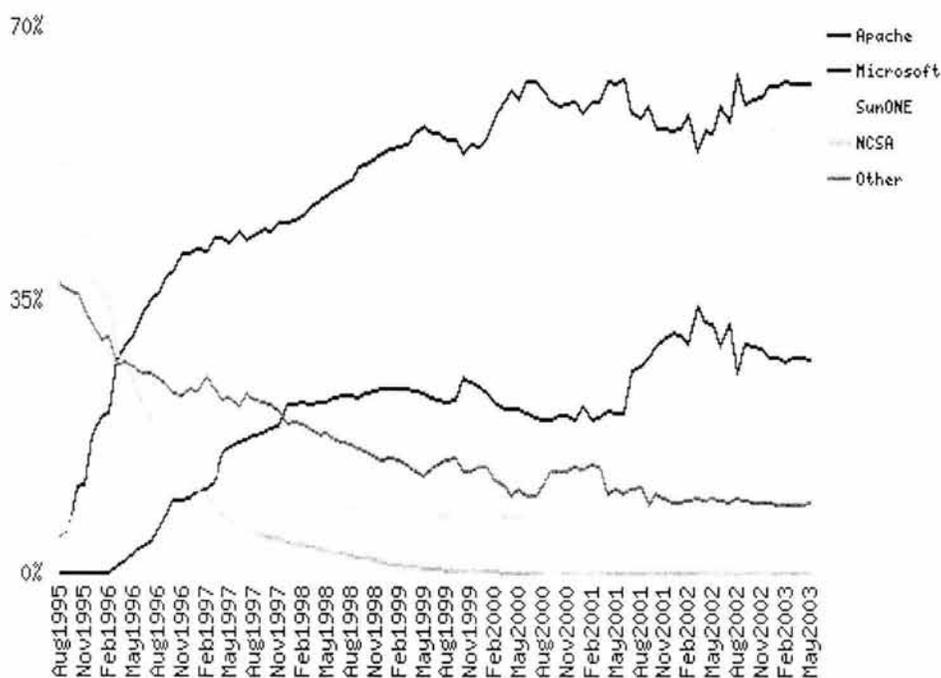
Este tipo de servicios ha crecido de manera muy importante desde el surgimiento de las intranets. Estas son redes locales que permiten acceder información por la misma metodología que en internet, solo que debe estar protegida para accesos fuera de esa red.

GNU/Linux ofrece soporte para varios servidores web dentro de los cuales el más famoso y robusto se llama Apache. Este servidor web viene incluido en la mayoría de las distribuciones gratuitamente y con una interfase fácil de instalación.

Este servidor http o servidor web es actualmente más usado.

Por sondeo y estadísticas se concluyó que aproximadamente el 53% de los servidores web en el mundo son Apache, estas cifras son impresionantes ya que empresas como netscape enterprise server o el internet information server de Microsoft no tienen una penetración así, el proyecto apache permite la utilización de módulos para aumentar su funcionalidad sin tener que recompilar el producto.

El netcraft (Web Server Survey ), es un software que inspecciona servidores Web a continuación se muestra una estadística de los servidores Web más utilizados



Servidores que se utilizan para el Desarrollo Web.<sup>19</sup>

Servidor	Abril 2003	Porcentaje	Mayo 2003	Porcentaje	Cambio
Apache	25090844	62.57	25290478	62.53	-0.04
Microsoft	11007434	27.45	10989025	27.17	-0.28
Zeus	789324	1.97	780936	1.93	-0.04
SunOne	415999	1.04	420398	1.04	0.00

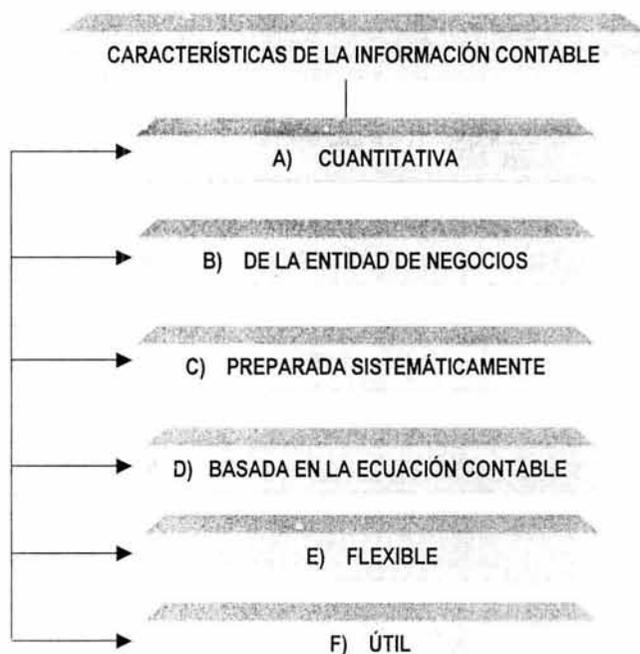
<sup>19</sup> <http://www.netcarft.com>

## CAPITULO 4. CONCEPTOS GENERALES

### 4.1 DEFINICIÓN DE CONTABILIDAD.

La contabilidad..... es una técnica que se utiliza para producir sistemática y estructuradamente información cuantitativa expresada en unidades monetarias de las transacciones que realiza una entidad económica y de ciertos eventos económicos identificables y cuantificables que la afectan, con objeto de facilitar a los diversos interesados, el tomar decisiones en relación con dicha entidad económica.

Las características de la información contable que la distinguen de otra clase de informes de negocios, son las siguientes.<sup>1</sup>



<sup>1</sup> La información Contable y la Información financiera

- A) Es información principalmente cuantitativa.
- B) Es información sobre las entidades económicas de negocios.
- C) Es información obtenida sistemática y estructuradamente a través de sistemas de teneduría y control.
- D) Se encuentra basada en la ecuación de activos, pasivos y capital.
- E) Es información sintética y analítica que puede combinarse desde diversos puntos de vista, es decir flexible.
- F) Debe ser útil para la toma de decisiones de diversos usuarios interesados en la entidad económica.

#### 4.1.1 CARACTERÍSTICAS DE LA INFORMACIÓN CONTABLE

El esquema de la teoría de la partida doble supone la existencia previa de una estructura financiera de activos, pasivos y capital, la cual es alterada por las transacciones y ciertos eventos económicos que la afectan.

El objetivo de la contabilidad es capturar esas alteraciones y mantener información actualizada sobre el estatus de la estructura financiera, para lo cual se debe crear una base conceptual suficiente con criterios básicos sobre los que va a operar el sistema de contabilidad y obtenerse los informes contables.<sup>2</sup>

---

<sup>2</sup> La información Contable y la Información Financiera

Los estados financieros, como parte básica de la información financiera, son conjuntos de elementos de información que se combinan de manera determinada para proporcionar un mensaje concreto, (por ejemplo el balance general y el estado de resultados) de esta forma se podría definir que los criterios básicos de la contabilidad que adecúan los procedimientos a los propósitos que se quieren lograr o conseguir con la elaboración y presentación de informes contables se refieren a:

1. **Definiciones.** Cómo se deben clasificar las transacciones y otros eventos en los elementos de los estados financieros.
2. **Métodos de Valuación.** Cómo se deben valorar y cuantificar los efectos de las transacciones y otros eventos en los elementos clasificados.
3. **Métodos de reconocimiento.** Cuándo se deben reconocer los efectos de las transacciones y otros eventos en los elementos informativos.
4. **Formas de Presentación y Revelación.** Cómo se presentan los elementos en los estados financieros y que información debe acompañarlos.

El universo de interesados en información contable es extenso y diverso en cuanto a sus requerimientos particulares de información. Por mencionar algunos tenemos a los acreedores, accionistas o propietarios, administradores y funcionarios de la entidad, clientes, gobierno fiscal, analistas financieros y sociedad en general.

Dado que la contabilidad en general es el esfuerzo de satisfacer racionalmente las demandas de información contable con propósitos diferentes de usuarios distintos. Y que además es muy extensa su aplicación, para la presente tesis solo trataremos un tema: la información o contabilidad financiera.

## 4.2 LA INFORMACIÓN FINANCIERA

La información financiera, llamada también información externa, se encuentra dirigida a los usuarios externos del negocio que no están involucrados en la administración. Pero pretende satisfacer las demandas de rendición de cuentas por parte del administrador, las demandas de información para la toma de decisiones de crédito e inversión de los acreedores y accionistas; trabajadores, gobierno en sus funciones de evaluación y promoción económica, requerimientos de las bolsas de valores, analistas, en fin, a cualquier interesado que sea un usuario externo a la entidad que informa.

La información financiera es información sintética que muestra los aspectos más generales de la entidad de negocios, su situación financiera, sus resultados, sus flujos de efectivo y los riesgos financieros que soporta, entre otras cosas.

### 4.2.1 DEFINICIÓN DE CONTABILIDAD FINANCIERA

Los informes contables son una clase de informes de negocios y la información financiera es una subclase, se puede proponer entonces una definición de contabilidad financiera como una técnica que se utiliza para producir de acuerdo a principios de contabilidad generalmente aceptados, sistemática y estructuradamente, información financiera de las transacciones que realiza una entidad económica y de ciertos eventos económicos identificables y cuantificables que la afectan, con objeto de ser útil para propósitos generales de los usuarios externos a la misma<sup>3</sup>.

### 4.2.2 UTILIDAD DE LA INFORMACIÓN FINANCIERA

La utilidad de la información financiera de las entidades, para propósitos generales de usuarios externos, es una necesidad concreta de las sociedades económicas y constituye la causa final que justifica la existencia de la práctica de la contabilidad financiera.

---

<sup>3</sup> Teoría de la Contabilidad Financiera

## Estructura de conceptos de la Contabilidad Financiera



Los objetivos de la información financiera pretenden dar respuesta a estas preguntas:

¿Quiénes son los usuarios externos?

¿Para qué utilizan la información financiera?

¿Cuál es la información financiera?

La información financiera debe ser útil para los propósitos señalados en los objetivos de la información financiera. Para lograr dicha utilidad la información debe cubrir de manera equilibrada los requisitos de calidad siguientes:

- La información debe ser **relevante**.
- La información debe ser **confiable**.

Los supuestos básicos son proposiciones generales que constituyen ingredientes de las reglas de contabilidad. Unos se refieren a las condiciones impuestas por el mismo ambiente de negocios, y otros, son relativos a la mejor manera de llevar a cabo los procedimientos contables.

Los objetivos y requisitos de calidad establecen el panorama que deben cumplir las reglas o normas de contabilidad y los supuestos se refieren más bien al contenido relevante de las reglas de contabilidad.

Por mencionar algunos supuestos o principios; Entidad, Negocio en Marcha, Período Contable, Realización, Valor Histórico Original, Dualidad Económica, Revelación Suficiente, Consistencia.

Los criterios de aplicación también son herramientas para elaborar reglas de contabilidad, sin embargo son combinaciones de los supuestos básicos, limitaciones o extensiones de los mismos, que se encuentran también en la frontera de las reglas particulares de contabilidad, entre los cuales podemos mencionar algunos:

Criterio Prudencial, Costo – Beneficio, Reconocimiento Contable, Reconocimiento de Ingresos, Identificación de Costos y Gastos Contra Ingresos, Reconocimiento de Activos, Reconocimiento de Pasivos, Concepto de lo Devengado, compensación, Criterio de Utilidad, Criterio de Mantenimiento de Capital, Criterio de utilidad Integral.

Y por último las normas de contabilidad financiera establecen las normas concretas que deben observar los procedimientos contables en sus distintas etapas de definición, cuantificación, reconocimiento, presentación y revelaciones, de los elementos que integran los estados financieros.

### 4.3 INFORMES CONTABLES

Ahora que tenemos una idea más objetiva de lo que es la obtención de la información financiera es necesario ver reflejados estos resultados en los informes contables.

Los informes contables son el producto terminado del proceso contable, sin embargo se considera ventajoso empezar el estudio de la contabilidad precisamente considerando su forma y contenido. **Los informes contables sirven para proporcionar a la administración y a los terceros interesados, una visión concreta acerca de la rentabilidad y posición financiera de un negocio.**

Se debe tener presente que para poder obtener estos informes contables es necesario introducir datos financieros al sistema de información financiera, y estos datos representan cambios ocurridos en la empresa.

Si no los ha habido, no se necesitan elementos de entrada; los que ya están contenidos en el sistema representan la situación de la empresa antes de que presentara cambios de índole financiera. Cuando se presenten, es preciso determinarlos y registrarlos, para poder darles entrada al sistema.

Los datos financieros provienen de dos fuentes: las operaciones externas y las que ocurren en el seno de la empresa, originando los cambios en ellas, y son, por consiguiente, fuente de datos financieros.

Los actos financieros que se efectúan entre las empresas y otras personas o entidades ajenas a ellas, se denominan operaciones externas. Los cambios que se originan en operaciones externas deben

interpretarse en función financiera; y los datos financieros que se especifican así se registran, para poder introducirlos al sistema de información.

En las operaciones externas intervienen muchas clases de modificaciones financieras: la compra de terrenos y edificios, el pago de pasivos y los ingresos recibidos. El análisis de las operaciones externas proporciona los datos para reflejar los cambios, en el sistema de información. Una vez que se han adquirido los recursos, el uso de ellos por parte de la empresa engendra cambios que es preciso reflejar en la introducción de datos en el sistema de información.

Los tipos de las modificaciones que pueden incurrir son numerosos ; por ejemplo, cuando el fabricante de armarios compra madera, es fácil ver que esa operación externa aumenta la existencia de madera disponible para fabricar armarios y que disminuye la existencia de efectivo (si la pagó al recibirla).

Los datos de estos cambios se introducirán al sistema, cuando la madera se traslada del almacén a la fábrica y es transformada, convirtiéndose en armarios, ha ocurrido otro cambio, aunque la madera no ha sido trasladada a ninguna otra entidad o persona.

Los datos reflejo de este cambio deben introducirse al sistema de información, por que es evidente que la empresa posee armarios fabricados y no madera en bruto. Estos datos corresponden a operaciones internas.

Los datos que representan cambios ocurridos en el seno de la empresa pueden registrarse como introducciones en la fecha en que acontecen; o pueden resumirse o capturarse al final del periodo financiero; y los cambios de todo el período pueden registrarse en un solo dato de introducción.

#### **4.3.1 ESPECIFICACIÓN DE INFORMES EN BASE A LOS DATOS REQUERIDOS**

En todos los sistemas de información pueden registrarse los datos de las operaciones externas solamente si alguien ha determinado específicamente las clases de datos que sea forzoso registrar en

cada tipo de operación. Los datos específicos para cada tipo de operaciones externas dependen de la información que rinda el sistema.

La cuestión de que informes se requieren acerca de un acto, nos hace volver a hablar de las decisiones concretas basadas en ellos. La información financiera proveniente de un sistema contable que se usará principalmente para elaborar modelos financieros de la empresa en forma de estados (balance general y estado de resultados por mencionar algunos), gracias a los cuales se dispone de informes para hacer un análisis. Se estructuran los informes de manera que contengan información para numerosos tipos diferentes de decisiones, haciéndose, sin embargo, hincapié en la destinada a los inversionistas.

La información diaria y periódica para la gerencia forzosamente deberán estar contenidas en la información financiera requerida.

#### **4.3.2 INFORMACIÓN ACERCA DE LOS ACTIVOS**

Para que el inversionista pueda analizar bien la estabilidad financiera y la liquidez (disponibilidad del efectivo) de la empresa, es preciso conocer la magnitud de los fondos asignados a los diferentes tipos de activos (o sea inmovilizados en ellos).

Esta información es necesaria para formular planes y proyectos acerca de las utilidades netas o resultados de la empresa.

Es necesario conocer el monto de lo vendido a crédito a cada cliente y el cobrado a cada uno, para poder mandarle las facturas que lo hagan pagar a la empresa. Las fechas en las que se hacen las ventas son indispensables para este propósito y para que la gerencia pueda formular sus decisiones acerca de las gestiones de cobro.

#### **4.3.3 INFORMACIÓN ACERCA DE LOS PASIVOS.**

Es indispensable que los accionistas conozcan la magnitud de los pasivos a corto y a largo plazo(cantidades que debe la empresa o sea créditos a su cargo), porque revela la fuente de un importe concreto del capital de la empresa e indica el monto de los derechos a su cargo, que ha de satisfacer.

Las fechas de las compras a crédito y las condiciones crediticias concebidas por acreedores específicos son importantes para asegurarse de pagar a los acreedores con tiempo suficiente para que puedan aprovecharse todos los descuentos por pronto pago, antes de que venzan los adeudos

#### **4.3.4 INFORMACIÓN ACERCA DEL CAPITAL CONTABLE**

Ésta muestra el monto del capital invertido y las ganancias de la empresa. Para casi todos los propósitos basta que esta información sea reflejada en modo resumen.

#### **4.3.5 INFORMACIÓN ACERCA DE LA UTILIDADES NETAS**

Las utilidades netas son el ingreso neto, o acrecentamiento de recursos ganados por la empresa. Se calcula como el excedente de la entrada bruta de recursos ganados (ingresos) sobre los recursos consumidos para obtener esos ingresos (gastos).

#### **4.3.6 INFORMACIÓN ACERCA DE LOS INGRESOS**

Al analizar la forma en que la empresa obtiene sus ganancias o utilidades netas, es muy importante para los accionistas tener informes acerca de los ingresos, que parte de estos se empleó para diversas clases de gastos y la parte que queda como utilidad neta.

#### 4.3.7 INFORMACIÓN ACERCA DE LOS GASTOS

Esto se refiere al consumo de activos (recursos) de la empresa en sus operaciones, es resultado directo de las decisiones administrativas y de la eficiencia de los gerentes y otros de sus empleados. Los accionistas utilizan ésta clase de información para estimar la eficiencia relativa de la empresa y cuentan con su ayuda en el pronóstico de lo que ésta realizará en el futuro.

Estos ejemplos nos sirven para darnos cuenta de cuan importante es obtener información verídica, confiable y oportuna a través de los datos que son introducidos al sistema de información. Una vez que se han revisado estos ejemplos acerca de la importancia de la información financiera, es necesario definir que la aplicación de la contabilidad es sumamente extensa y por lo tanto sería imposible describir ahora toda la información acerca de las empresas que necesiten los numerosos usuarios de ella.

Sin embargo cuando se proyecta el sistema de información financiera, es indispensable concretar cuáles son las necesidades informativas, ésta manifestación detallada de los informes requeridos es la guía para especificar los datos por registrar, de diferentes tipos de operaciones externas y de otros actos. También sirve de orientación para clasificar esos datos así como su acumulación y resumen para crear la información requerida. Para la presente tesis solo haremos referencia a dos estados financieros : el balance general y el estado de resultados. Por lo que a continuación se dará una breve explicación acerca de su contenido e importancia.

#### 4.4 GENERALIDADES SOBRE LOS ESTADOS FINANCIEROS

Los estados financieros frecuentemente se clasifican como principales y secundarios. Pero ¿qué es lo determinante para que un estado financiero se considere principal?. La importancia de la información revelada por un estado financiero debería ser la base para clasificar un estado financiero como principal o secundario. Sin embargo, se consideran como principales aquellos estados financieros sobre los

cuales dictaminan los auditores. Por lo general los dictámenes de contadores públicos hacen referencia al Balance General y al Estado de Resultados, pudiendo ser éstos Estados Consolidados.

#### 4.4.1 EL BALANCE GENERAL

El balance general refleja "razonablemente" la situación financiera de una empresa, conforme a principios de contabilidad generalmente aceptados. Ha habido serias dudas con respecto a si el balance general en realidad presenta la situación financiera de una empresa o no.

Con todos los criterios contables enfocados a la determinación de utilidades más realistas, utilizando el método U.E.P.S., para determinar el costo de los inventarios y métodos de depreciación acelerada con relación a los activos fijos, en un intento para contrarrestar los efectos inflacionarios sobre la determinación de utilidades.

Como consecuencia de lo anterior, tan sólo podríamos atrevernos a definir el balance general como aquel estado financiero que muestra los activos, pasivos y el capital contable de una empresa<sup>4</sup>.

#### 4.4.2 FORMAS DE PRESENTACIÓN

A la forma tradicional de presentación del Balance General se le denomina "en forma de cuenta". Presenta todos los activos en el lado izquierdo y a la derecha se presentan los pasivos y el capital contable.

Otra presentación del Balance general se le denomina "en forma de reporte". Se presentan las partidas para un periodo determinado en forma vertical; primero los activos habiéndose de obtener la suma total de ellos.

---

<sup>4</sup> Análisis e interpretación de la Información financiera

Esta forma de presentación es muy útil cuando se deseen comparar balances generales de varios periodos, o sea, cuando se trate de balances comparativos, siendo particularmente útil para efectos del análisis de estados financieros. En este proyecto el balance se emitirá en forma de reporte.

#### 4.4.3 IMPORTANCIA DEL ESTADO DE RESULTADOS

Tradicionalmente al estado de resultados se le conocía con el nombre de Estado de Pérdidas y ganancias. En un principio se pensaba que el Estado de resultados era solo un anexo del Balance General; en donde se desglosaba la partida de utilidad del ejercicio que se incluía en el Balance General.

Algunos contadores han llegado a afirmar que el estado de resultados es más importante que el balance general debido a que la determinación de la utilidades periódicas son de sumo interés para muchas personas físicas o morales.

Con base en las utilidades se pagan los impuestos y se reparten las utilidades a los trabajadores. Las utilidades sirven de base para el reparto de dividendos, e inclusive para los financiamientos a mediano o largo plazo, se espera que las empresas obtengan flujos provenientes de las operaciones para poder liquidar sus adeudos.

#### 4.4.4 FORMAS DE PRESENTACIÓN

El estado de resultados convencional presenta las ventas brutas en primer término, a las cuales se deducen las devoluciones, rebajas y descuentos sobre ventas para obtener las ventas netas. A estas se les resta el costo de ventas para así determinar la utilidad bruta.

Se deducen luego los gastos de operación para obtener la utilidad de operación. Los gastos y productos financieros se restan o se suman a continuación para determinar la utilidades después de financiamiento.

Las utilidades antes de impuestos sobre la renta y reparto de utilidades a los trabajadores se obtienen incluyendo mediante sumas y restas las ganancias y pérdidas extraordinarias. Por último se deducen el impuesto sobre la renta y el reparto de utilidades a los trabajadores para así determinar la cifra de utilidad neta.

#### 4.5 OBJETIVOS DE LOS ESTADOS FINANCIEROS

Registrar las operaciones de un negocio con el solo fin de registrar no tiene sentido alguno. La contabilidad no es un fin en sí misma. La contabilidad se justifica en cuanto sirve a un sistema de información.

El objetivo básico de los estados financieros es el de proporcionar información que sea útil para la toma de decisiones, satisfaciendo así las necesidades de información tanto del sector privado como del sector público.

Puesto que las personas que están dentro de una organización tienen acceso a todos los registros contables, si así lo desean, serán las necesidades de información de los usuarios externos las que se deberán tener en mente para su satisfacción. A continuación se mencionan algunos de los objetivos de los estados financieros<sup>5</sup>.

1. Satisfacer las necesidades de información de aquellas personas que tengan menos posibilidad de obtener información y que dependan de los estados financieros como principal fuente de ésta acerca de las actividades económicas de la empresa.

---

<sup>5</sup> MEIGS, MOSICH, JOHNSON, KELLER Contabilidad Intermedia

2. Proporcionar a los inversionistas y acreedores información útil que les permita predecir, comparar y evaluar los flujos potenciales de efectivo en cuanto concierne al monto de dichos flujos, su oportunidad o fechas en que se hayan de obtener e incertidumbre con respecto a su obtención.
3. Proporcionar información útil para evaluar la capacidad de administración para utilizar con eficacia los recursos de la empresa, alcanzando así la meta primordial de la empresa.
4. Proporcionar información acerca de las transacciones y demás eventos que sirvan para predecir, comparar y evaluar la capacidad generadora de utilidades de la empresa.
5. Presentar un estado de la posición financiera de la empresa, el cuál revelará información concerniente a las transacciones de la empresa y demás eventos que constituyen una parte de ciclos de utilidades no concluidos. Será necesario que se presenten los valores actuales cuando estos difieran marcadamente de los costos históricos. Los activos y los pasivos deberán agruparse o clasificarse conforme a su grado de incertidumbre con respecto a su importe o fecha de realización o liquidación.
6. Presentar un estado de utilidades periódicas que sirva para predecir, compara y evaluar la capacidad generadora de utilidades de la empresa.
7. Proporcionar información útil para el proceso de predicción. Se deberán presentar pronósticos financieros que faciliten la contabilidad de las predicciones de los usuarios.

Basándose en estos objetivos, los usuarios de los estados financieros deberán predecir, comparar y evaluar los resultados de tales decisiones. Es de desearse que los estados financieros incluyan información tanto objetiva como subjetiva, pero deberá distinguirse con toda precisión una de otra.

Dentro de esta toma de decisiones encontramos a las razones financieras que ayudan a los usuarios de los estados financieros a medir la solvencia, la rentabilidad y la estabilidad de la empresa.

Para la presente tesis sólo se mencionan aquellas que se han clasificado como las más importantes. Una razón es sencillamente el cociente de la división de un número por otro, es decir, se calcula dividiendo el número básico entre la otra cifra.

Se concibe fácilmente que con los datos de los estados financieros pueden calcularse docenas de razones, pero generalmente sólo unas cuantas de ellas son útiles para resolver una situación dada; de esta manera no tienen caso buscar la solución de un problema computando inicialmente todas las razones financieras definidas sino que primero deben definirse cuales razones son significativas en el tipo de investigación que se haga y entonces calcular únicamente las que se consideren útiles.

#### 4.6 RAZONES FINANCIERAS

A) Razón Circulante. La razón circulante sirve para determinar grosso modo la capacidad de pago de una empresa. Se determina dividiendo los activos circulantes entre los pasivos circulantes. Por ejemplo la razón circulante de la compañía "A" se calculará de la siguiente manera:

$$\frac{\text{Activos Circulantes} : \$20\,000}{\text{Pasivos Circulantes} : \$10\,000} = 2$$

Después de efectuada tal división, la relación se expresa de varias formas: los activos circulantes de la compañía "A" equivalen a dos veces sus pasivos circulantes; la compañía "A" tiene \$2 de activos circulantes por cada \$1 de pasivos circulantes; o simplemente, la razón circulante de la compañía "A" es de 2 a 1.

La razón circulante es la relación entre los activos y los pasivos circulantes, expresada de una manera matemática. Una elevada razón circulante refleja una proporción alta entre activos y pasivos circulantes.

Entre más alta sea la razón, más líquida será la posición financiera de la compañía, a corto plazo; esto es, tendrá mayor capacidad para cubrir sus compromisos a corto plazo.<sup>6</sup>

- B) Razón de la Prueba del Ácido. Para constatar uno de los factores determinantes para juzgar la adecuación de la razón circulante, "la integración de las partidas del activo circulante", se utiliza la razón de liquidez, que es la razón de los "activos líquidos" a "pasivos circulantes".

Los activos líquidos los constituyen el efectivo, los documentos por cobrar, las cuentas por cobrar y los valores negociables. Constituyen los activos circulantes que pueden convertirse con suma facilidad en efectivo.

La prueba del ácido se podrá calcular dividiendo la suma de los activos líquidos entre el total de pasivos circulantes. Se diferencia de la razón circulante tan sólo en lo que concierne al numerador de la razón de ésta.

- C) Rotación de Cuentas por Cobrar. La rotación de cuentas por cobrar se calcula dividiendo las ventas netas del año entre el saldo de las cuentas por cobrar. La rotación de cuentas por cobrar constituye un indicador de la rapidez con la cual una compañía cobra sus cuentas. La rotación de Cuentas por Cobrar se calcula de la siguiente manera:

a) Ventas Netas para el año: . . . . . \$960000

b) Saldo de Cuentas por Cobrar: . . . . . 68000

Número de veces de rotación: . . . . . 14.1

<sup>6</sup> Principios Fundamentales de contabilidad

## CONCLUSIONES

El sistema desarrollado para el presente proyecto de tesis, fue a través de las herramientas de libre distribución, las cuales se han mencionado con anterioridad, de las que podemos decir que son muy accesibles y sobre todo nos dan otro tipo de panorama, como el que se pueda estar trabajando en ambiente Internet, ya que la tendencia actual del desarrollo de sistemas va enfocado hacia ese ambiente.

En cuanto a las herramientas como GNU/Linux es un sistema operativo ideal para el desarrollo de aplicaciones en red que requieren de un funcionamiento robusto dado que se realizan programas que permiten el acceso a los datos de la organización, datos que son completamente vitales para el buen funcionamiento de la misma.

Al utilizar las herramientas de libre distribución para este proyecto me permitió el realizarlo sin ningún costo y comprobar al mismo tiempo que se pueden crear sistemas de calidad utilizando una gran variedad de estas herramientas.

Dentro de esta gran variedad está el servidor Apache, otra de las herramientas utilizadas reconociendo también que es servidor web por excelencia. Ha sido uno de los mayores éxitos del software libre y su supremacía entre los servidores web se demuestra a través de su desarrollo continuo y su portabilidad le han llevado plataformas como Windows NT, Windows 95 y OS/MAC, a parte de casi todos los sistemas UNIX.

Una de las principales características de Apache es su extensibilidad basada en una gran modularidad del código fuente, facilitando enormemente la programación de aplicaciones en el lado del servidor, especialmente en el campo de acceso a bases de datos.

En el transcurso del desarrollo de este proyecto y utilizar PHP, me doy cuenta de que tiene múltiples ventajas frente a otros lenguajes de programación que necesariamente harán que este lenguaje se imponga como una alternativa para el desarrollo de todo tipo de aplicaciones.

Por mencionar alguna se puede decir que se ejecuta a través de una interfaz que le resulta muy amigable al usuario y no es necesario instalar ningún software adicional en la estación cliente para usar un programa PHP aparte del propio navegador WEB.

De la misma forma la ejecución de un programa PHP se puede realizar desde un cliente web de cualquier plataforma: el usuario puede escoger su sistema operativo y su cliente web preferidos.

El propio diseño de PHP lleva incorporada esta virtud, el programa se ejecuta en un servidor al cual se puede acceder desde cualquier puesto e una red. El servidor siempre podría limitar el acceso a solo determinados puestos y además obligar a la autenticación del usuario para poder acceder a ciertas partes de un programa.

Desde mi punto de vista hay muchas situaciones por las cuales se podría escoger PHP como lenguaje de programación para entorno web, en comparación con otros existentes en el mercado, lo más probable es que si algún día llegasen a tener alguna oportunidad de trabajar con PHP, ellos mismos darían las opciones por las cuales escoger este lenguaje.

Mientras tanto podemos listar algunos de ellas:

- Es muy fácil de aprender, siempre y cuando se tengan los conocimientos básicos de programación
- Se puede hacer cualquier cosa con PHP
- Está ampliamente probado como herramienta, más de un millón y medio de servidores por el momento, en todo el mundo lo avalan como una plataforma para desarrollar aplicaciones de portales, comercio electrónico, aplicaciones en intranets etc.
- Se puede utilizar prácticamente en cualquier plataforma
- Aunque no dispone de soporte comercial, existen numerosas listas de correo en las que se pueden obtener soluciones a los problemas que se pudieran presentar.
- Existen recursos en la web que pueden facilitar el desarrollo de nuestras aplicaciones, existen bibliotecas de clases que resuelven los problemas más frecuentes con los que se puede encontrar el programador.

- Además de todo esto no requiere de recursos desmesurados para trabajar, es posible tener PHP funcionando en un portátil 486 con 16M de RAM para desarrollo, aun que claro en un equipo un poco más grande estará mucho más cómodo.

La facilidad con que PHP es capaz de consultar una base de datos le configura un apotencia excepcional a la hora de generar páginas web dinámicas.

No obstante hay que reconocer que la mayoría de las aplicaciones desarrolladas con PHP con un gestor de base de datos utiliza MySql, un sistema gestor eficiente, suficiente para la mayoría de las aplicaciones que se desarrolla. Además las últimas versiones que han surgido parecen muy prometedoras. Quizá una de las características más importantes que incorpora es la facilidad para consultar bases de datos y generar páginas en función de los resultados obtenidos en la correspondiente consulta.

Es fácil advertir que el realizar este proyecto me ha llenado de una gran satisfacción no solo por ser mi proyecto de tesis y mi posibilidad de titulación, sino también por que en el transcurso de esta investigación y desarrollo he adquirido muchos conocimientos, creo que con nada se puede superar esta sensación de bienestar una vez que se ha logrado uno de los objetivos más soñados de cualquier universitario.

Espero que este proyecto sea de gran utilidad para las generaciones venideras, le doy las gracias a la Universidad Nacional Autónoma de México y en especial a la Facultad de Estudios Superiores Cuautitlán Campus 4.

## **BIBLIOGRAFÍA.**

### **Introducción a los sistemas de Bases de Datos**

Versión en Español de Jaime Malpica

Addison-Wesley Iberoamericana

S.I.T.E.S.A

### **Las Finanzas en la Empresa**

Información, Análisis, Recursos y Planeación

Joaquín A. Moreno Fernández

Mac Graw Hill

### **Aprendiendo HTML 3.0 para WEB en una semana**

Segunda Edición

Laura Lemay

Prentice Hall . Hispanoamericana

### **HTML 4. Guía de Referencia y Tutorial**

Jose Luis Raya, Jose A. Moreno, Antonio López

Alfa Omega RA-MA

### **Contabilidad Intermedia**

Donald E. Kieso, Jerry J. Weygandt

Limusa Noriega Editores

### **La contabilidad en la Administración de Empresas**

Robert N., Anthony D.C.S

UTEHA

**Introducción al Diseño de sistemas**

Enid Squire

Fondo Educativo Interamericano

**Análisis y Diseño de Sistemas de Información**

James A. Senn

McGraw Hill

**Java Script (Lenguaje de Programación para Computadora)**

David Flanagan

Andrew Schulman

**HTML 4.0 E. Stephen Mack, Janan Platt**

Anaya Multimedia C. 1988

**Análisis e interpretación de la Información Financiera**

C.P. Alberto García Mendoza

Cia. Editorial Continental S.A de C.V México

ISBN 968-26-0026-X

**Contabilidad financiera. Sistema de Información p/tomar decisiones**

I Eugene McNeil. Universidad de Houston

Editorial Diana ISBN 968-13-0913-8

**Teoría de la Contabilidad Financiera**

Instituto Mexicano de Contadores Públicos

Javier Cocina Martínez

ISBN 968-6964-96-7

**Principios fundamentales de Contabilidad**

William W. Pyle, Jhon Arch White

Editorial Continental S. A de C.V

ISBN 968-26-0268-8

### **Análisis e Interpretación de los Estados Financieros**

C.P. Cesar Calvo Langarica

Publicaciones Administrativas y Contables.

### **Creación de Aplicaciones WEB con PHP4**

Prentice Hall

Ratschiller y Gerken.

### **Aprediendo Mysql en 21 Días**

Marlakowski Mark, butcher Tony

Pearson Edicación, México 2001

### **PHP – 4**

Fabrega Pedro Pablo

Prentice Hall, España.

### **REFERENCIAS :**

[www.linux.com](http://www.linux.com)

[www.linuxdoc.org](http://www.linuxdoc.org)

[www.linux-mandrake.com](http://www.linux-mandrake.com)

[www.gnu.org](http://www.gnu.org)

<http://www.intranet.com>

<http://www.lavariable.com>

[www.gpl.com](http://www.gpl.com)

## ANEXO 1. CASO PRÁCTICO

El Sistema Contable Financiero para Plataforma Linux, es una aplicación que toma como base el modelo de Ciclo de Vida Clásico y es desarrollada pensando en satisfacer las demandas básicas de información contable por parte de las pequeñas y medianas empresas.

Para efectos del presente proyecto, la aplicación solo constará de los siguientes módulos:

- **Archivo** : En donde se realizará la captura del catálogo de cuentas y la captura de las pólizas.
- **Reportes** : En esta parte se podrán realizar las impresiones de los auxiliares, balance general y el estado de resultados.
- **Razones Financieras**: En esta sección de la aplicación el sistema solo contará con las razones financieras más básicas, pero permitirá que el usuario pueda definir sus propias fórmulas.
- **Configuración** : Esta es la parte medular del sistema en donde el usuario administrador dará de alta los usuarios, asignará perfiles, dará de alta datos generales de empresas, realizará traspasos de saldos, creación de periodos, etc....

La construcción del Sistema Contable para Plataforma Linux, está conformada por las siguientes fases:

### Investigación Preliminar.

Esta fase contempla la recopilación de información por parte de los posibles usuarios del sistema. Se contempla el Estudio de Factibilidad.

**Factibilidad Técnica.** Se contemplan situaciones como el equipo actual, la tecnología que existe en donde se pretende implantar la aplicación e incluso el personal con el que se cuenta. La

presente Tesis, está basada en un Sistema Operativo y lenguajes que no requieren de una gran inversión por lo que tiene una gran ventaja. Y el lenguaje utilizado nos permite diseñar sistemas de fácil manejo y muy amigables para los usuarios

**Factibilidad Económica.** ¿Los beneficios que se obtienen son suficientes como para aceptar los costos?. En definitiva, el sistema permite tener un absoluto control sobre la información. La cual estando situada dentro de las bases de datos nos permite explosionarla e implosionarla de manera sencilla y rápida, permitiendo obtener reportes de la información de forma veraz y oportuna, para la toma de decisiones.

**Factibilidad Operacional.** El diseño del sistema se realiza de forma tan amigable que el usuario se adapta fácilmente y no opondrá mucha resistencia, permitiendo de esa forma obtener el máximo beneficio e incluso se motiva al usuario a dar nuevas ideas o aportar datos que hagan más robusto el sistema.

#### (a) Análisis.

Dentro de esta fase se contempla toda la información que se recopiló con las entrevistas a diferentes usuarios del área contable, extrayendo sus opiniones sobre por qué ocurren las cosas, que soluciones proponen y que ideas tiene para cambiar el proceso.

Dentro de este análisis se contemplaron documentos y reportes generados por otros sistemas a fin de revisarlos junto con los entrevistados y poder obtener sus opiniones y mejoras al respecto.

Basándose en este análisis se determinaron los siguientes requerimientos:

#### Requerimientos básicos:

- El sistema tendrá dos tipos de usuarios (usuario operativo, usuario administrador).

- Validar el acceso al usuario.
- Funciones especiales por cada tipo de usuario.

#### **Funciones para el Usuario operativo:**

- Captura e impresión del Catalogo de cuentas.
- Captura e impresión de Polizas
- Impresión de Reportes auxiliares
- Impresión de Balance General
- Impresión de Estado de Resultados
- Impresión de razones financieras

#### **Funciones para el Usuario Administrador:**

- Captura e impresión del Catalogo de cuentas.
- Captura e impresión de Pólizas
- Impresión de Reportes auxiliares
- Impresión de Balance General
- Impresión de Estado de Resultados
- Impresión de razones financieras
- Configuración del Catálogo de Cuentas
- Configuración de y creación de Perfiles
- Alta de Empresas)
- Traspaso de Saldos
- Creación de Periodos
- Respaldo

Una vez terminado el análisis de realizaron los siguientes diagramas, que nos permiten tener una visión más detallada acerca de la estructura del sistema:

- **Diagrama Modular:** Permite visualizar los módulos por los que está compuesto el sistema. (Ver Diagrama Modular)
  
- **Diagrama de Casos de Uso:** Permite especificar de una manera más clara y sencilla de cada uno de los requerimientos que debe contener el sistema.(Ver diagrama de Casos de Uso)
  
- **Tabla de visualización de Contenidos:** Tiene el objetivo de mostrar cual es el comportamiento general del sistema que se va a proponer. (Ver Tabla de visualización de Contenidos.
  
- **Modelo Entidad – Relación:** En este modelo podremos visualizar de forma más detallada la relación que existe entre tablas y campos. (Ver diagrama Entidad-Relación)
  
- Basándose en estos modelos se define a detalle las tablas que contendrá el sistema.

Diagrama Modular



Diagrama de Casos de Uso

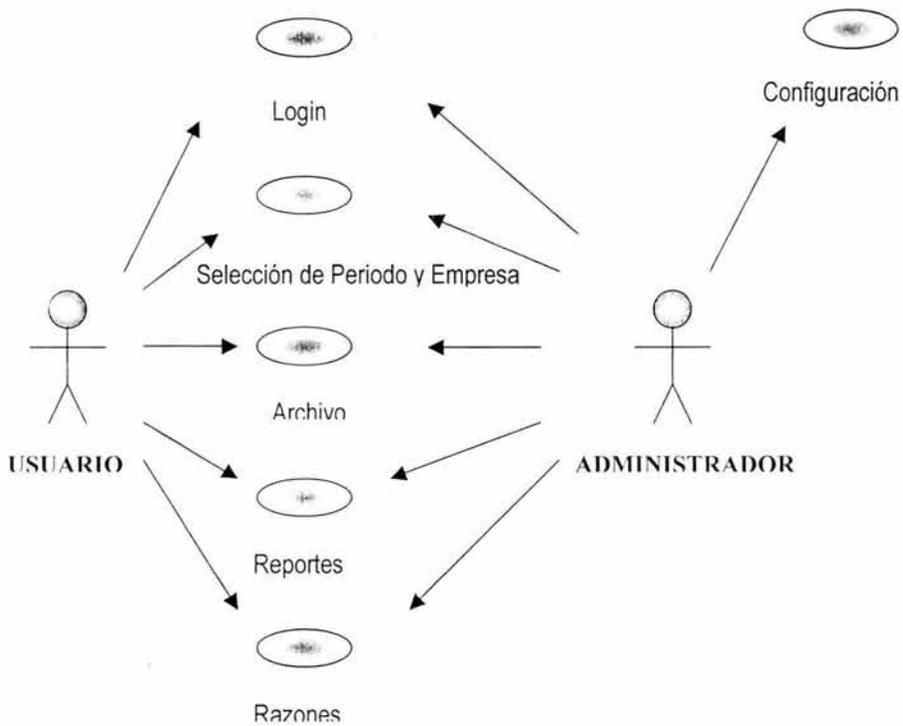
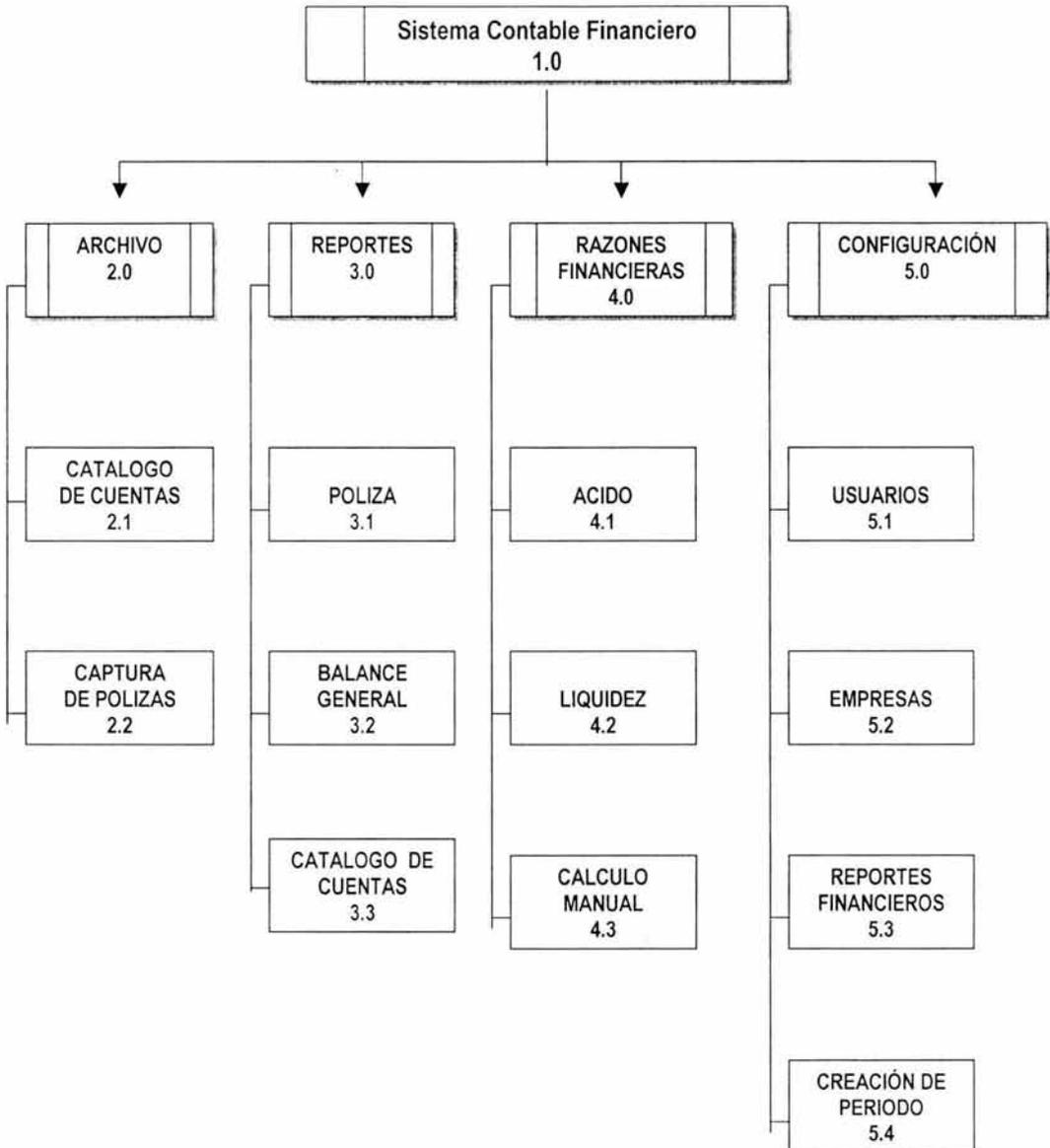


Tabla de Visualización de Contenido



## Descripción de Tabla de Visualización de contenidos

### 1.0 Sistema Contable Financiero.

2.0 **Archivo:** En este módulo se realizarán las capturas correspondientes, como son el catalogo de cuentas y la captura de pólizas.

2.1 **Catalogo de Cuentas:** Se da de alta el catálogo de cuentas, permitiendo su impresión y modificación (realizando las validaciones correspondientes).

2.2 **Captura de Pólizas:** Se realiza la captura de pólizas dentro del sistema.

3.0 **Reportes:** Basándose en la captura realizada en el módulo anterior obtendremos la información para los siguientes reportes.

3.1 **Póliza:** Se realiza la impresión de la póliza tal como se refleja en la pantalla.

3.2 **Balance General:** Se realiza la impresión del Balance general en forma de Reporte.

3.3 **Catalogo de Cuentas:** Se realiza la impresión del Catalogo forma de reporte.

4.0 **Razones Financieras:** El sistema permitirá realizar las impresiones de razones financieras que ya se encuentran establecidas dentro del sistema .

4.1 **Razón de Ácido:** Realiza la impresión de la razón del ácido basándose en un determinado periodo que el usuario digita.

4.2 **Razón de Liquidez:** Realiza la impresión de la razón del ácido basándose en un determinado periodo que el usuario digita.

5.0 **Configuración:** El usuario Administrador define la configuración de todo el sistema.

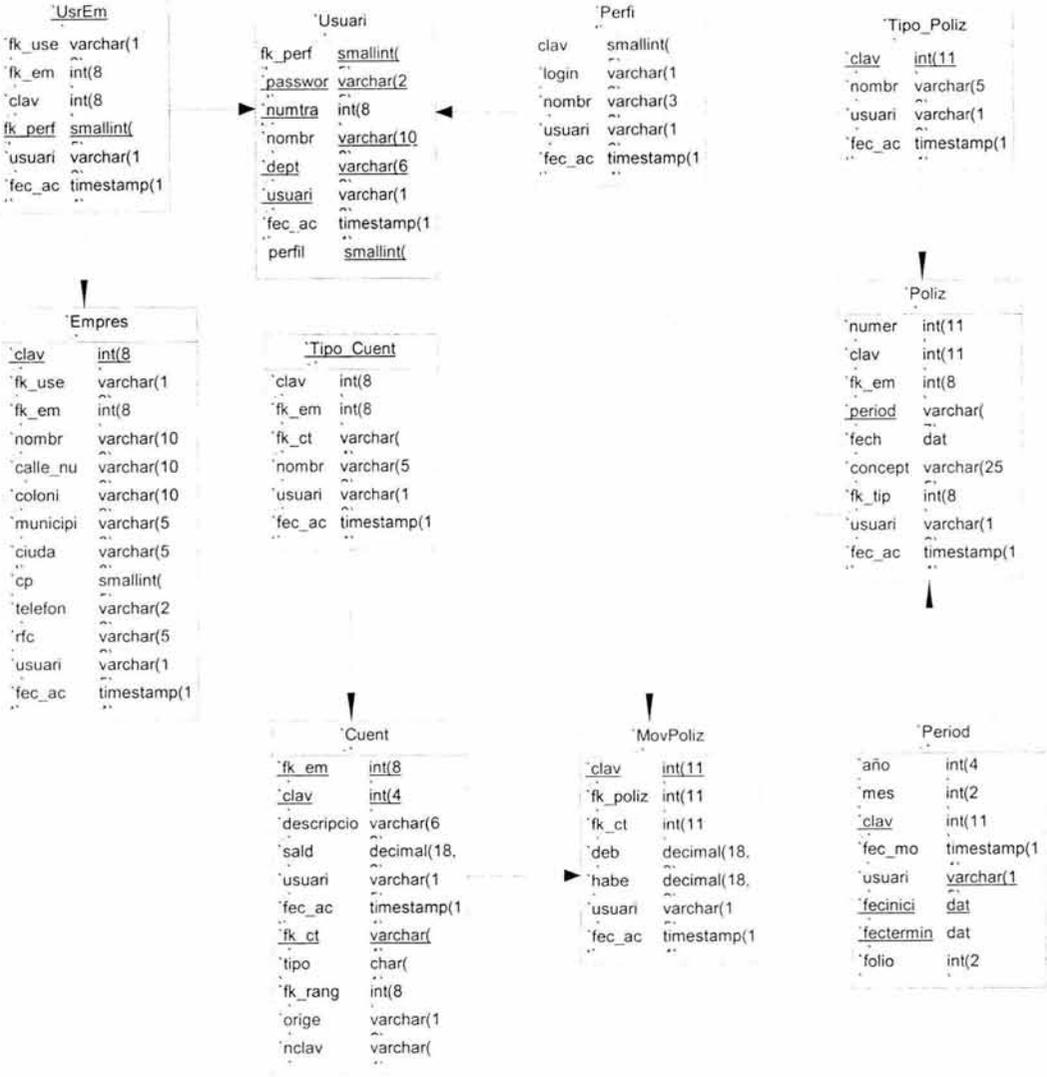
5.1 **Usuarios:** Se dan de alta los usuarios definiendo su perfil de operación.

5.2 **Empresas:** Se dan de alta las empresas en el sistema.

5.3 **Reporte financieros:** se definen los rangos de cuentas, para la impresión de los reportes financieros y razones financieras.

5.4 **Creación de siguiente Periodo:** Se corre proceso que genera nuevas bases de datos para comenzar con el nuevo periodo.

**Modelo Entidad Relación**



**Falta página**

**N° 188**

Tabla MOVPOLIZA en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>clave</u>	int(11)	No	Primary		auto_increment
<u>fk_poliza</u>	int(11)	No	None	0	
<u>fk_cta</u>	int(11)	No	None	0	
<u>movimiento</u>	int(1)	No	None	0	
<u>cantidad</u>	decimal(18,2)	Yes	None		
<u>usuario</u>	varchar(15)	Yes	None		
<u>fec_act</u>	timestamp(14)	Yes	None		

Tabla PERFIL en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>clave</u>	smallint(5)	No	Primary		auto_increment
<u>nombre</u>	varchar(30)	Yes	None		
<u>usuario</u>	varchar(10)	No	None		
<u>fec_act</u>	Timestamp(14)	Yes	None		

Tabla PERIODO en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>año</u>	int(4)	No	Primary	0	
<u>mes</u>	int(2)	No	Primary	0	
<u>fec_mov</u>	timestamp(14)	Yes	None		
<u>usuario</u>	varchar(15)	Yes	None		
<u>fecinicio</u>	Date	Yes	None		
<u>fectermino</u>	Date	Yes	None		
<u>folio</u>	int(2)	Yes	None		

Tabla POLIZA en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>numero</u>	int(11)	Yes	None		
<u>clave</u>	int(11)	No	Primary		auto_increment
<u>fk_emp</u>	int(8)	No	None	0	
<u>periodo</u>	varchar(7)	No	None		
<u>fecha</u>	Date	No	None	0000-00-00	
<u>concepto</u>	varchar(255)	No	None		
<u>fk_tipo</u>	int(8)	No	None	0	
<u>usuario</u>	varchar(10)	Yes	None		
<u>fec_act</u>	timestamp(14)	Yes	None		

Tabla TIPO\_CUENTA en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>clave</u>	int(8)	No	Primary		auto_increment
<u>nombre</u>	varchar(50)	No	None		
<u>usuario</u>	varchar(10)	No	None		
<u>fec_act</u>	timestamp(14)	Yes	None		

Tabla TIPO\_POLIZA en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>clave</u>	int(11)	No	Primary		auto_increment
<u>nombre</u>	varchar(50)	No	None		
<u>usuario</u>	varchar(10)	No	None		
<u>fec_act</u>	timestamp(14)	Yes	None		

Tabla USREMP en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>fk_user</u>	varchar(10)	No	Primary		
<u>fk_emp</u>	int(8)	No	Primary	0	
<u>usuario</u>	varchar(10)	No	None		
<u>fec_act</u>	timestamp(14)	Yes	None		

Tabla USUARIO en Base de Datos DBCONTA

Field name	Type	Allow nulls?	Key	Default value	Extras
<u>login</u>	varchar(10)	No	Primary		
<u>password</u>	varchar(25)	Yes	None		
<u>numtrab</u>	int(8)	Yes	None		
<u>nombre</u>	varchar(100)	Yes	None		
<u>depto</u>	varchar(60)	Yes	None		
<u>usuario</u>	varchar(10)	Yes	None		
<u>fec_act</u>	timestamp(14)	Yes	None		
<u>fk_perfil</u>	smallint(5)	No	None	0	

## Diseño

El diseño de este proyecto está conformado por las siguientes partes:

Tanto el perfil de administrador, como el perfil de usuario dentro del sistema tienen que teclear su clave de acceso para tener acceso al sistema, esto se realiza en la siguiente pantalla principal de acceso al sistema:



En esta pantalla el usuario tecleará su usuario y password designados por el administrador del sistema.

Al acceder por primera vez, se hará a través de la clave del administrador para que realice las capturas necesarias y poder utilizar el sistema.

Una vez que se ha tecleado el usuario y password del administrador aparece el siguiente menú:



En el cual comenzará a realizar las siguientes capturas

Creación del periodo:



En esta pantalla se dan de alta los periodos sobre los que se trabajará durante el año en la siguiente pantalla:



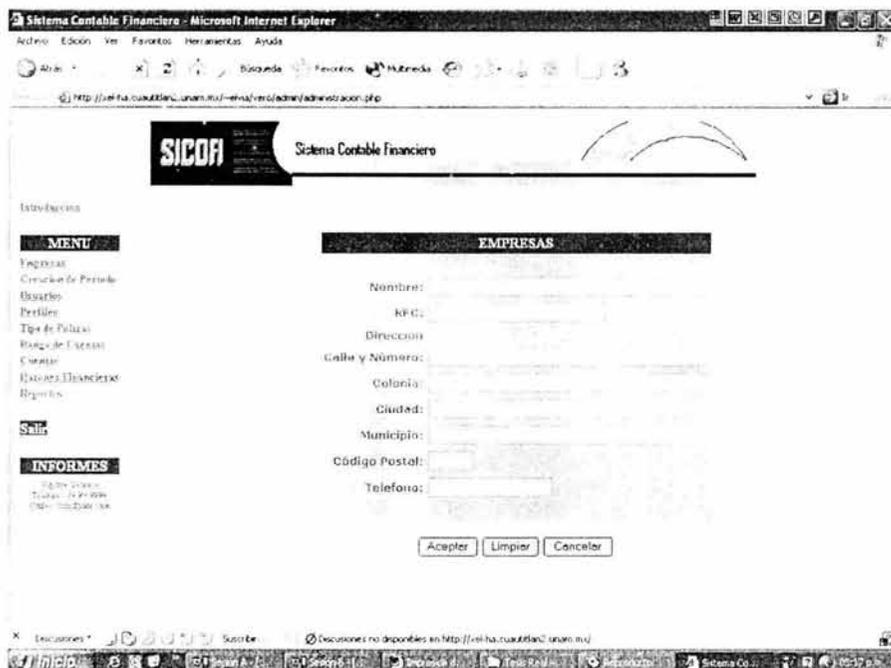
Se da de alta el año y mes en cuestión, así como el rango de fechas que abarcará el periodo definido, adicional a esto se inicializa el folio en el que iniciará la numeración de pólizas.

#### Alta de Empresas:



Al agregar una nueva empresa al catálogo se visualiza la siguiente pantalla en donde el usuario administrador tecleará los siguientes datos:

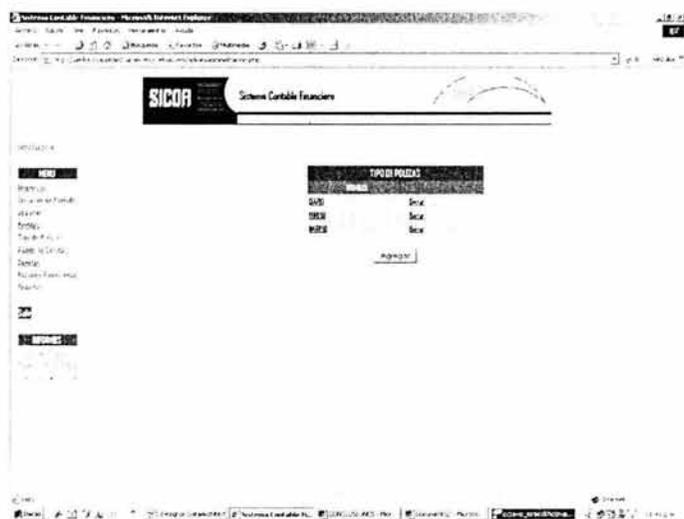
- Nombre
- R.F.C
- Dirección
- Colonia
- Ciudad
- Municipio
- Código postal
- Teléfono



## Alta de usuarios:



## Alta del catalogo de Pólizas:



## Alta del Catálogo de cuentas:

The screenshot shows the SICOR web application interface. The main content area displays a table titled 'Cuentas' (Accounts) with the following data:

Cuentas	Descripción	Saldo	Debe	Haber
101	Caja	10000.00		
102	Bancos	20000.00		
103	Clientes por Cobrar	30000.00		

Below the table, there is an 'Agregar' (Add) button. The left sidebar contains a navigation menu with options like 'Inicio', 'Cuentas', 'Cuentas de Periodo', 'Periodo', 'Cuentas de Periodo', 'Cuentas de Estado', 'Cuentas de Estado', 'Cuentas de Estado', and 'Cuentas de Estado'. The top of the page features the SICOR logo and the title 'Sistema Contable Financiero'.

Una vez que el usuario administrador ha cargado toda esta información el sistema ya puede ser operado de manera funcional por los usuarios del mismo.