

6 872748



UNIVERSIDAD DON VASCO, A. C.

**INCORPORACION No. 8727-48 A LA
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

ESCUELA DE INFORMATICA

**"DESARROLLO DE UN SISTEMA DE CONTROL MÉDICO
MEDIANTE MICROSOFT ACCESS 97 HISTOCLIN VER. 1.0 ."**

**SEMINARIO DE INVESTIGACIÓN
QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN INFORMATICA
P R E S E N T A :**

CARLOS ALBERTO ESCALERA GÓMEZ

ASESOR: ISC. FRANCISCO MANUEL LÓPEZ CALLEGOS



**UNIVERSIDAD
DON VASCO, A.C.**

URUAPAN,

MICHOACAN.

2001



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria:

A mis padres por todo el apoyo que he recibido de ellos.

**Primitivo Escalera Campoverde
María Gómez Gutiérrez**

INDICE

Introducción.....	6
Capítulo Uno	
1.1 Informática o Computación	13
1.2 Ordenador o Computadora.....	14
1.2.1 Tipos de ordenadores o computadoras.....	14
1.2.1.1 Ordenadores analógicos.....	15
1.2.1.2 Ordenadores digitales	16
1.2.1.2.1 Historia	18
1.2.1.2.2 La máquina analítica.....	18
1.2.1.2.3 Primeros ordenadores.....	19
1.2.1.3 Ordenadores electrónicos	20
1.2.1.3 .1 Circuitos integrados	21
1.2.1.3 .2 Hardware	22
1.2.1.3 .3 CPU (unidad central de proceso)	23
1.2.1.3 .4 Dispositivos de entrada	24
1.2.1.3 .5 Dispositivos de almacenamiento	25
1.2.1.3 .6 Dispositivos de salida	27
1.3 Sistemas operativos	28
1.4 Programación.....	28
1.5 Lenguajes	29
1.5.1 Lenguaje máquina	30
1.5.2 Lenguaje ensamblador.....	30

1.5.3 Lenguajes de alto nivel	31
1.5.4 Evolución futura.....	34
1.6 Lenguaje de programación	35
1.6.1 Lenguajes de bajo nivel	36
1.6.2 Lenguajes de alto nivel	36
1.6.3 Intérpretes y compiladores	37
1.7 Programa	38
1.8 Código fuente.....	39
1.9 Programa ejecutable	39
1.10 Algoritmo.....	40

Capítulo Dos

2.1 Base de datos	42
2.1.1 Funciones de las bases de datos.....	43
2.1.2 Ventajas de los sistemas de bases de datos	44
2.2 Tipos de bases de datos.....	45
2.2.1 Base de datos relacional.....	45
2.2.1.1 Normalización.....	46
2.2.1.2 Conversión de los diagramas entidad relación a tablas.....	47
2.2.2 Bases de datos jerárquico	47
2.2.3 Bases de datos de redes.....	48
2.2.4 Bases de datos orientadas a objetos.....	48
2.3 Diccionario de datos	49
2.4 Programación estructurada.....	50

2.5 Programación orientada a objetos.....	50
2.6 Interfaz de programación de aplicaciones	51
2.7 Aplicación (informática).....	52
2.8 Sistema operativo.....	53
2.8.1 Cómo funciona un sistema operativo.....	53
2.8.2 Sistemas operativos actuales.....	55
2.8.3 Tecnologías futuras.....	56
2.9 Microsoft Corporation	56
2.10 Windows.....	60
2.11 Interfaz gráfica de usuario	60
2.12 Multitarea.....	61
2.13 Ventana.....	62

Capítulo Tres

3.1 Conceptos de análisis y diseño de sistemas.....	64
3.2 Tipos de sistemas de información	65
3.2.1 Sistemas de procesamiento de transacciones	65
3.2.2 Sistemas de automatización de oficina y sistemas de manejo de conocimiento	65
3.2.3 Sistemas de información gerencial.....	66
3.2.3 Sistema de apoyo a decisiones.....	67
3.2.4 Sistemas expertos e inteligencia artificial.....	67
3.2.5 Sistemas de apoyo a decisiones de grupo.....	69
3.2.6 Sistemas de apoyo a ejecutivos.....	70

3.3 Componentes estructurales de los sistemas de información	70
3.3.1 Bloque de entrada	72
3.3.2 Bloque de modelos.....	73
3.3.3 Bloque de salida	74
3.3.5 Bloque de tecnología	76
3.3.6 Bloque de base de datos.....	78
3.3.7 Bloque de controles.....	79
3.4 Fuerzas de diseño de los sistemas de información.....	80
3.4.1 Integración	82
3.4.2 Interfaz usuario / sistema.....	84
3.4.3 Fuerzas competitivas	85
3.4.4 Calidad y utilidad de la información.....	85
3.4.5 Requerimientos de sistemas	86
3.4.6 Requerimientos de procesamientos de datos	88
3.4.7 Factores organizacionales.....	90
3.4.8 Requerimientos de costo- eficacia.....	91
3.4.9 Factores humanos	91
3.5 Paradigmas de la ingeniería de software.....	96
3.5.1 El ciclo de vida clásico.....	97
3.5.1.1 Ingeniería y análisis del sistema	97
3.5.1.2 Análisis de los requisitos del software.....	98
3.5.1.3 Diseño.....	98
3.5.1.4 Codificación	99

3.5.1.5 Prueba.....	99
3.5.1.6 Mantenimiento	99
3.5.2 Construcción de prototipos.....	101
3.5.3 El modelo en espiral	103

Capítulo Cuatro

4.1 Conceptos básicos de análisis y diseño orientado a objetos	106
4.2 Notación	108
4.3 Análisis	109
4.4 Modelado de datos.....	110
4.5 Diseño.....	112
4.5.1 Atributos y características de la Base de Datos.....	112
4.5.2 Diseño y características de las tablas.....	114
4.5.3 Diagrama de estados del programa.....	121
4.4.4 Pruebas al sistema.....	126
Conclusiones.....	128
Bibliografía.....	134

3.3 Componentes estructurales de los sistemas de información	70
3.3.1 Bloque de entrada	72
3.3.2 Bloque de modelos.....	73
3.3.3 Bloque de salida	74
3.3.5 Bloque de tecnología	76
3.3.6 Bloque de base de datos.....	78
3.3.7 Bloque de controles.....	79
3.4 Fuerzas de diseño de los sistemas de información.....	80
3.4.1 Integración	82
3.4.2 Interfaz usuario / sistema.....	84
3.4.3 Fuerzas competitivas	85
3.4.4 Calidad y utilidad de la información.....	85
3.4.5 Requerimientos de sistemas	86
3.4.6 Requerimientos de procesamientos de datos	88
3.4.7 Factores organizacionales.....	90
3.4.8 Requerimientos de costo- eficacia.....	91
3.4.9 Factores humanos	91
3.5 Paradigmas de la ingeniería de software.....	96
3.5.1 El ciclo de vida clásico.....	97
3.5.1.1 Ingeniería y análisis del sistema	97
3.5.1.2 Análisis de los requisitos del software.....	98
3.5.1.3 Diseño.....	98
3.5.1.4 Codificación	99

3.5.1.5 Prueba.....	99
3.5.1.6 Mantenimiento	99
3.5.2 Construcción de prototipos.....	101
3.5.3 El modelo en espiral.....	103

Capítulo Cuatro

4.1 Conceptos básicos de análisis y diseño orientado a objetos	106
4.2 Notación	108
4.3 Análisis.....	109
4.4 Modelado de datos.....	110
4.5 Diseño.....	112
4.5.1 Atributos y características de la Base de Datos.....	112
4.5.2 Diseño y características de las tablas.....	114
4.5.3 Diagrama de estados del programa.....	121
4.4.4 Pruebas al sistema.....	126
Conclusiones.....	128
Bibliografía.....	134

INTRODUCCIÓN

Planteamiento del problema

En el manejo de la información se vuelve cada vez más complejo está, y más aún cuando se hace de una forma manual, y que se puede contar con las siguientes desventajas:

- ✓ Retraso en la búsqueda de información.
- ✓ Lentitud en la escritura de los datos.
- ✓ Posibles errores de legibilidad, que no se pueda entender lo que se escribió.
- ✓ Por lo general, el tener un error implica que se tiene que comenzar de nuevo el documento.
- ✓ Poca presentación en los documentos, entre otras desventajas.

En el caso de los médicos es muy laborioso el llenado de expedientes de los pacientes, por lo que se vuelve una tarea poco grata y tediosa el realizar los distintos documentos que tienen que elaborar.

Un médico recibe muy poca información sobre el uso de las computadoras,

por lo que se vuelve más complejo el problema, y los que tienen experiencia sobre el uso de los equipos de cómputo, algunos han tomado cursos especiales de informática, ya que se reconoce como una necesidad.

El contar con una herramienta especializada que les facilite el manejo de la información en su computadora y además que sea sencillo su uso, así le podrán sacar provecho a los equipos de cómputo, por que por medio del sistema HISTOCLIN ver. 1.0 resolverían problemas específicos de control médico.

Con el uso de HISTOCLIN versión 1.0 se obtendrán entre otros, los siguientes beneficios:

- ✓ Contar con una base de datos con todos los datos necesarios de sus pacientes.
- ✓ Realizar reportes con una mayor calidad.
- ✓ Tener una interfaz gráfica de usuario amigable.
- ✓ Evitar la redundancia en la información
- ✓ Hacer consultas complejas por medio de SQL.
- ✓ Amigables formularios para la captura de la información.
- ✓ Rapidez en realizar sus informes.

Cabe mencionar que hasta el momento no existe un programa con interfaz gráfica que resuelva los principales problemas del tratamiento de la información para un médico. Existen paquetes pero fueron desarrollados bajo ambiente MS-DOS.

En esencia para un médico el contar con información confiable, rápida y oportuna se vuelve de suma importancia, dado que una confusión entre expedientes puede tener consecuencias graves.

En la actualidad gran parte del mundo de la información se maneja por medio de equipos electrónicos, dando como resultado una mayor rapidez, confiabilidad y así ser más productivos en el desempeño de nuestro trabajo.

Para realizar el procesamiento de los datos de forma electrónica, es necesario contar con equipo de cómputo que es también conocido como *hardware*, y por otro lado tenemos que dotar al equipo de cómputo de programas informáticos que son conocidos como *software*.

El *software* o programas informáticos es donde llevamos a cabo las tareas

necesarias para nuestro trabajo, el *software* es desarrollado para satisfacer una necesidad en el que implica un largo proceso para la consecución de un sistema de información, ya que éste se compone de varios pasos dependiendo del método o paradigma de la ingeniería de *software*.

Para desarrollar los programas informáticos existen varias herramientas que nos permiten la realización de programas, como son los lenguajes de programación, administradores de bases de datos y también las herramientas CASE entre otros.

En la tesis se desarrolla un sistema de control médico que tiene el nombre de HISTOCLIN versión 1.0, el cual se realizará en un administrador de bases de datos que es Microsoft Access 97 para Windows 95, con esta herramienta y el programa HISTOCLIN versión 1.0, el principal objetivo es el de agilizar el trabajo de papeleo cotidiano de los médicos como son: realizar historias clínicas, recetas, la impresión de estos documentos y realizar algunas consultas especiales para la elaboración de estadísticas.

En HISTOCLIN versión 1.0 también se trata de probar que se puede desarrollar aplicaciones en un administrador de bases de datos como es Microsoft Access 97 sin tener que utilizar en gran medida el uso de la programación, que en

este caso utiliza la herramienta de programación que es Microsoft Visual Basic, para la creación del sistema nos apoyamos en los asistentes y en el generador de eventos y procedimientos.

Es importante hacer notar que HISTOCLIN versión 1.0 estará desarrollada utilizando la interfaz gráfica más popular y utilizada mundialmente Microsoft Windows. HISTOCLIN versión 1.0 utiliza hace uso de la interfaz gráfica para facilitar el uso del programa por parte de los médicos.

Objetivos

- ✓ Realizar un programa de control de historias clínicas con el menor uso de la programación.
- ✓ Evitar la redundancia en la información de los expedientes.
- ✓ Contar con una base de datos de información relevante y confiable.
- ✓ Facilitar el uso del programa.
- ✓ Establecer reportes necesarios para los médicos.

Hipótesis

Facilitar el control de la información de un médico, mediante el uso de HISTOCLIN versión 1.0.

Capítulo uno

En el capítulo uno se abarca lo que son los fundamentos de la informática, desde que es un programa informático, que es informática, así como los tipos de computadoras que han existido hasta el momento. También los tipos de lenguajes de programación, su evolución y sus principales características de éstos.

Capítulo dos

En el capítulo dos se hace énfasis en los tipos de bases de datos que existen, sus características, sus ventajas. También se menciona algo de historia de la compañía más grande en el desarrollo de *software* que es Microsoft, así como de su popular Microsoft Windows y sus interfaces gráficas de usuario. Por otro lado se toca el punto de los métodos de programación más popular que existen.

Capítulo tres

En capítulo tres de la tesis se enfatiza en los métodos para el desarrollo de sistema de información, sus ventajas y también cuál utilizar dependiendo del tipo de sistema o programa que se tenga que desarrollar. En forma importante se menciona los tipos de sistemas de información, componentes de los sistemas de información y las fuerzas de diseño de los sistemas de información.

Capítulo cuatro

En el capítulo cuatro y último se encuentra el análisis y diseño del HISTOCLIN versión 1.0 así como toda la documentación de éste. El análisis y el diseño se realizó utilizando el enfoque que tiene Coad y Yourdon, por lo que se utiliza su notación. La técnica de Coad y Yourdon esta orientada para el uso del análisis, diseño y programación orientada a objetos. También se utiliza el paradigma de programación en espiral.

CAPÍTULO UNO

En este capítulo se abarca lo que son los fundamentos de la informática, desde que es un programa informático, que es informática, así como los tipos de computadoras que han existido hasta el momento. También los tipos de lenguajes de programación, su evolución y sus principales características de éstos.

1.1 Informática o Computación

Conjunto de conocimientos científicos y de técnicas que hacen posible el tratamiento automático de la información por medio de computadoras. La informática combina los aspectos teóricos y prácticos de la ingeniería, electrónica, teoría de la información, matemáticas, lógica y comportamiento humano. Los aspectos de la informática cubren desde la programación y la arquitectura informática hasta la inteligencia artificial y la robótica. (Enciclopedia Microsoft® Encarta® 98 ©, 1993-1997).

También cubre las áreas de contabilidad y finanzas además auxiliarse de la administración. Cabe recordar que la contabilidad fue una de las áreas donde más rápidamente se empezaron a utilizar las computadoras personales,

1.2 Ordenador o Computadora

Dispositivo electrónico capaz de recibir un conjunto de instrucciones y ejecutarlas realizando cálculos sobre los datos numéricos, o bien compilando y correlacionando otros tipos de información.

El mundo de la alta tecnología nunca hubiera existido de no ser por el desarrollo del ordenador o computadora. Toda la sociedad utiliza estas máquinas, en distintos tipos y tamaños, para el almacenamiento y manipulación de datos. Los equipos informáticos han abierto una nueva era en la fabricación gracias a las técnicas de automatización, y han permitido mejorar los sistemas modernos de comunicación. Son herramientas esenciales prácticamente en todos los campos de investigación y en tecnología aplicada.

1.2.1 Tipos de ordenadores o computadoras

En la actualidad se utilizan dos tipos principales de ordenadores: analógicos y digitales. Sin embargo, el término ordenador o computadora suele utilizarse para referirse exclusivamente al tipo digital. Los ordenadores analógicos aprovechan la similitud matemática entre las interrelaciones físicas de determinados problemas y emplean circuitos electrónicos o hidráulicos para simular el problema físico. Los ordenadores digitales resuelven los problemas realizando cálculos y tratando cada

número dígito por dígito.

Las instalaciones que contienen elementos de ordenadores digitales y analógicos se denominan ordenadores híbridos. Por lo general se utilizan para problemas en los que hay que calcular grandes cantidades de ecuaciones complejas, conocidas como integrales de tiempo. En un ordenador digital también pueden introducirse datos en forma analógica mediante un convertidor analógico digital, y viceversa (convertidor digital a analógico).

1.2.1.1 Ordenadores Computadoras analógicas

El ordenador analógico es un dispositivo electrónico o hidráulico diseñado para manipular la entrada de datos en términos de, por ejemplo, niveles de tensión o presiones hidráulicas, en lugar de hacerlo como datos numéricos. El dispositivo de cálculo analógico más sencillo es la regla de cálculo, que utiliza longitudes de escalas especialmente calibradas para facilitar la multiplicación, la división y otras funciones. En el típico ordenador analógico electrónico, las entradas se convierten en tensiones que pueden sumarse o multiplicarse empleando elementos de circuito de diseño especial. Las respuestas se generan continuamente para su visualización o para su conversión en otra forma deseada.

1.2.1.2 Ordenadores o Computadoras digitales

Todo lo que hace un ordenador digital se basa en una operación: la capacidad de determinar si un conmutador, o 'puerta', está abierto o cerrado. Es decir, el ordenador puede reconocer sólo dos estados en cualquiera de sus circuitos microscópicos: abierto o cerrado, alta o baja tensión o, en el caso de números, 0 o 1. Sin embargo, es la velocidad con la cual el ordenador realiza este acto tan sencillo lo que lo convierte en una maravilla de la tecnología moderna. Las velocidades del ordenador se miden en megahercios, o millones de ciclos por segundo. Un ordenador con una velocidad de reloj de 100 MHz, velocidad bastante representativa de un microordenador o microcomputadora, es capaz de ejecutar 100 millones de operaciones discretas por segundo. Las microcomputadoras de las compañías pueden ejecutar entre 150 y 200 millones de operaciones por segundo, mientras que las supercomputadoras utilizadas en aplicaciones de investigación y de defensa alcanzan velocidades de miles de millones de ciclos por segundo.

La velocidad y la potencia de cálculo de los ordenadores digitales se incrementan aún más por la cantidad de datos manipulados durante cada ciclo. Si un ordenador verifica sólo un conmutador cada vez, dicho conmutador puede representar solamente dos comandos o números. Así, ON simbolizaría una operación o un número, mientras que OFF simbolizará otra u otro. Sin embargo, al verificar grupos de conmutadores enlazados como una sola unidad, el ordenador aumenta el número de operaciones que puede reconocer en cada ciclo. Por

ejemplo, un ordenador que verifica dos conmutadores cada vez, puede representar cuatro números (del 0 al 3), o bien ejecutar en cada ciclo una de las cuatro operaciones, una para cada uno de los siguientes modelos de conmutador: OFF-OFF (0), OFF-ON (1), ON-OFF (2) u ON-ON (3). En general, los ordenadores de la década de 1970 eran capaces de verificar 8 conmutadores simultáneamente; es decir, podían verificar ocho dígitos binarios, de ahí el término bit de datos en cada ciclo. Un grupo de ocho bits se denomina byte y cada uno contiene 256 configuraciones posibles de ON y OFF (o 1 y 0). Cada configuración equivale a una instrucción, a una parte de una instrucción o a un determinado tipo de dato; estos últimos pueden ser un número, un carácter o un símbolo gráfico. Por ejemplo, la configuración 11010010 puede representar datos binarios, en este caso el número decimal 210 (véase Sistemas numéricos), o bien estar indicando al ordenador que compare los datos almacenados en estos conmutadores con los datos almacenados en determinada ubicación del chip de memoria. El desarrollo de procesadores capaces de manejar simultáneamente 16, 32 y 64 bits de datos ha permitido incrementar la velocidad de los ordenadores. La colección completa de configuraciones reconocibles, es decir, la lista total de operaciones que una computadora es capaz de procesar, se denomina conjunto, o repertorio, de instrucciones. Ambos factores, el número de bits simultáneos y el tamaño de los conjuntos de instrucciones, continúa incrementándose a medida que avanza el desarrollo de los ordenadores digitales modernos.

1.2.1.2.1 Historia

La primera máquina de calcular mecánica, un precursor del ordenador digital, fue inventada en 1642 por el matemático francés Blaise Pascal. Aquel dispositivo utilizaba una serie de ruedas de diez dientes en las que cada uno de los dientes representaba un dígito del 0 al 9. Las ruedas estaban conectadas de tal manera que podían sumarse números haciéndolas avanzar el número de dientes correcto. En 1670 el filósofo y matemático alemán Gottfried Wilhelm Leibniz perfeccionó esta máquina e inventó una que también podía multiplicar.

El inventor francés Joseph Marie Jacquard, al diseñar un telar automático, utilizó delgadas placas de madera perforadas para controlar el tejido utilizado en los diseños complejos. Durante la década de 1880 el estadístico estadounidense Herman Hollerith concibió la idea de utilizar tarjetas perforadas, similares a las placas de Jacquard, para procesar datos. Hollerith consiguió compilar la información estadística destinada al censo de población de 1890 de Estados Unidos mediante la utilización de un sistema que hacía pasar tarjetas perforadas sobre contactos eléctricos.

1.2.1.2.2 La máquina analítica

También en el siglo XIX el matemático e inventor británico Charles Babbage

elaboró los principios de la computadora digital moderna. Inventó una serie de máquinas, como la máquina diferencial, diseñadas para solucionar problemas matemáticos complejos. Muchos historiadores consideran a Babbage y a su socia, la matemática británica Augusta Ada Byron (1815-1852), hija del poeta inglés Lord Byron, como a los verdaderos inventores de la computadora digital moderna. La tecnología de aquella época no era capaz de trasladar a la práctica sus acertados conceptos; pero una de sus invenciones, la máquina analítica, ya tenía muchas de las características de un ordenador moderno. Incluía una corriente, o flujo de entrada en forma de paquete de tarjetas perforadas, una memoria para guardar los datos, un procesador para las operaciones matemáticas y una impresora para hacer permanente el registro.

1.2.1.2.3 Primeros ordenadores

Los ordenadores analógicos comenzaron a construirse a principios del siglo XX. Los primeros modelos realizaban los cálculos mediante ejes y engranajes giratorios. Con estas máquinas se evaluaban las aproximaciones numéricas de ecuaciones demasiado difíciles como para poder ser resueltas mediante otros métodos. Durante las dos guerras mundiales se utilizaron sistemas informáticos analógicos, primero mecánicos y más tarde eléctricos, para predecir la trayectoria de los torpedos en los submarinos y para el manejo a distancia de las bombas en la

aviación.

1.2.1.3 Ordenadores electrónicos

Durante la II Guerra Mundial (1939-1945), un equipo de científicos y matemáticos que trabajaban en Bletchley Park, al norte de Londres, crearon lo que se consideró el primer ordenador digital totalmente electrónico: el *Colossus*. Hacia diciembre de 1943 el *Colossus*, que incorporaba 1.500 válvulas o tubos de vacío, era ya operativo. Fue utilizado por el equipo dirigido por Alan Turing para descodificar los mensajes de radio cifrados de los alemanes. En 1939 y con independencia de este proyecto, John Atanasoff y Clifford Berry ya habían construido un prototipo de máquina electrónica en el Iowa State College (EEUU). Este prototipo y las investigaciones posteriores se realizaron en el anonimato, y más tarde quedaron eclipsadas por el desarrollo del Calculador e integrador numérico digital electrónico (ENIAC) en 1945. El ENIAC, que según mostró la evidencia se basaba en gran medida en el 'ordenador' Atanasoff-Berry (ABC, acrónimo de Electronic Numerical Integrator and Computer), obtuvo una patente que caducó en 1973, varias décadas más tarde.

El ENIAC contenía 18.000 válvulas de vacío y tenía una velocidad de varios cientos de multiplicaciones por minuto, pero su programa estaba conectado al procesador y debía ser modificado manualmente. Se construyó un sucesor del

ENIAC con un almacenamiento de programa que estaba basado en los conceptos del matemático húngaro-estadounidense John von Neumann. Las instrucciones se almacenaban dentro de una llamada memoria, lo que liberaba al ordenador de las limitaciones de velocidad del lector de cinta de papel durante la ejecución y permitía resolver problemas sin necesidad de volver a conectarse al ordenador.

A finales de la década de 1950 el uso del transistor en los ordenadores marcó el advenimiento de elementos lógicos más pequeños, rápidos y versátiles de lo que permitían las máquinas con válvulas. Como los transistores utilizan mucha menos energía y tienen una vida útil más prolongada, a su desarrollo se debió el nacimiento de máquinas más perfeccionadas, que fueron llamadas ordenadores o computadoras de segunda generación. Los componentes se hicieron más pequeños, así como los espacios entre ellos, por lo que la fabricación del sistema resultaba más barata.

1.2.1.3 .1 Circuitos integrados

A finales de la década de 1960 apareció el circuito integrado (CI), que posibilitó la fabricación de varios transistores en un único sustrato de silicio en el que los cables de interconexión iban soldados. El circuito integrado permitió una posterior reducción del precio, el tamaño y los porcentajes de error. El microprocesador se convirtió en una realidad a mediados de la década de 1970, con la introducción del circuito de integración a gran escala (LSI, acrónimo de Large Scale Integrated) y,

más tarde, con el circuito de integración a mayor escala (VLSI, acrónimo de Very Large Scale Integrated), con varios miles de transistores interconectados soldados sobre un único sustrato de silicio.

1.2.1.3 .2 Hardware

Todos los ordenadores digitales modernos son similares conceptualmente con independencia de su tamaño. Sin embargo, pueden dividirse en varias categorías según su precio y rendimiento: el ordenador o computadora personal es una máquina de coste relativamente bajo y por lo general de tamaño adecuado para un escritorio (algunos de ellos, denominados portátiles, o *laptops*, son lo bastante pequeños como para caber en un maletín); la estación de trabajo, un microordenador con gráficos mejorados y capacidades de comunicaciones que lo hacen especialmente útil para el trabajo de oficina; el miniordenador o minicomputadora, un ordenador de mayor tamaño que por lo general es demasiado caro para el uso personal y que es apto para compañías, universidades o laboratorios; y el mainframe, una gran máquina de alto precio capaz de servir a las necesidades de grandes empresas, departamentos gubernamentales, instituciones de investigación científica y similares (las máquinas más grandes y más rápidas dentro de esta categoría se denominan supercomputadoras).

En realidad, un ordenador digital no es una única máquina, en el sentido en el que la mayoría de la gente considera a los ordenadores. Es un sistema

compuesto de cinco elementos diferenciados: una CPU (unidad central de proceso); dispositivos de entrada; dispositivos de almacenamiento de memoria; dispositivos de salida y una red de comunicaciones, denominada bus, que enlaza todos los elementos del sistema y conecta a éste con el mundo exterior.

1.2.1.3 .3 CPU (unidad central de proceso)

La CPU puede ser un único chip o una serie de chips que realizan cálculos aritméticos y lógicos y que temporizan y controlan las operaciones de los demás elementos del sistema. Las técnicas de miniaturización y de integración han posibilitado el desarrollo de un chip de CPU denominado microprocesador, que incorpora un sistema de circuitos y memoria adicionales. El resultado son unos ordenadores más pequeños y la reducción del sistema de circuitos de soporte. Los microprocesadores se utilizan en la mayoría de los ordenadores personales de la actualidad.

La mayoría de los chips de CPU y de los microprocesadores están compuestos de cuatro secciones funcionales: una unidad aritmética/lógica; unos registros; una sección de control y un bus interno. La unidad aritmética/lógica proporciona al chip su capacidad de cálculo y permite la realización de operaciones aritméticas y lógicas. Los registros son áreas de almacenamiento temporal que contienen datos, realizan un seguimiento de las instrucciones y conservan la ubicación y los resultados de dichas operaciones. La sección de control tiene tres

tareas principales: temporiza y regula las operaciones de la totalidad del sistema informático; su decodificador de instrucciones lee las configuraciones de datos en un registro designado y las convierte en una actividad, como podría ser sumar o comparar, y su unidad interruptora indica en qué orden utilizará la CPU las operaciones individuales y regula la cantidad de tiempo de CPU que podrá consumir cada operación.

El último segmento de un chip de CPU o microprocesador es su bus interno, una red de líneas de comunicación que conecta los elementos internos del procesador y que también lleva hacia los conectores externos que enlazan al procesador con los demás elementos del sistema informático. Los tres tipos de bus de la CPU son: el bus de control que consiste en una línea que detecta las señales de entrada y de otra línea que genera señales de control desde el interior de la CPU; el bus de dirección, una línea unidireccional que sale desde el procesador y que gestiona la ubicación de los datos en las direcciones de la memoria; y el bus de datos, una línea de transmisión bidireccional que lee los datos de la memoria y escribe nuevos datos en ésta.

1.2.1.3 .4 Dispositivos de entrada

Estos dispositivos permiten al usuario del ordenador introducir datos, comandos y programas en la CPU. El dispositivo de entrada más común es un teclado similar al de las máquinas de escribir. La información introducida con el

mismo, es transformada por el ordenador en modelos reconocibles. Otros dispositivos de entrada son los lápices ópticos, que transmiten información gráfica desde tabletas electrónicas hasta el ordenador; *joysticks* y el ratón o *mouse*, que convierte el movimiento físico en movimiento dentro de una pantalla de ordenador; los escáneres luminosos, que leen palabras o símbolos de una página impresa y los traducen a configuraciones electrónicas que el ordenador puede manipular y almacenar; y los módulos de reconocimiento de voz, que convierten la palabra hablada en señales digitales comprensibles para el ordenador. También es posible utilizar los dispositivos de almacenamiento para introducir datos en la unidad de proceso.

1.2.1.3 .5 Dispositivos de almacenamiento

Los sistemas informáticos pueden almacenar los datos tanto interna (en la memoria) como externamente (en los dispositivos de almacenamiento). Internamente, las instrucciones o datos pueden almacenarse por un tiempo en los chips de silicio de la RAM (memoria de acceso aleatorio) montados directamente en la placa de circuitos principal de la computadora, o bien en chips montados en tarjetas periféricas conectadas a la placa de circuitos principal del ordenador. Estos chips de RAM constan de conmutadores sensibles a los cambios de la corriente eléctrica. Los chips de RAM estática conservan sus bits de datos mientras la corriente siga fluyendo a través del circuito, mientras que los chips de RAM

dinámica (DRAM, acrónimo de Dynamic Random Access Memory) necesitan la aplicación de tensiones altas o bajas a intervalos regulares aproximadamente cada dos milisegundos para no perder su información.

Otro tipo de memoria interna son los chips de silicio en los que ya están instalados todos los conmutadores. Las configuraciones en este tipo de chips de ROM (memoria de sólo lectura) forman los comandos, los datos o los programas que el ordenador necesita para funcionar correctamente. Los chips de RAM son como pedazos de papel en los que se puede escribir, borrar y volver a utilizar; los chips de ROM son como un libro, con las palabras ya escritas en cada página. Tanto los primeros como los segundos están enlazados a la CPU a través de circuitos

Los dispositivos de almacenamiento externos, que pueden residir físicamente dentro de la unidad de proceso principal del ordenador, están fuera de la placa de circuitos principal. Estos dispositivos almacenan los datos en forma de cargas sobre un medio magnéticamente sensible, por ejemplo una cinta de sonido o, lo que es más común, sobre un disco revestido de una fina capa de partículas metálicas. Los dispositivos de almacenamiento externo más frecuentes son los disquetes y los discos duros, aunque la mayoría de los grandes sistemas informáticos utiliza bancos de unidades de almacenamiento en cinta magnética. Los discos flexibles pueden contener, según sea el sistema, desde varios centenares de miles de bytes hasta bastante más de un millón de bytes de datos. Los discos duros no pueden extraerse de los receptáculos de la unidad de disco, que contienen los dispositivos electrónicos para leer y escribir datos sobre la

superficie magnética de los discos y pueden almacenar desde varios millones de bytes hasta algunos centenares de millones. La tecnología de CD-ROM, que emplea las mismas técnicas láser utilizadas para crear los discos compactos (CD) de audio, permiten capacidades de almacenamiento del orden de varios cientos de megabytes (millones de bytes) de datos.

1.2.1.3 .6 Dispositivos de salida

Estos dispositivos permiten al usuario ver los resultados de los cálculos o de las manipulaciones de datos de la computadora. El dispositivo de salida más común es la unidad de visualización (VDU, acrónimo de Video Display Unit), que consiste en un monitor que presenta los caracteres y gráficos en una pantalla similar a la del televisor. Por lo general, las VDU tienen un tubo de rayos catódicos como el de cualquier televisor, aunque los ordenadores pequeños y portátiles utilizan hoy pantallas de cristal líquido (LCD, acrónimo de Liquid Crystal Displays) o electroluminiscentes. Otros dispositivos de salida más comunes son las impresoras y los módem. Un módem enlaza dos ordenadores transformando las señales digitales en analógicas para que los datos puedan transmitirse a través de las telecomunicaciones.

1.3 Sistemas operativos

Los sistemas operativos internos fueron desarrollados sobre todo para coordinar y trasladar estos flujos de datos que procedían de fuentes distintas, como las unidades de disco o los coprocesadores (chips de procesamiento que ejecutan operaciones simultáneamente con la unidad central, aunque son diferentes). Un sistema operativo es un programa de control principal, almacenado de forma permanente en la memoria, que interpreta los comandos del usuario que solicita diversos tipos de servicios, como visualización, impresión o copia de un archivo de datos; presenta una lista de todos los archivos existentes en un directorio o ejecuta un determinado programa.

1.4 Programación

Un programa es una secuencia de instrucciones que indican al *hardware* de un ordenador qué operaciones debe realizar con los datos. Los programas pueden estar incorporados al propio *hardware*, o bien pueden existir de manera independiente en forma de *software*. En algunas computadoras especializadas las instrucciones operativas están incorporadas en el sistema de circuitos; entre los ejemplos más comunes pueden citarse los microordenadores de las calculadoras, relojes de pulsera, motores de coches y hornos microondas. Por otro lado, un

ordenador universal, o de uso general, contiene algunos programas incorporados (en la ROM) o instrucciones (en el chip del procesador), pero depende de programas externos para ejecutar tareas útiles. Una vez programado, podrá hacer tanto o tan poco como le permita el *software* que lo controla en determinado momento. El *software* de uso más generalizado incluye una amplia variedad de programas de aplicaciones, es decir, instrucciones al ordenador acerca de cómo realizar diversas tareas.

1.5 Lenguajes

Las instrucciones deben darse en un lenguaje de programación, es decir, en una determinada configuración de información digital binaria. En las primeras computadoras, la programación era una tarea difícil y laboriosa ya que los conmutadores ON-OFF de las válvulas de vacío debían configurarse a mano. Programar tareas tan sencillas como ordenar una lista de nombres requería varios días de trabajo de equipos de programadores. Desde entonces se han inventado varios lenguajes informáticos, algunos orientados hacia funciones específicas y otros centrados en la facilidad de uso.

1.5.1 Lenguaje máquina

El lenguaje propio del ordenador, basado en el sistema binario, o código máquina, resulta difícil de utilizar para las personas. El programador debe introducir todos y cada uno de los comandos y datos en forma binaria, y una operación sencilla como comparar el contenido de un registro con los datos situados en una ubicación del chip de memoria puede tener el siguiente formato: 11001010 00010111 11110101 00101011. La programación en lenguaje máquina es una tarea tan tediosa y consume tanto tiempo que muy raras veces lo que se ahorra en la ejecución del programa justifica los días o semanas que se han necesitado para escribir el mismo.

1.5.2 Lenguaje ensamblador

Uno de los métodos inventados por los programadores para reducir y simplificar el proceso es la denominada programación con lenguaje ensamblador. Al asignar un código mnemotécnico (por lo general de tres letras) a cada comando en lenguaje máquina, es posible escribir y depurar o eliminar los errores lógicos y de datos en los programas escritos en lenguaje ensamblador, empleando para ello sólo una fracción del tiempo necesario para programar en lenguaje máquina. En el lenguaje ensamblador, cada comando mnemotécnico y sus operadores simbólicos

equivalen a una instrucción de máquina. Un programa ensamblador traduce el código fuente, una lista de códigos de operación mnemotécnicos y de operadores simbólicos, a código objeto (es decir, a lenguaje máquina) y, a continuación, ejecuta el programa.

Sin embargo, el lenguaje ensamblador puede utilizarse con un solo tipo de chip de CPU o microprocesador. Los programadores, que dedicaron tanto tiempo y esfuerzo al aprendizaje de la programación de un ordenador, se veían obligados a aprender un nuevo estilo de programación cada vez que trabajaban con otra máquina. Lo que se necesitaba era un método abreviado en el que un enunciado simbólico pudiera representar una secuencia de numerosas instrucciones en lenguaje máquina, y un método que permitiera que el mismo programa pudiera ejecutarse en varios tipos de máquinas. Estas necesidades llevaron al desarrollo de lenguajes de alto nivel.

1.5.3 Lenguajes de alto nivel

Los lenguajes de alto nivel suelen utilizar términos ingleses del tipo LIST, PRINT u OPEN como comandos que representan una secuencia de decenas o de centenas de instrucciones en lenguaje máquina. Los comandos se introducen desde el teclado, desde un programa residente en la memoria o desde un dispositivo de almacenamiento, y son interceptados por un programa que los

traduce a instrucciones en lenguaje máquina.

Los programas traductores son de dos tipos: intérpretes y compiladores. Con un intérprete, los programas que repiten un ciclo para volver a ejecutar parte de sus instrucciones, reinterpretan la misma instrucción cada vez que aparece. Por consiguiente, los programas interpretados se ejecutan con mucha mayor lentitud que los programas en lenguaje máquina. Por el contrario, los compiladores traducen un programa íntegro a lenguaje máquina antes de su ejecución, por lo cual se ejecutan con tanta rapidez como si hubiesen sido escritos directamente en lenguaje máquina.

Se considera que fue la estadounidense Grace Hopper quien implementó el primer lenguaje de ordenador orientado al uso comercial. Después de programar un ordenador experimental en la Universidad de Harvard, trabajó en los modelos UNIVAC I y UNIVAC II, desarrollando un lenguaje de alto nivel para uso comercial llamado FLOW-MATIC. Para facilitar el uso del ordenador en las aplicaciones científicas, IBM desarrolló un lenguaje que simplificaría el trabajo que implicaba el tratamiento de fórmulas matemáticas complejas. Iniciado en 1954 y terminado en 1957, el FORTRAN (acrónimo de Formula Translator) fue el primer lenguaje exhaustivo de alto nivel de uso generalizado.

En 1957 una asociación estadounidense, la *Association for Computing Machinery* comenzó a desarrollar un lenguaje universal que corrigiera algunos de los defectos del FORTRAN. Un año más tarde fue lanzado el ALGOL (acrónimo de Algorithmic Language), otro lenguaje de orientación científica. De gran difusión en

Europa durante las décadas de 1960 y 1970, desde entonces ha sido sustituido por nuevos lenguajes, mientras que el FORTRAN continúa siendo utilizado debido a las gigantescas inversiones que se hicieron en los programas existentes. El COBOL (acrónimo de Common Business Oriented Language) es un lenguaje de programación para uso comercial y empresarial especializado en la organización de datos y manipulación de archivos, y hoy día está muy difundido en el mundo empresarial.

El lenguaje BASIC (acrónimo de Código de Instrucciones Simbólicas de Uso General para Principiantes) fue desarrollado en el Dartmouth College a principios de la década de 1960 y está dirigido a los usuarios de ordenador no profesionales. Este lenguaje se universalizó gracias a la popularización de los microordenadores en las décadas de 1970 y 1980. Calificado de lento, ineficaz y poco estético por sus detractores, BASIC es sencillo de aprender y fácil de utilizar. Como muchos de los primeros microordenadores se vendieron con BASIC incorporado en el *hardware* (en la memoria ROM), se generalizó el uso de este lenguaje.

Aunque existen centenares de lenguajes informáticos y de variantes, hay algunos dignos de mención, como el PASCAL, diseñado en un principio como herramienta de enseñanza, hoy es uno de los lenguajes de microordenador más populares; el Logo fue desarrollado para que los niños pudieran acceder al mundo de la informática; el C, un lenguaje de Bell Laboratories diseñado en la década de 1970, se utiliza ampliamente en el desarrollo de programas de sistemas, al igual que su sucesor, el C++. El LISP y el PROLOG han alcanzado amplia difusión en el

campo de la inteligencia artificial.

1.5.4 Evolución futura

Una tendencia constante en el desarrollo de los ordenadores es la microminiaturización, iniciativa que tiende a comprimir más elementos de circuitos en un espacio de chip cada vez más pequeño. Además, los investigadores intentan agilizar el funcionamiento de los circuitos mediante el uso de la superconductividad, un fenómeno de disminución de la resistencia eléctrica que se observa cuando se enfrían los objetos a temperaturas muy bajas.

Las redes informáticas se han vuelto cada vez más importantes en el desarrollo de la tecnología de computadoras. Las redes son grupos de computadoras interconectados mediante sistemas de comunicación. La red pública Internet es un ejemplo de red informática planetaria. Las redes permiten que las computadoras conectadas intercambien rápidamente información y, en algunos casos, compartan una carga de trabajo, con lo que muchas computadoras pueden cooperar en la realización de una tarea. Se están desarrollando nuevas tecnologías de equipo físico y soporte lógico que acelerarán los dos procesos mencionados.

Otra tendencia en el desarrollo de computadoras es el esfuerzo para crear computadoras de quinta generación, capaces de resolver problemas complejos en formas que pudieran llegar a considerarse creativas. Una vía que se está

explorando activamente es el ordenador de proceso paralelo, que emplea muchos chips para realizar varias tareas diferentes al mismo tiempo. El proceso paralelo podría llegar a reproducir hasta cierto punto las complejas funciones de realimentación, aproximación y evaluación que caracterizan al pensamiento humano. Otra forma de proceso paralelo que se está investigando es el uso de computadoras moleculares. En estas computadoras, los símbolos lógicos se expresan por unidades químicas de ADN en vez de por el flujo de electrones habitual en las computadoras corrientes. Las computadoras moleculares podrían llegar a resolver problemas complicados mucho más rápidamente que las actuales supercomputadoras y consumir mucha menos energía. (Enciclopedia Microsoft® Encarta® 98 ©, 1993-1997)

1.6 Lenguaje de programación

Es cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o 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.

1.6.1 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. 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.

1.6.2 Lenguajes de alto nivel

Por lo general se piensa que los ordenadores 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 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.

1.6.3 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. 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 el ordenador 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.

Enciclopedia Microsoft® Encarta® 98 © 1993-1997

1.7 Programa

Es sinónimo de *software*, el conjunto de instrucciones que ejecuta un ordenador o computadora. El término puede referirse al código fuente original o a la versión ejecutable (en lenguaje máquina) de un componente de *software*. Cuando se habla de un programa se supone un cierto grado de terminación, o sea, se da por hecho que están presentes todas las instrucciones y archivos necesarios para la interpretación o compilación del programa. Por otro lado, se entiende que un programa ejecutable puede cargarse en un entorno determinado y ejecutarse independientemente de otros programas. (Enciclopedia Microsoft® Encarta® 98 ©

1993-1997).

1.8 Código fuente

Son instrucciones de programa legibles por el programador y escritas en un lenguaje ensamblador o de más alto nivel. El concepto contrario es el código objeto, que se deriva del código fuente y está diseñado para ser legible sólo por la máquina. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

1.9 Programa ejecutable

Programa que ha sido traducido a código máquina en un formato que puede cargarse en la memoria y ejecutarse. Sin embargo, en los lenguajes interpretados puede referirse simplemente al código fuente en su formato apropiado. La mayoría de los programas ejecutables en MS-DOS tiene la extensión de nombre de archivo .EXE. Para ejecutar el programa, el usuario sólo tiene que escribir el nombre del archivo (sin la extensión EXE) junto al símbolo del sistema y, a continuación, presionar la tecla Intro. El usuario no tiene que modificar el programa en modo alguno para poder ejecutarlo. Las aplicaciones como los programas de procesamiento de textos son programas ejecutables. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

1.10. Algoritmo

De una forma sencilla un algoritmo es realizar por medio de lenguaje común la solución de un programa, pero sin la necesidad de codificarlo en la computadora.

En matemáticas, método de resolución de problemas complicados mediante el uso repetido de otro método de cálculo más sencillo. Un ejemplo básico es el cálculo de la división larga en aritmética. En la actualidad, el término *algoritmo* se aplica a muchos de los métodos de resolver problemas que empleen una secuencia mecánica de pasos, como en el diseño de un programa de ordenador o computadora. Esta secuencia se puede representar en la forma de un diagrama de flujo para que sea más fácil de entender.

Al igual que los algoritmos usados en aritmética, los algoritmos para ordenadores pueden ser desde muy sencillos hasta bastante complejos. En todos los casos, sin embargo, la tarea que el algoritmo ha de realizar debe ser definible. Esta definición puede incluir términos matemáticos o lógicos o una compilación de datos o instrucciones escritas. Usando el lenguaje de la informática, esto quiere decir que un algoritmo debe ser programable, incluso si al final se comprueba que el problema no tiene solución.

En las computadoras con lógica de microordenadores incorporada, esta lógica es un tipo de algoritmo. A medida que los equipos informáticos se hacen más complejos, más y más algoritmos del *software* toman la forma del llamado *hard-*

software. Esto es, cada vez más, se están convirtiendo en parte de los circuitos básicos de las computadoras o en módulos auxiliares; también están apareciendo por sí solos en máquinas específicas como las calculadoras de nóminas. En la actualidad, existen muchos algoritmos para diversas aplicaciones y algunos sistemas avanzados como los algoritmos de inteligencia artificial llegarán a ser corrientes en el futuro. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

CAPÍTULO DOS

En este capítulo se hace énfasis en los tipos de bases de datos que existen, sus características, sus ventajas. También se menciona algo de historia de la compañía más grande en el desarrollo de *software* que es Microsoft, así como de su popular Microsoft Windows y sus interfaces gráficas de usuario. Por otro lado se toca el punto de los métodos de programación más popular que existen.

2.1 Base de datos

Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. Los datos suelen aparecer en forma de texto, números o gráficos. Desde su aparición en la década de 1950, se han hecho imprescindibles para las sociedades industriales

Hay cuatro modelos principales de bases de datos: el modelo jerárquico, el modelo en red, el modelo relacional (el más extendido hoy en día; los datos se almacenan en tablas a los que se accede mediante consultas escritas en SQL) y el modelo de bases de datos deductivas. Otra línea de investigación en este campo son las bases de datos orientadas a objeto, o de objetos persistentes. (Enciclopedia

Microsoft® Encarta® 98 © 1993-1997).

En el desarrollo de HISTOCLIN versión 1.0, el modelo de base de datos que se esta utilizando es el relacional, cabe mencionar que gran parte de las bases de datos que existen son de este tipo, por que tiene una serie de ventajas como es subdividir las base de datos en tablas y por lo tanto se mas pequeñas y así agilizar las búsquedas.

2.1.1 Funciones de las bases de datos

Un sistema manejador de bases de datos realiza, entre otras, las siguientes operaciones:

- Crear nuevos archivos para la base de datos.
- Insertar nuevos datos en los archivos existentes.
- Recuperar datos de archivos existentes.
- Actualizar datos de los archivos existentes.
- Borrar datos de los archivos existentes.
- Borrar los archivos existentes.

Un sistema manejador de bases de datos es el *software* que maneja todos los accesos a la base de datos y proporciona una interfaz al usuario al

sistema.

2.1.2 Ventajas de los sistemas de bases de datos

1. Son compactos.
2. Tienen una alta velocidad de procesamiento.
3. Realizan el trabajo pesado y tedioso.
4. Es fácil estar actualizando.

Las bases de datos proporcionan un control centralizado de los datos de operación. El control centralizado tiene la siguientes ventajas:

1. La redundancia de datos puede ser controlada.
2. Se puede evitar la inconsistencia de los datos.
3. Se pueden compartir los datos.
4. Se pueden establecer estándares en el manejo de datos.
5. Se pueden aplicar restricciones de seguridad. Se puede mantener la integridad de los datos. (Morales López, Enero 1993: 3-5).

2.2 Tipos de bases de datos

2.2.1 Base de datos relacional

Tipo de base de datos o sistema de administración de bases de datos, que almacena información en tablas (filas y columnas de datos) y realiza búsquedas utilizando los datos de columnas especificadas de una tabla para encontrar datos adicionales en otra tabla. En una base de datos relacional, las filas representan registros (conjuntos de datos acerca de elementos separados) y las columnas representan campos (atributos particulares de un registro). Al realizar las búsquedas, una base de datos relacional hace coincidir la información de un campo de una tabla con información en el campo correspondiente de otra tabla y con ello produce una tercera tabla que combina los datos solicitados de ambas tablas. Por ejemplo, si una tabla contiene los campos NÚM-EMPLEADO, APELLIDO, NOMBRE y ANTIGÜEDAD y otra tabla contiene los campos DEPARTAMENTO, NÚM-EMPLEADO y SALARIO, una base de datos relacional hace coincidir el campo NÚM-EMPLEADO de las dos tablas para encontrar información, como por ejemplo los nombres de los empleados que ganan un cierto salario o los departamentos de todos los empleados contratados a partir de un día determinado. En otras palabras, una base de datos relacional utiliza los valores coincidentes de dos tablas para relacionar información de ambas. Por lo general, los productos de

bases de datos para microcomputadoras o microordenadores son bases de datos relacionales. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

Para HISTOCLIN ver. 1.0 se usa una base de datos relacional la cual es Microsoft Access 97, ya que por medio de tablas se realizan sistemas y también se pueden realizar consultas por medio de SQL, lo que resulta un método sencillo de hacer éstas.

2.2.1.1 Normalización

Es el proceso de conversión de una relación en una forma estándar, con el objetivo de evitar la redundancia de la información.(Hansen, 1997: 145)

Para la normalización existen las siguientes formas:

1. **Primera forma normal.** Que los valores de los atributos no pueden ser un conjunto de valores o un grupo repetitivo, es decir , no deben existir atributos no atómicos o multievaluados.
2. **Segunda forma normal.** Se ocupan de las relaciones entre los atributos claves y no claves.
3. **Tercera forma normal.** Establece las normas de las relaciones entre los atributos claves y no claves.
4. **Cuarta forma normal.** Se encarga de los atributos multievaluados

2.2.1.2 Conversión de los diagramas entidad relación a tablas (relacional)

Una base de datos del modelo entidad relación se puede ajustar al modelo relacional de una colección de tablas. Para entidad se realiza una tabla y los atributos se quedan de forma similar. Para HISTOCLIN ver. 1.0 por utilizar un análisis orientado a objetos será necesario un filtro para que sea soportado en modelo relacional que utiliza el Microsoft Access 97.

2.2.2 Bases de datos jerárquico

En un sistema jerárquico los datos se representan en forma de conjuntos de estructuras de arboles. Los operadores para manipular tales estructuras incluyen operadores para recorrer las trayectorias de los arboles hacia arriba y hacia abajo; operadores para localizar un árbol específico; borrar un registro, entre otros.

Un modelo jerárquico tiene las siguientes características:

- Un registro de una base de datos física esta asociado a un solo segmento raíz.
- La raíz puede tener n segmentos hijos.

- Cada hijo tiene m segmentos hijos y así sucesivamente.

2.2.3 Bases de datos de redes

Un modelo de red puede ser considerado como una extensión del modelo jerárquico. La principal diferencia entre ambos modelos es: En la estructura jerárquica un registro hijo tiene exactamente un padre; en la estructura de red, un hijo puede tener cualquier número de padres.

Una base de datos consiste de dos conjuntos, un conjunto de registros y un conjunto de ligas. Mas exactamente, un conjunto de múltiples ocurrencias de cada uno de los diferentes tipos de registros, junto con un conjunto de múltiples ocurrencias de cada una de las diferentes tipos de ligas.

Cada tipo de liga contiene dos tipos de registros, un registro padre y un registro hijo. Cada ocurrencia de un tipo de liga consiste de una ocurrencia del registro padre, junto con un conjunto ordenado de múltiples ocurrencias de los registros hijos.

2.2.4 Bases de datos orientados a objetos

Debido a la cada vez más grande complejidad de los sistemas de

información que los sistemas de bases de datos relacionales ya no pueden resolver, y en base una nueva metodología se creó los sistemas de bases de datos orientados a objetos.

Este sistema hace uso de la programación orientada a objetos y para este tipo de bases de datos se recomienda el modelado de bases de datos conceptuales o también conocidos como el modelo entidad relación.

2.3 Diccionario de datos

Es una base de datos acerca de la terminología que se utilizará en un sistema de información. Para comprender mejor el significado de un diccionario de datos, puede considerarse su contenido como "datos acerca de los datos"; es decir, descripciones de todos los demás objetos (archivos, programas, informes, sinónimos...) existentes en el sistema. Un diccionario de datos almacena la totalidad de los diversos esquemas y especificaciones de archivos, así como sus ubicaciones. Si es completo incluye también información acerca de qué programas utilizan qué datos, y qué usuarios están interesados en unos u otros informes. Por lo general, el diccionario de datos está integrado en el sistema de información que describe. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

Además de ser una excelente herramienta para realizar un sistema de manera ordenada, y así otras personas también lo puedan comprender para que en

dado caso le den mantenimiento al esté.

2.4 Programación estructurada

Se refiere a un tipo de programación que produce código con un flujo limpio, un diseño claro y un cierto grado de modularidad o de estructura jerárquica. Entre los beneficios de la programación estructurada se encuentran la facilidad de mantenimiento y la legibilidad por parte de otros programadores.

También tener la posibilidad de ser modular es muy factible para poder repartirse el trabajo cuando se trate de un proyecto grande, que requiera de varios programadores.

2.5 Programación orientada a objetos

Es un estilo de programación en el que un programa se contempla como un conjunto de objetos limitados que, a su vez, son colecciones independientes de estructuras de datos y rutinas que interactúen con otros objetos. Una clase define las estructuras de datos y rutinas de un objeto. Un objeto es una instancia de una clase, que se puede usar como una variable en un programa. En algunos lenguajes orientados a objetos, éste responde a mensajes, que son el principal medio de

comunicación. En otros lenguajes orientados a objeto se conserva el mecanismo tradicional de llamadas a procedimientos.

2.6 Interfaz de programación de aplicaciones

Es un conjunto de rutinas que utiliza un programa de aplicación para solicitar y efectuar servicios de nivel inferior ejecutados por un sistema operativo informático. Un programa de aplicación efectúa dos tipos de tareas: las relacionadas con el trabajo que se está realizando, por ejemplo aceptar la entrada de texto o de números en un documento u hoja de cálculo, y las relacionadas con las tareas de mantenimiento, como la gestión de archivos y la presentación de la información en la pantalla. Estas tareas de mantenimiento son realizadas por el sistema operativo y la interfaz de programación de aplicaciones (API) proporciona al programa los medios para comunicarse con el sistema, indicándole qué tarea básica del sistema debe realizar y cuándo. En los equipos que funcionan con una interfaz gráfica de usuario (GUI), como el Apple Macintosh, una API también ayuda a los programas de aplicación a gestionar las ventanas, menús, iconos... En las redes de área local (LAN), una API como la NetBIOS de IBM proporciona a las aplicaciones métodos uniformes de solicitar servicios a los niveles inferiores de la red. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

De una manera más sencilla es el intermediario entre funciones de bajo nivel

con las de alto nivel que realiza una computadora.

2.7 Aplicación (informática)

Programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Posee ciertas características que le diferencia de un sistema operativo (que hace funcionar a la computadora), de una utilidad (que realiza tareas de mantenimiento o de uso general) y de un lenguaje (con el cual se crean los programas informáticos). Suele resultar una solución informática para la automatización de ciertas tareas complicadas como puede ser la contabilidad o la gestión de un almacén. Ciertas aplicaciones desarrolladas 'a medida' suelen ofrecer una gran potencia ya que están exclusivamente diseñadas para resolver un problema específico. Otros, llamados paquetes integrados de *software*, ofrecen menos potencia pero a cambio incluyen varias aplicaciones, como un programa procesador de textos, de hoja de cálculo y de base de datos.

Desde al punto de vista anterior HISTOCLIN versión 1.0 es una aplicación, que controla esencialmente las historias clínicas de los médicos haciendo uso de un administrador de base de datos como es Microsoft Access 97.

2.8 Sistema operativo

Es el *software* básico que controla una computadora. El sistema operativo tiene tres grandes funciones: coordina y manipula el *hardware* de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el *mouse*; organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas, y gestiona los errores de *hardware* y la pérdida de datos.

2.8.1 Cómo funciona un sistema operativo

Los sistemas operativos controlan diferentes procesos de la computadora. Un proceso importante es la interpretación de los comandos que permiten al usuario comunicarse con el ordenador. Algunos intérpretes de instrucciones están basados en texto y exigen que las instrucciones sean tecleadas. Otros están basados en gráficos, y permiten al usuario comunicarse señalando y haciendo clic en un icono. Por lo general, los intérpretes basados en gráficos son más sencillos, pero muchos usuarios expertos prefieren los intérpretes de instrucciones basados en texto porque son más potentes.

Los sistemas operativos pueden ser de tarea única o multitarea. Los sistemas operativos de tarea única, más primitivos, sólo pueden manejar un

proceso en cada momento. Por ejemplo, cuando la computadora está imprimiendo un documento, no puede iniciar otro proceso ni responder a nuevas instrucciones hasta que se termine la impresión.

Todos los sistemas operativos modernos son multitarea y pueden ejecutar varios procesos simultáneamente. En la mayoría de los ordenadores sólo hay una UCP; un sistema operativo multitarea crea la ilusión de que varios procesos se ejecutan simultáneamente en la UCP. El mecanismo que se emplea más a menudo para lograr esta ilusión es la multitarea por segmentación de tiempos, en la que cada proceso se ejecuta individualmente durante un periodo de tiempo determinado. Si el proceso no finaliza en el tiempo asignado, se suspende y se ejecuta otro proceso. Este intercambio de procesos se denomina conmutación de contexto. El sistema operativo se encarga de controlar el estado de los procesos suspendidos. También cuenta con un mecanismo llamado planificador que determina el siguiente proceso que debe ejecutarse. El planificador ejecuta los procesos basándose en su prioridad para minimizar el retraso percibido por el usuario. Los procesos parecen efectuarse simultáneamente por la alta velocidad del cambio de contexto.

Los sistemas operativos pueden emplear memoria virtual para ejecutar procesos que exigen más memoria principal de la realmente disponible. Con esta técnica se emplea espacio en el disco duro para simular la memoria adicional necesaria. Sin embargo, el acceso al disco duro requiere más tiempo que el acceso a la memoria principal, por lo que el funcionamiento de la computadora resulta más

lento.

Es lento por que en proceso de emulación se tiene que subir y bajar la información cada vez que se hace la solicitud, y es rápido si se cuenta con bastante espacio en disco de lo contrario los bloques de emulación son pequeños.

2.8.2 Sistemas operativos actuales

Los sistemas operativos empleados normalmente son UNIX, Macintosh OS, MS-DOS, OS/2 , Windows-NT y el popular Windows 95. El UNIX y sus clones permiten múltiples tareas y múltiples usuarios. Su sistema de archivos proporciona un método sencillo de organizar archivos y permite la protección de archivos. Sin embargo, las instrucciones del UNIX no son intuitivas. Otros sistemas operativos multiusuario y multitarea son OS/2, desarrollado inicialmente por Microsoft Corporation e International Business Machines (IBM) y Windows-NT, desarrollado por Microsoft. El sistema operativo multitarea de las computadoras Apple se denomina Macintosh OS. El DOS y su sucesor, el MS-DOS, son sistemas operativos populares entre los usuarios de computadoras personales. Sólo permiten un usuario y una tarea.

2.8.3 Tecnologías futuras

Los sistemas operativos siguen evolucionando. Los sistemas operativos distribuidos están diseñados para su uso en un grupo de ordenadores conectados pero independientes que comparten recursos. En un sistema operativo distribuido, un proceso puede ejecutarse en cualquier ordenador de la red (normalmente, un ordenador inactivo en ese momento) para aumentar el rendimiento de ese proceso. En los sistemas distribuidos, todas las funciones básicas de un sistema operativo, como mantener los sistemas de archivos, garantizar un comportamiento razonable y recuperar datos en caso de fallos parciales, resultan más complejas. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997)

2.9 Microsoft Corporation

Microsoft Corporation es la empresa desarrolladora del administrador de base de datos Microsoft Access 97 en el cual se desarrollo HISTOCLIN versión 1.0.

Es la compañía estadounidense líder en el mercado de *software* para computadoras, con sede en Seattle (Washington). Microsoft fue fundada en 1975 por William H. Gates III y Paul Allen. Ambos se habían conocido durante sus estudios, a través de su afición común a programar con la computadora PDP-10 de Digital Equipment Corporation. En 1975 colaboraron en la primera versión del

lenguaje de programación BASIC para el equipo Altair de MITS, la primera computadora personal. Esto llevó a la formación de Microsoft en Albuquerque, Nuevo México, en ese mismo año. En 1979 Gates y Allen trasladaron la compañía a Redmond. Allen dimitió en 1983 por problemas de salud, pero ahora se ha reincorporado a la Junta Directiva.

La compañía pasó de tener 15 empleados y 500.000 dólares de facturación en 1978 a más de 14.000 empleados y 3.800 millones de dólares de facturación en 1993. Más del 50% de estos ingresos proceden de las ventas internacionales convirtiendo a Microsoft en uno de los principales exportadores de Estados Unidos. En 1981 Microsoft dio su primer paso para diversificarse más allá del mercado de los lenguajes de programación y lanzó al mercado MS-DOS, el sistema operativo para el primer ordenador o computadora personal (PC) de IBM. Microsoft convenció entonces a otros fabricantes de equipos informáticos para distribuir MS-DOS bajo licencia, decisión que lo convirtió en el *software* estándar por excelencia. La colaboración de Microsoft con IBM durante la década de 1980 dio como resultado el primer fenómeno mundial de mercado en masa de la industria de las computadoras, aprovechando el sistema operativo MS-DOS y la disponibilidad de chips y de componentes. La aceptación del MS-DOS como *software* estándar para los PC (hoy en día se encuentra en más de 100 millones de computadoras en todo el mundo) situó a Microsoft en el lugar que ahora ocupa en esta industria. En 1991 Microsoft e IBM finalizaron una década de colaboración cuando decidieron seguir caminos separados en la siguiente generación de sistemas operativos para

computadoras personales. IBM decidió continuar con un antiguo proyecto en común con Microsoft, denominado sistema operativo OS/2, mientras Microsoft decidió desarrollar su sistema operativo Windows. Microsoft ha adaptado el *software* de Windows a todo tipo de computadoras, desde dispositivos portátiles hasta los grandes sistemas multiprocesador. Actualmente Windows está instalado en 15 millones de computadoras de todo el mundo.

Otro aspecto significativo del negocio de Microsoft ha sido su división de *software* de aplicaciones. En 1984 Microsoft era una de las pocas compañías de *software* que desarrollaban programas de aplicación para los equipos Apple Macintosh. El temprano apoyo que Microsoft prestó al Macintosh dio como resultado un tremendo éxito para los programas procesadores de textos y de hoja de cálculo de Microsoft. Cuando Microsoft lanzó Windows al mercado, su sistema operativo gráfico para computadoras personales compatibles con IBM, su experiencia en aplicaciones gráficas para el Macintosh le proporcionó el éxito conseguido en aplicaciones para Windows tales como la hoja de cálculo Microsoft Excel y el programa de tratamientos de textos Microsoft Word. Actualmente estas aplicaciones funcionan de forma similar en Windows y en Macintosh. Microsoft ha entrado en el mercado editorial con Microsoft Press, y desarrolla *software* de bases de datos y productos multimedia.

En un mercado de soporte lógico que crece rápidamente, Microsoft ha sido objeto de acusaciones de prácticas empresariales monopolísticas. En 1990, la Comisión Federal de Comercio estadounidense (FTC, siglas en inglés) comenzó a

investigar a Microsoft por supuestas prácticas contrarias a la libre competencia, pero fue incapaz de llegar a una decisión y cerró el caso. El Departamento de Justicia norteamericano continuó la investigación, que en 1994 desembocó en un acuerdo en el que se pedía a Microsoft que modificase la forma de vender y conceder licencias para sus sistemas operativos a los fabricantes de computadoras. Sin embargo, en febrero de 1995, un juez de distrito estadounidense se negó a aprobar el acuerdo. Tanto Microsoft como el Departamento de Justicia recurrieron a dicha decisión.

En marzo de 1995, Microsoft anunció un proyecto conjunto con DreamWorks SKG para lanzar nuevos programas de ocio multimedia interactivos. En 1995, Microsoft lanzó la Red Microsoft, un servicio en línea de Internet que proporciona acceso a Internet y una gran variedad de contenido, como noticias, información de interés especial o material de referencia. También en 1995, Microsoft y la cadena de televisión NBC se asociaron para crear MSNBC, una cadena de cable con noticias, coloquios e informaciones durante las 24 horas y un servicio auxiliar de Internet. Según las previsiones, MSNBC debía comenzar sus actividades en julio de 1996. En 1996, Microsoft reorganizó algunas de sus divisiones para mejorar su capacidad de desarrollar productos y tecnologías para Internet. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

Microsoft enfrenta hoy en día una serie de demandas tanto de su competencia como de sus consumidores por el mal desempeño de sus productos, por lo que son innumerables las paginas Web que están en contra de Microsoft.

2.10 Windows

Es un entorno multitarea dotado de una interfaz gráfica de usuario, que se ejecuta en computadoras diseñadas para MS-DOS. Windows proporciona una interfaz estándar basada en menús desplegables, ventanas en pantalla y un dispositivo señalador como el *mouse* (ratón). Los programas deben estar especialmente diseñados para aprovechar estas características. (Enciclopedia Microsoft® Encarta® 98 © 1993-1997).

2.11 Interfaz gráfica de usuario

Es el tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón. (Véase Interfaz de usuario).

Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con el ordenador o computadora. Esto hace que el programador pueda concentrarse en la funcionalidad, ya que no

está sujeto a los detalles de la visualización ni a la entrada a través del ratón o del teclado. También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, porque la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo. Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios.

2.12 Multitarea

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitarea. La conmutación de contextos (*context switching*) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que sólo se está procesando la aplicación que se encuentra en primer plano (la que ve el usuario). Para activar otra tarea que se encuentre en segundo plano, el usuario debe traer al primer plano la ventana o pantalla que contenga esa aplicación. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentre en primer

plano (por ejemplo, cuando esta aplicación está esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo. Para mantener el sistema en orden, cada tarea recibe un nivel de prioridad o se procesa en orden secuencial. Dado que el sentido temporal del usuario es mucho más lento que la velocidad de procesamiento de la computadora, las operaciones de multitarea en tiempo compartido parecen ser simultáneas.

Pero realmente lo que sucede es que se decide cierto espacio de tiempo para cada tarea que tiene en uso en usuario en forma simultánea y por ser fracciones de segundo el usuario no logra percatarse de lo que realmente está sucediendo en la UCP.

En los últimos años se están haciendo más populares los sistemas operativos con soporte multimedia y por lo tanto cada día existen en el mercado mas aplicaciones y programa que soportan la multitarea.

2.13 Ventana

En aplicaciones informáticas e interfaces gráficas de usuario, una parte de la pantalla que puede contener su propio documento o mensaje. En programas basados en ventanas, la pantalla puede dividirse en varias ventanas, cada una de

las cuales tiene sus propios límites y puede contener un documento diferente (o una presentación distinta del mismo documento). Cada ventana puede contener su propio menú u otros controles, y el usuario puede ampliarla o reducirla mediante un dispositivo señalador (puntero), que se acciona con el ratón o *mouse*.

Un entorno basado en ventanas es un sistema que presenta al usuario distintas ventanas, como por ejemplo el Finder de los equipos Apple Macintosh, Microsoft Windows y el OS/2 Presentation Manager.

HISTOCLIN versión 1.0 hace uso de las ventanas para la presentación de su información, informes y consultas. Por que hace uso de Microsoft Windows 95 y toda su organización se realiza a través de ventanas.

Los atributos con que cuenta casi toda ventana de Windows son los siguientes:

- ✓ Botón de Maximizar.
- ✓ Botón de minimizar.
- ✓ Botón de cerrar.
- ✓ Capacidad de moverse.
- ✓ Interactuar entre varias ventanas.
- ✓ Barras de desplazamiento.

CAPÍTULO TRES

En este capítulo de la tesis se enfatiza en los métodos para el desarrollo de sistema de información, sus ventajas y también cuál utilizar dependiendo del tipo de sistema o programa que se tenga que desarrollar. En forma importante se menciona los tipos de sistemas de información, componentes de los sistemas de información y las fuerzas de diseño de los sistemas de información.

3.1 CONCEPTOS DE ANALISIS Y DISEÑO DE SISTEMAS.

Los sistemas de información son desarrollados con propósitos diferentes dependiendo de las necesidades del negocio. Los sistemas de procesamiento de transacciones (TPS por sus siglas en inglés) funcionan al nivel operacional de la organización, los sistemas de automatización de oficina (OAS por sus siglas en inglés) que dan cabida al trabajo a nivel del conocimiento. Los sistemas de mas alto nivel incluyen a los sistemas de apoyo a decisiones (DSS por sus siglas en inglés) así como en los sistemas de información gerencial (MIS por sus siglas en inglés).

Los sistemas expertos aplican la experiencia de los tomadores de decisiones para resolver problemas especificos estructurados. Al nivel estratégico de la administración encontramos sistemas de apoyo a ejecutivos (ESS por sus siglas en inglés) y los sistemas de apoyo a las decisiones de grupo (GDSS por sus siglas en inglés) ayudan a la toma de decisiones al mismo nivel, en una forma sin estructura o

semiestructurada.

La base del funcionamiento de un sistema de información integral, debe estar sustentado en un sistema de procesamiento de transacciones, por que ahí es donde se recaba toda la información por los empleados de un bajo nivel jerarquico y se da pie a los demás tipos de sistemas.

3.2 TIPOS DE SISTEMAS DE INFORMACIÓN

3.2.1 Sistemas de procesamiento de transacciones.

Los sistemas de procesamiento de transacciones (TPS) son sistemas de información computarizados desarrollados para procesar una gran cantidad de datos para transacciones rutinarias de los negocios, tales como nómina e inventario. Los TPS eliminan el tedio de las transacciones operacionales necesarias y reducen el tiempo que alguna vez se requirió para ejecutarlas manualmente, aunque la gente todavía debe alimentar datos a los sistemas computarizados.

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 por el TPS para información al minuto acerca de lo que está pasando en sus empresas, es esencial para las operaciones diarias que estos sistemas funciones lentamente y sin interrupción.

3.2.2 Sistemas de automatización de oficina y sistemas de manejo de

conocimiento

Al nivel de conocimientos de la organización hay dos clases de sistemas. Los sistemas de automatización de oficina (OAS) que dan soporte a los trabajadores de datos, quienes, por lo general, no crean un nuevo conocimiento sino que usan la información para analizarla y transformar datos, o para manejarla en alguna forma y luego compartirla y diseminarla formalmente por toda la organización y algunas veces mas allá de ella. Los aspectos familiares de los OAS incluyen procesamiento de palabras, hojas de calculo, editor de publicaciones, calendarización electrónica y comunicación mediante el correo de voz, correo electrónico y videoconferencias.

Los sistemas de manejo de conocimiento (KWS) dan soporte a los trabajadores profesionales, tales como científicos, ingenieros y doctores, les ayudan a crear un nuevo conocimiento que contribuya a la organización o a toda la sociedad.

3.2.3 Sistemas de información gerencial

Los sistemas de información gerencial (MIS) no reemplazan a los sistemas de procesamiento de transacciones, sino que todos los MIS incluyen procesamiento de transacciones. Los MIS son sistema de información computarizada que trabajan debido a la interacción resuelta entre gentes y computadoras. Requieren que las gentes, el *software* (programas de computadora) y el *hardware* (computadoras, impresoras, etc.) trabajen a unísono. Los sistemas de información dan soporte a un espectro mas amplio de tareas organizacionales que los sistemas de procesamiento

de transacciones, incluyendo el análisis de decisiones y la toma de decisiones.

Para poder ligar la información, los usuarios de un sistema de información gerencial comparten una base de datos común. La base de datos guarda modelos que ayudan a los usuarios a interpretar y aplicar esos mismos datos. Los sistemas de información gerencial producen información que es usada en la toma de decisiones. Un sistema de información gerencial también puede llegar a unificar alguna de las funciones de información computarizada, aunque no exista como una estructura singular en ningún lugar del negocio.

3.2.3 Sistema de apoyo a decisiones

Una clase de mas alto nivel en los sistemas de información computarizada son los sistemas de apoyo a decisiones (DSS). El DSS es similar al sistema de información gerencial tradicional en que ambos dependen de una base de datos como fuente. Un sistema de apoyo a decisiones se aparta del sistema de información gerencial tradicional en el que enfatiza el apoyo a la toma de decisiones en todas sus fases, aunque la decisión actual es todavía del dominio del tomador de decisiones. Los sistemas de apoyo a decisiones están hechos a la medida de la persona o a grupo que los usa que los sistemas de información gerencial tradicional.

3.2.4 Sistemas expertos e inteligencia artificial

La inteligencia artificial (AI por sus siglas en inglés) puede ser considerada

como la meta de los sistemas expertos. El empuje general de la AI ha sido desarrollar máquinas que se comporten de forma inteligente. Dos caminos de la investigación de la AI son la comprensión del lenguaje natural y el análisis de la habilidad para razonar un problema y llegar a conclusiones lógicas. Los sistemas expertos usan los enfoques del razonamiento de la AI (inteligencia artificial) para resolver los problemas que le plantean los usuarios.

Los sistemas expertos son un caso muy especial de un sistema de información, cuyo uso ha sido factible para los negocios a partir de la reciente y amplia disponibilidad de *hardware* y *software* tal como las microcomputadoras y sistemas expertos. Un sistema experto (también llamado un sistema basado en el conocimiento) captura en forma efectiva y usa el conocimiento de un experto para resolver un problema en particular experimentando en una organización. Observe que la diferencia del DSS, que deja la decisión final al tomador de decisiones, un sistema experto selecciona la mejor solución a un problema o a una clase específica de problemas.

Los componentes básicos de un sistema experto son la base de un conocimiento, una máquina de inferencia que conecta al usuario con el sistema, procesando preguntas por medio de lenguajes tales como SQL (lenguaje de consultas estructurado), y la interfaz del usuario. Gente llamada ingenieros del conocimiento capturan la experiencia de los expertos, construyen un sistema de computadora donde incluyen el conocimiento del experto y luego lo implementan. Es totalmente posible que la construcción e implementación de sistemas expertos sea el trabajo futuro de muchos analistas de sistemas.

En la industria mexicana de software es muy poco el desarrollo en este campo de la informática y los pocos sistemas que existen son para los grandes corporativos y los negocios de finanzas.

3.2.5 Sistemas de apoyo a decisiones de grupo

Cuando los grupos necesitan trabajar juntos para la toma de decisiones semiestructuradas o sin estructura, un sistema de apoyo a decisiones de grupo puede plantear una solución. Los sistemas de apoyo a decisiones de grupo (GDSS) son usados en cuartos especiales, equipados en varias configuraciones diferentes, que permiten que los miembros del grupo interactúen con apoyo electrónico, frecuentemente en forma de *software* especializado y con una persona que da facilidades al grupo. Los sistemas para decisiones de grupos están orientados para reunir a un grupo, a fin que resuelva un problema con la ayuda de varios apoyos como votaciones, cuestionarios, aportaciones de ideas y creación de escenarios. El *software* GDSS puede ser diseñado para minimizar el comportamiento negativo típico de un grupo, tal como la falta de participación debido al miedo a represiones por expresar un punto de vista no popular y conflictivo, denominación por miembros del grupo con voz dominante y la toma de decisiones de pensamiento en grupo.

Alguna veces los GDSS son tratados bajo el termino más general trabajo colaborativo apoyado por computadoras (CSCW por sus siglas en inglés), que puede incluir el apoyo de *software* llamado "grupware" para la colaboración en un equipo

por medio de computadoras en red.

3.2.6 Sistemas de apoyo a ejecutivos

Cuando los ejecutivos se acercan a la computadora, frecuentemente están buscando formas que le ayuden a tomar decisiones a nivel estratégico. Un sistema de apoyo a ejecutivos (ESS por sus siglas en inglés) ayuda a éstos, para organizar sus interacciones con el ambiente externo, proporcionando apoyo de gráficos y comunicaciones a lugares accesibles tales como salas de juntas u oficinas personales corporativas. Aunque el ESS se apoyan en la información generada por los TPS y los MIS, los sistemas de apoyo a ejecutivos ayudan a sus usuarios a que toquen problemas de decisión sin estructura, que no son específicos de una aplicación, creando un ambiente que ayude a pensar acerca de los problemas estratégicos de una manera informada. Los ESS extienden y dan apoyo a las capacidades de los ejecutivos para encontrar sentido a sus ambientes. (Kendall & Kendall, 1997: 2 a la 5).

3.3 COMPONENTES ESTRUCTURALES DE LOS SISTEMAS DE INFORMACIÓN

Sin importar las organizaciones a las que sirven o la forma o la forma en que se desarrollan y diseñan, todos los sistemas de información están compuestos de

los siguientes seis componentes estructurales: entrada, modelos, salida, tecnología, base de datos y controles. Estos componentes estructurales pueden tomar diferentes formas, valores y contenido; pueden parecer diferentes y trabajar en forma diferente; algunos pueden soportar sistemas bien diseñados; otros pueden soportar sistemas diseñados con deficiencia; algunos pueden ser imperfectos; algunos pueden ser altamente sofisticados, todo ello es irrelevante. Estos son los seis componentes estructurales básicos de todos los sistemas de información. El que tan bien se combinen y el tipo de sistema de información que resulte depende del diseñador, que es el arquitecto de los sistemas. Así como un estuche para construcción de juguetes conteniendo unos cuantos componentes básicos que, con adecuada dosis de imaginación pueden transformarse en camiones, aviones, molinos de viento, ruedas de la fortuna, puentes, e incluso robots y dinosaurios, los componentes estructurales de sistemas de información pueden conjuntarse para obtener sistemas de información funcionales que satisfagan las necesidades de las organizaciones y sus usuarios. La comprensión de estos componentes estructurales, sus relaciones y acoplamientos y su contenido lógico y físico, proporciona los conocimientos básicos para describir, desarrollar y diseñar sistemas de información. En este momento, y simplemente con fines de definición, se estudiarán brevemente los componentes estructurales. Mas adelante se hará una presentación detallada de cada uno de los componentes estructurales.

3.3.1 Bloque de entrada

La entrada representa todos los datos, textos, voz e imágenes que entran al sistema de información y los métodos y los medios por lo cuales se capturan e introducen. La entrada está compuesta de transacciones, solicitudes, consultas, introducciones y mensajes. Por lo general, la entrada sigue un protocolo y un formato para que el contenido, la identificación, la autorización, el arreglo y el procesamiento sean adecuados. La introducción puede hacerse mediante escritura manual, formas en papel, reconocimientos de características físicas como geometría manual y huellas digitales, teclados, bastones de mando, gatos, ratones, voz, sensores táctiles, y caracteres y códigos ópticos y magnéticos.

En la actualidad, los medios mas comunes para la introducción de transacciones y textos son las lectoras de códigos de barras y láser y el teclado, respectivamente. Con frecuencia se puede conseguir eficiencia en la entrada combinando los métodos. Por ejemplo, la entrada 'por voz puede complementarse mediante un teclado para proporcionar comandos como introducir, archivar o pararse, para dar entrada a códigos numéricos, o para la entrada de mensajes. Una forma aún mejor sería desarrollar sistemas de entrada de voz como una alternativa viable al teclado. Con la entrada por voz, el cuello de botella durante la entrada se reduciría sustancialmente. El usuario probablemente trabaja mejor mediante entrada por voz mediante teléfono, por ejemplo, es altamente eficiente debido a que elimina los pasos intermedios de anotar y volver a teclear mensajes. De

hecho, una de las principales tendencias en la siguiente década será hacia los sistemas de reconocimiento de voz y escritura manual, tanto fijos, como portátiles. También, una diversidad de dispositivos con entrada eficiente, como las pantallas sensibles al tacto, responden directamente a la presión de los dedos. También se cuenta con plumas luminosas, gatos y ratones, para ser empleados con visualización por pantalla y gráficas.

3.3.2 Bloque de modelos

Este componente consta de modelos lógico-matemáticos que manipulan de diversas formas la entrada y los datos almacenados, para producir los resultados deseados de salida. Un modelo lógico-matemático puede combinar ciertos elementos de datos para proporcionar una respuesta adecuada a una consulta, o puede reducir o agregar volúmenes de datos para obtener un reporte conciso. Puede ser tan simple como:

$$\text{Ganancias} = \text{Ingresos} - \text{Gastos}$$

El componente del modelo también contiene una descripción de algunas de las técnicas de modelado más populares empleadas por los analistas de sistemas para diseñar y documentar las especificaciones de los sistemas. Estas técnicas incluyen tablas y árboles de decisión, diagramas de flujo tradicionales, diagramas

Nassi-Shneiderman, HIPO, diagramas de estructura, diagramas Warnier-Orr y el empleo de prototipos.

3.3.3 Bloque de salida

El producto del sistema de información es la salida información de calidad y documentos para todos los niveles de la gerencia y para todos los usuarios dentro y fuera de la organización. La salida es, en gran medida, el componente que guía e influye en los otros componentes. Si el diseño de este componente no satisface las necesidades del usuario, entonces los otros componentes tienen poca importancia.

La salida representa el otro extremo de la entrada y obviamente no puede ser mejor que la entrada y los modelos empleados para producirla. Con frecuencia, la entrada y la salida son interactivas. La entrada se convierte en salida; la salida se convierte en entrada. La bocina de un teléfono es un dispositivo de entrada; el auricular es un dispositivo de salida. El teclado de una máquina de escribir introduce datos; los tipos de impresión producen salida en papel. Un usuario hace una consulta mediante una unidad de despliegue visual (VDU) y obtiene una respuesta; con base en esta respuesta, se hace otra consulta.

De manera lógica, la salida está compuesta de elementos tales como estados financieros, facturas, órdenes de compra, cheques de pago, reportes de presupuestos, respuestas a consultas, mensajes, órdenes, resultados de una toma

de decisiones programada, escenarios y simulaciones, y reglas de decisión. La calidad de esta salida se basa en su exactitud, oportunidad y relevancia. Además esta salida debe tratarse en función de su destino, uso, frecuencia de uso y seguridad.

El estudio anterior se ocupó del aspecto referente al mensaje ¿Pero qué hay acerca del medio?. La salida se puede producir en pantallas, impresoras, dispositivos de audio o microfilm. En el pasado, el principal medio de salida era pilas de reportes, impresos, denominados listados de computadora. En la actualidad las personas están eligiendo otras formas de salida que vayan más de acuerdo a sus gustos, como gráficas, vídeo y audio. Debido a que la mayoría de los trabajadores pasan entre el 50 y 90 por ciento de su tiempo manejando información y comunicándose, están buscando, de hecho, una gama más amplia de medios que se aproximen a los métodos o manos naturales de emitir información y de comunicarse con el sistema y entre sí.

El empleo de pantallas grandes planas para exhibición en la pared se está volviendo muy popular, especialmente en donde se están utilizando gráficas y videoconferencias. También están creciendo en popularidad las pantallas divididas y otras técnicas de manipulación y presentación de la información para hacer que la pantalla se parezca más a un escritorio y aun archivero conteniendo muchos documentos y reportes diferentes. Otros métodos populares de salida incluyen a los graficadores y equipos de fotocomposición. Como puntos intermedios entre el papel y las pantallas para la presentación de la información están los microfilms y

las microfichas. También, por supuesto los sistemas de respuestas por voz y correo electrónico son formas ideales para la salida de mensajes; estas salidas ciertamente son del gusto de la mayoría de la gente debido a su atractivo y conveniencia.

3.3.5 Bloque de tecnología

La tecnología es la "caja de herramientas" del trabajo en sistemas de información. Captura la entrada, activa los modelos, almacena y accesa datos, produce y transmite salida, y ayuda a controlar todo el sistema. Hace todo el trabajo pesado y une a todos los componentes estructurales. La tecnología consta de tres componentes principales: la computadora y el almacenamiento auxiliar, las telecomunicaciones y el *software*. Las telecomunicaciones comprenden el empleo de medios electrónicos y de transmisión de luz para la comunicación entre nodos a lo largo de una distancia. El *software* corresponde a los programas que hace que funcione el *hardware* de la computadora y le dan instrucciones sobre la forma de procesar los modelos. El *hardware* está compuesto de una variedad de dispositivos que proporcionan el soporte físico para los componentes estructurales. Por ejemplo, para una terminal sirve como dispositivo de entrada para las transacciones contables; una unidad central de procesamiento (CPU) acciona a los modelos contables con los datos apropiados las impresoras apropiadas a varias divisiones a lo largo de todo el país y conectadas a la CPU mediante satélites y estaciones

terrestres producen como salida estados contables; un disco magnético en la base de datos almacena los archivos maestros de contabilidad, los diarios y los libros; y un dispositivo codificador y cifrador ayuda a controlar la confidencialidad de la contabilidad y demás información sensible a medida que se trasmite y también mientras se almacena en la base de datos. A decir verdad el problema de la escasez tecnológica ha sido reemplazado por el problema de elegir la tecnología correcta para el trabajo que necesita realizarse. Para los sistemas de información actuales se puede aplicar una gama más amplia de soluciones en vez de sólo una.

En su esencia misma, la tecnología es un sustituto del esfuerzo humano. De los seis componentes estructurales, la tecnología es la más evidente. La mayoría de los sistemas de información actuales y del futuro estarán basados en la tecnología. Sin embargo, un peligro latente es el de un sobreencantamiento con la tecnología en tanto se descuidan las necesidades de información de los usuarios. Otro peligro es el de suponer que la adquisición de una computadora y su tecnología relacionada es equivalente a implementar un sistema de información. Nada puede estar más alejado de la verdad. Una computadora por sí misma no hace un sistema de información. A decir verdad, la gerencia ha efectuado muchos trabajos inútiles en que, sin pensar previamente y sin un mínimo de análisis de sistemas han adquirido computadoras de un millón de dólares y luego le han dicho al personal de la organización que las "use". No es de sorprender que muchas computadoras adquiridas en tales circunstancias hayan sido abandonadas sólo para llenarse de polvo. Además, la profesión de sistemas ha tenido una buena

cantidad de chapuceros tecnológicos y personas extravagantes que tienen poca consideración de las necesidades de los usuarios. No obstante, por el otro lado de la moneda se encuentra un buen número de conceptos sobre sistemas de información que seguirían siendo solamente conceptos sino hubiera sido por la llegada de la tecnología. Algunos ejemplos son los dispositivos de captura en el punto de venta y un despliegue de equipo de telecomunicaciones, los cuales efectivamente reducen el tiempo y el espacio.

3.3.5 Bloque de base de datos

La base de datos es el lugar en donde se almacenan todos los datos necesarios para atender las necesidades de todos los usuarios. Nuevamente, los datos pueden ser una combinación de voz, imágenes, texto y números. La base de datos considera desde dos puntos de vista, el físico y el lógico. La base de datos física está compuesta de los medios de almacenamiento, como las cintas, discos, disquetes, casetes, tarjetas magnéticas, pastillas (chips) y microfilms. Esta es la forma en que los datos se almacenan realmente. Sin embargo otro problema probablemente más importante es cómo buscar, asociar y recuperar los datos almacenados para satisfacer necesidades específicas de información. Esto, por supuesto, es el lado lógico de la base de datos y, si está estructurada correctamente, asegura la recuperación oportuna, relevante y exacta de la información. También tiene que ver con el componente de *software* del sistema e

incluye técnicas lógicas y asociativas de datos como índices, directorios, listas, llaves, apuntadores, redes, árboles y relaciones.

3.3.6 Bloque de controles

Todos los sistemas de información están sujetos a una diversidad de peligros y amenazas como desastres naturales, incendios, fraude, fallas de los sistemas, errores y omisiones, interceptación secreta, deficiencias, sabotajes y multiplicaciones maliciosas. En muchos casos, sin embargo, los peores abusos del sistema provienen de procedimientos operacionales inadecuados, empleados incompetentes y una administración pobre. Algunos de los controles que necesitan diseñarse en el sistema para asegurar su protección, integración y operación uniforme son la instalación de un sistema de administración de registros, la aplicación de controles contables tradicionales, el desarrollo de un plan maestro de sistemas de información, la creación de un plan de contingencias la aplicación de procedimientos para el personal, como verificación de antecedentes, capacitación, rotación de tareas, vacaciones obligatorias, etc., la preparación de una documentación completa y actualizada, la aplicación de monitores de *hardware* y *software*, el establecimiento de sistemas de respaldo y almacenamiento fuera de las instalaciones, la instalación de sistemas ininterrumpidos de energía y sistemas contra incendio, el empleo de adecuados procedimientos de programación y

controles, y la aplicación de una diversidad de procedimientos de seguridad, dispositivos y controles de acceso.

3.4 FUERZAS DE DISEÑO DE LOS SISTEMAS DE INFORMACIÓN

Leonardo da Vinci, el gran pintor florentino, escultor, arquitecto e ingeniero, tuvo el apoyo para crear y diseñar sin límites una diversidad de artefactos. Él no tenía que satisfacer el mercado, satisfacer necesidades caprichosas de usuarios, o preocuparse de costo y programas; tenía suficiente apoyo financiero para seguir hacia donde sus instintos creativos lo condujeran. Muchas de sus creaciones que alguna vez fueron considerados inútiles actualmente son consideradas como obras maestras.

Las personas que tienen la responsabilidad del desarrollo y diseño de automóviles, puentes, edificios, sistemas de información, etc., no pueden darse el lujo de da Vinci para "hacer lo que quieren". Están, en gran medida, influenciados y limitados por un número de fuerzas de diseño. Para tener éxito, los diseñadores deben de determinar cuáles son las fuerzas del diseño, cómo afectan a sus proyectos de diseño y guiarse por ellas, incorporando al mismo tiempo creatividad e innovación en su trabajo. Es, a decir verdad, un acto complicado de equilibrio.

Utilicemos el diseño de automóviles como una analogía. Todos los automóviles constan de los mismos componentes estructurales: carrocería,

interiores, instrumentos y controles de manejo; ruedas y ejes; un tren motriz compuesto de una unidad de potencia, una fuente de energía, una transmisión y un diferencial. Sin embargo, debido a un número de fuerzas de diseño de los automóviles, la forma y sustancia de estos componentes estructurales han combinado con el tiempo. Por ejemplo, los nuevos controles de seguridad y contaminación, y los requerimientos de economía de combustible han obligado a un rediseño completo de los automóviles. Adicionalmente, algunos fabricantes de automóviles hace varios años ponían poca atención a los cambios radicales de gustos en el mercado y diseñaban automóviles inaceptables para una gran número de consumidores. Sólo después de que estos fabricantes dejaron de diseñar automóviles de forma aislada y empezaron a diseñarlo de acuerdo a fuerzas de diseño, fue como recuperaron la pista y lograron recapturar una gran porción de mercado. A decir verdad, el tener conciencia de las fuerzas del diseño y su consideración, hizo que estos fabricantes regresaran a las operaciones con utilidad.

Los diseñadores de sistemas de información deben de considerar 10 fuerzas de diseño que afectan su trabajo: integración, interfaz usuario/sistema, requerimientos de procesamiento de datos, factores organizacionales, requerimientos costo - beneficio, factores humanos y requerimientos de factibilidad. Los diseñadores deben definir en detalle y en cada situación las fuerzas del diseño y determinar el nivel de impacto y su interacción entre sí, y en ocasiones su conflicto.

3.4.1 Integración

Los sistemas de información tendrán que diseñarse con un acoplamiento más estrecho entre la oficina y la planta. A decir verdad, el sistema de información llegara hacer tan importante en la planta de la fabrica como en la oficina. Además, la conectividad y las comunicaciones intra y entre los departamentos deben ser mejores dentro de la oficina y de la fábrica. La tecnología informática estará inserta en las organizaciones y enlazada para una sincronización completa y una coordinación de las operaciones. El sistema ya no estará separado funcional y especialmente del lugar de trabajo. Este diseño dará por resultado una malla de información para la organización. La mano derecha sabrá lo que está haciendo la izquierda y viceversa. Esta malla de información unirá a las funciones.

El método tradicional de diseñar piezas metálicas por ejemplo, ha sido el crear una serie de dibujos en papel, desarrollar un prototipo y luego tomar decisiones de producción que conduzcan a la fabricación de un lote específico de piezas por un grupo de máquinas – herramientas. En el pasado, las actividades del diseñador y la de los planificadores y programadores de producción han sido distintas y en consecuencia, la secuencia de manufactura no ha estado integrada. Los desarrollos de la manufactura integrada por computadora (CIM), y en especial los robots programables, están obligando a una mayor integración entre los departamentos de planeación, programación, contabilidad de costos, control de inventarios y comercialización. Con el incremento de celdas de manufactura, robots

y sistemas de manufactura flexible, se están volviendo imperativos los ciclos de retroalimentación en línea y la coordinación con todos los componentes de la organización.

Los robots se utilizarán cada vez más debido a que (1) aumentan la productividad y la calidad de la salida; (2) eliminan las tareas tediosas, peligrosas y sucias; (3) tienen la habilidad para explorar y trabajar en ambientes peligrosos y hostiles (p. ej., exploración submarina, minería y manufactura con base en la luna, reparación de satélites); (4) se beneficiarán de la creciente economía de la microelectrónica; y (5) debido a su capacidad de programación pueden ser utilizados en una diversidad de procesos de manufactura. A decir verdad, los robots y los sistemas de información trabajan bien en conjunto, y a medida que los robots se vuelven más "inteligentes" y asumen más tareas, especialmente en áreas remotas, será mayor la necesidad de información más oportuna y mejores ciclos de retroalimentación. Adicionalmente, la integración de los robots y CIM con los sistemas de información aumenta la flexibilidad de la manufactura, dando de esta manera a una organización la habilidad para modificar el diseño de los productos en forma barata y producir eficientemente una mezcla de productos más diversa. Esta habilidad representa un avance significativo debido a que unas fabricas son productoras masivas eficientes pero inflexibles; otras fabricas son productoras de lotes flexibles pero ineficientes. Para sobrevivir en la década de los noventas, las fabricas deberán ser tanto flexibles como eficientes.

3.4.2 Interfaz usuario / sistema

Tarde o temprano, la información tiene que fluir por el cerebro humano para tener algún valor. Por lo tanto, entre mejor sea la interfaz entre el usuario y el sistema, sin obstrucción, interferencia externa o dependencia de intermediarios, mejor será el flujo de información. La riqueza de la interfaz usuario / sistema dependerá de la fuerza y variedad de las capacidades de los componentes estructurales. Los esquemas de entrada le permitirán al usuario comunicarse con el sistema a través de una combinación de preguntas y respuestas, comandos, teclas de función, menús, cajas de diálogo y verificación, todos ellos introducidos por medio de lenguajes naturales y voz, tableros táctiles, bastones de mando, teclados, etc. Los modelos ayudaran en las tareas de transacciones, el trabajo de oficinas, el diseño, la producción y la administración. Los modelos se acoplan con la base de datos, la cual servirá más como una base de conocimientos de material fuente especialmente durante los diálogos gerenciales. La salida además de satisfacer las necesidades de información de los usuarios atraerá y servirá a los sentidos del oído, la vista y el tacto. La tecnología reducirá el tiempo y el espacio entre el usuario y el sistema. Los controles les proporcionaran a los usuarios autorizados un sistema confiables y todo el acceso que necesitan para realizar sus tareas o sostener un dialogo con el sistema.

3.4.3 Fuerzas competitivas

Actualmente, las organizaciones están entrando a una era de competencia feroz doméstica y global, un mundo de cambio rápido y significativo que demanda un flujo mejor y más oportuno de información de calidad. Para que las organizaciones sobrevivan y prosperen en el mundo del mañana deberán diseñar sistemas que soporten y mejoren la actividad gerencial, la diferenciación de productos y servicios.

3.4.4 Calidad y utilidad de la información

Obviamente, una de las fuerzas principales que afectan el diseño del sistema de información se deriva de los requerimientos específicos de información del usuario. Un usuario quiere un estado de flujo de efectivo, otro quiere conocer las ventas por territorio y las ventas por cliente de este mes, otro desea hacer varias consultas al sistema acerca del estado del inventario y otro mas desea poder formular varias preguntas del tipo "que tal si". En la medida en que se pueden identificar los requerimientos de información de un individuo es posible proporcionar la información relevante necesaria para satisfacer estos requerimientos. Sin embargo, las formas en que los datos pueden volverse o convertirse en información son casi tan numerosas como las situaciones específicas que se pueden identificar.

La producción de la información implica conocimientos tan sencillos como comunicar directamente a un receptor datos previamente capturados, o tan complejos como el desarrollo de un modelo matemático sofisticado. Adicionalmente, los otros componentes estructurales se deben diseñar para trabajar en armonía y asegurar que la información sea exacta y oportuna. Además de estos atributos, la información debe ser percibida por los receptores como útil.

3.4.5 Requerimientos de sistemas

Los requerimientos de sistemas son los requerimientos operacionales inherentes al sistema de información mismo y surgen de o están influenciados por otras fuerzas del diseño o requerimientos, especialmente la producción de información de calidad. Estos requerimientos son (1) confiabilidad, (2) disponibilidad, (3) flexibilidad, (4) programa de instalación, (5) expectativa de vida y potencial de crecimiento, y (6) capacidad para recibir mantenimiento.

1. **Confiabilidad.** La confiabilidad se refiere al grado de seguridad con que un recurso realiza su función, produciendo los mismos resultados en procesos sucesivos.
2. **Disponibilidad.** La disponibilidad significa que el sistema es accesible a los usuarios. Un sistema puede ser confiable, pero no estar disponible

cuando, por ejemplo, está siendo probado o cuando se le está agregando un componente. Por otra parte, un sistema puede estar disponible pero no ser confiable.

3. **Flexibilidad.** El requerimiento de flexibilidad se refiere a la habilidad del sistema para cambiar o adaptarse para satisfacer los requerimientos cambiantes de los usuarios.
4. **Programa de Instalación.** El programa de instalación comprende el espacio de tiempo existente entre el momento en que una organización reconoce una necesidad y el momento en que implementa la solución. Supuestamente, aunque no siempre, entre más tiempo se requiera para diseñar un sistema, mejor será el diseño. Por ejemplo, si a uno le dieran cinco años y suficientes recursos para diseñar un sistema, uno podría diseñar el sistema "perfecto". Normalmente uno no puede darse este lujo y tiene que diseñar el mejor sistema que pueda dentro del tiempo asignado.
5. **Expectativa de vida y potencial de crecimiento.** Algunos sistemas no cuentan con una expectativa de vida debido a que ya son obsoletos, en el momento en que se implementan. O bien, un sistema puede instalarse y trabajar muy bien durante cierto tiempo, pero debido a que es un sistema que tiene una sola salida sin capacidad de crecer, queda "patas pa' arriba" cuando aumentan las necesidades de los usuarios. Por lo tanto, los sistemas deben diseñarse para satisfacer requerimiento durante un

tiempo razonable (p. ej., cinco años) y ser también capaces de crecer si las necesidades cambian de manera significativa.

6. **Capacidad para recibir mantenimiento.** Una vez que el sistema se implanta, debe recibir mantenimiento, debido a que se deben corregir fallas, a que se deben satisfacer solicitudes especiales ya que deben efectuarse mejoras generales a los sistemas. La pregunta no es si uno tendrá o no que dar mantenimiento al sistema. Uno lo tendrá que hacer. La meta, por lo tanto, deberá ser la de diseñar sistemas que sean capaces de recibir mantenimiento mediante el empleo de nombres de datos estándar y lenguajes de programación, programación estructurada y modular, configuraciones estándar y procedimientos estándar de documentación.

3.4.6 Requerimientos de procesamientos de datos

Los requerimientos de procesamientos de datos se refieren al trabajo de detalle del sistema y se dividen en cuatro categorías: (1) el volumen de datos involucrados, (2) la complejidad de las operaciones de procesamientos de datos, (3) las restricciones de tiempo de procesamiento, y (4) las demandas computacionales.

1. **Volumen.** El volumen se refiere a la cantidad de datos que deben procesarse en un periodo dado para lograr una meta de la información. Una forma de cuantificar el volumen podría ser la de hacer referencia a las transacciones organizacionales (p. ej., tarjetas de tiempo, facturas, transacciones del inventario, elementos del presupuesto). Otra medida podría ser la de hacer referencia a las funciones de procesamiento (p. ej., 14 actualizaciones de archivo por registro, 300 registros maestros, 26 líneas por página).
2. **Complejidad.** La complejidad se refiere al número de operaciones de datos, intrincadas e interrelacionadas, que deben realizar para lograr una meta de la información. Por ejemplo, el procesar correctamente una nómina es necesario calcular correctamente el pago bruto de los empleados y considerar los impuestos federales, estatales, del condado y locales, una diversidad de prestaciones, las cuotas sindicales, los programas de inversión, las compras a la compañía, etc., para calcular el pago neto.
3. **Restricciones de tiempo.** Éstas se definen como la cantidad de tiempo permitido o aceptable entre el momento en que los datos están disponibles y el momento en que la información se requiere. Empleando nuevamente la función de la nómina, un contrato sindical puede establecer que ha todos los empleados se les pague completamente a las 9:00 A.M. del martes por el trabajo relacionado hasta la media noche

del viernes anterior. Otro ejemplo es la necesidad de proporcionar el estado del inventario de un producto a un vendedor que está tratando de satisfacer la necesidad urgente de un cliente. Esta persona necesita conocer el estado del inventario con respecto a la última salida o recepción, y no con respecto al último viernes o al final del mes.

4. **Demandas computacionales.** Las demandas computacionales son una combinación única de volumen, complejidad y restricciones de tiempo, para un requerimiento específico de información. Estas demandas computacionales pueden ser considerables si se debe procesar un modelo grande de programación lineal o si se deben dar mantenimiento en línea a una base de datos grande.

3.4.7 Factores organizacionales

Estos factores tienen gran influencia en la forma en que se diseña el sistema de información y la forma en que servirá a la organización. ¿Hasta qué grado estará integrado el sistema con la organización? ¿El sistema de información se extenderá más allá de los límites de la organización para conectarse con ciertos afectados e interesados? ¿El sistema se distribuirá a todos los usuarios, o será más centralizado?.

3.4.8 Requerimientos de costo- eficacia

Cuando uno gasta dinero para llamar a la oficina del estado del tiempo solicitando el último reporte, uno está comprando información. Si uno gasta \$ 800 por un juego de enciclopedias, uno está comprando no solamente libros, sino información. La información y el sistema de información son recursos. Un sistema de información se desarrolla con la idea de mejorar el desempeño gerencial, lograr una diferenciación de productos y servicios e incrementar la productividad para ganar o ahorrar dinero y ser competitivos. Sin embargo, los sistemas de información cuestan dinero y uno no desea gastar un dólar por una información que valga diez centavos de dólar. Por lo tanto, es necesario identificar los costos y beneficios que se van a obtener antes de gastar fondos significativos para desarrollar sistemas de información. La cantidad de dinero disponible para el desarrollo de un sistema de información tendrá, de hecho, un impacto directo y significativo sobre su diseño.

3.4.9 Factores humanos

Existen concepciones diferentes con relación al impacto que tendrán los sistemas de información en la vida del trabajo cotidiano de las personas. El sueño del especialista en sistemas es el de una oficina espaciosa, elegante y moderna,

equipada con los últimos dispositivos en tecnología informática, en donde haya secretarías y gerentes tranquilos y sonrientes que trabajen en armonía y lleven una vida agradable. Los sistemas de información basados en la tecnología realizaron todas las tareas rutinarias, dejando a las personas los trabajos creativos y de reto, auxiliados por una disponibilidad de información exacta, oportuna y relevante al alcance de la mano. La pesadilla del sociólogo es la de una "fábrica en la oficina" en donde trabajadores inexpertos alimenten datos a la computadora y llenen los vacíos que existen entre los procesos automatizados. El trabajo está altamente estructurado y se ha programado previamente, y se emplean tarifas a destajo y monitoreo de máquinas para controlar a los trabajadores. El temor del sociólogo puede ser superado debido a que se pueden diseñar sistemas que sean físicos y psicológicamente cómodos para los empleados, siempre y cuando el diseñador de sistemas considere efectivamente los factores humanos.

El área de los factores humanos está en expansión y en ocasiones es confusa, pero los diseñadores deberán tratar de entenderla y manejarla a fin de diseñar sistemas que trabajen con la gente, y no con ella; sistemas que sean fáciles de utilizar, amables y naturales. Debido a que el diseño del ser humano es fijo el diseño del sistema debe conformarse para adaptarse a él. El objetivo es el diseñar el sistema para que se adapte a los gustos, disgustos, hábitos, destreza y tareas del usuario. Los sistemas que se diseñan teniendo en mente los factores humanos tienen un impacto directo, positivo y esencial en la productividad. Claramente, a medida que se extiende el empleo de la computadora y la tecnología informática,

crece la necesidad de considerar los factores humanos.

Un elemento clave que debe considerarse es el diseño ergonómico de los sistemas de información y el ambiente de trabajo físico. Una de las metas principales de la ergonomía, conocida en ocasiones como ingeniería de los factores humanos, es la de optimizar y hacer compatible la interfaz personas/sistema, que es un determinante crucial de la eficacia del diseño de sistemas. La ergonomía reúne la aplicación sistemática de técnicas de una diversidad de disciplinas, como la fisiología, ingeniería, psicología y sociología, para mejorar el lugar de trabajo para las personas. La mayoría de los ergonomistas en el área de sistemas de información se ha centrado en la persona que trabaja en una terminal de despliegue de vídeo (VDT) o en una estación de trabajo completa. A decir verdad, este es un punto crítico debido a que como han expresado varias gentes: "Ocho horas enfrente de una de esas unidades te van a convertir en un tonto".

De manera específica, los ergonomistas se ocupan de la acústica, el clima de la sala, que la iluminación no sea deslumbrante, la decoración y los colores de las paredes, el descanso para los pies, la libertad en el movimiento de las piernas, la altura del tamaño del asiento, el apoyo para la espalda y la seguridad. Su objetivo, por supuesto, es el de reducir dolores y molestias, excesiva fatiga y tensión en las personas que trabajan en VDTs.

Además de la aplicación de la ergonomía, se pueden hacer los siguientes cosas para mejorar tanto los aspectos físicos como los psicológicos de los

sistemas:

1. **Áreas alternas.** Dividir el lugar de trabajo en dos áreas: área de trabajo y área de descanso. El área de trabajo se diseña ergonómicamente para que las personas puedan concentrarse y realizar su trabajo. El área de descanso, por otra parte, es un lugar de alivio o interrupción diseñada para que se reúnan las gentes en un contacto social, se relajen e intercambien ideas, bromas e historias de guerra. También acomoda a otros que desean meditar o simplemente pasar el rato. Otras comodidades para humanizar el ambiente de trabajo ser podrían centros de acondicionamiento físico o centros de cuidados diarios. Adicionalmente, se pueden instalar sistemas para proporcionar a los empleados posibilidad de compras electrónicas y servicios bancarios, o cualquier otra cosa que les ayude a reducir el factor de "confusión".
2. **Grupos de trabajo.** Aquellas personas que trabajan en departamentos diferentes, como ventas, contabilidad, personal o compras, requieren acceso a la misma información básica (p. ej., cuentas por cobrar, cuentas por pagar, listas de precios, inventarios, registros del personal). El sistema de información deberá diseñarse para unirlos y motivar la apertura, colaboración y compartición del trabajo.
3. **Especificaciones de los usuarios.** Algunos diseñadores de sistemas, al tratar de diseñar un solo sistema para satisfacer a todos los usuarios potenciales, terminan creando un sistema que no satisface a nadie en

particular. Los usuarios tienen diferentes necesidades; en consecuencia, el sistema debe diseñarse para adecuarse a estas necesidades.

4. **Presentación y respuesta.** Como se ha mencionado, las personas trabajan mejor con presentaciones mezcladas. Por lo tanto, siempre que sea posible, deberá emplearse una combinación de formas, textos, gráficas, vídeo y sonido. Para mejorar la presentación, utilice también diversas técnicas de visualización y codificación de entrada, como abreviaturas (p. ej., TX en lugar de Texas), colores para diferentes categorías de información, iconos para indicar categorías o localidades, y códigos variables de parpadeo y brillantez para llamar la atención. Además, un elemento crítico en el desempeño es el tiempo de respuesta. Las demoras grandes en las tareas que necesitan una respuesta rápida pueden conducir a una insatisfacción del usuario y, finalmente, a un desempeño pobre. Adicionalmente, el sistema deberá ser fácil de usar (p. ej., cámara Polaroid o elevador), natural y fácil de entender.
5. **Demandas psicológicas del trabajo.** Las personas deben estar orgullosas de su trabajo. Para quienes aspiran a esto, deben existir oportunidades de un trabajo desafiante y remunerador. (Newhouse, 1987:105-109).

3.5 PARADIGMAS DE LA INGENIERÍA DE SOFTWARE

Los paradigmas de programación son diferentes enfoques para desarrollar sistemas de información. Sin embargo mediante la combinación de métodos completos para todas las fases del desarrollo de *software*, mejores herramientas para optimizar estos métodos, bloques de construcción mas potentes para la implementación del *software*, mejores técnicas para la garantía de calidad del *software* y una filosofía predominante para la coordinación, control y gestión, podemos conseguir una disciplina para el desarrollo de *software* – una disciplina llamada ingeniería de *software*.

La ingeniería de *software* está compuesta por una serie de pasos que abarcan los métodos, herramientas y procedimientos. Estos pasos se denominan frecuentemente paradigmas de la ingeniería de *software* pero también se les conoce como métodos de desarrollo de sistemas de información, es necesario seguir los pasos que marca cada paradigma para poder desarrollar *software* de calidad.

La elección de un paradigma para la ingeniería de *software* se lleva a cabo de acuerdo a la naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y estrategias requeridos. Los siguientes son los tres paradigmas mas usados.

3.5.1 El ciclo de vida clásico

Algunas veces llamado modelo en cascada, el paradigma del ciclo de vida clásico exige un enfoque sistemático y secuencial del desarrollo del *software* que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. Modelizado a partir del ciclo convencional de una ingeniería, el paradigma del ciclo de vida clásico abarca las siguientes actividades.

3.5.1.1 Ingeniería y análisis del sistema.

Debido a que el *software* es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al *software*. Este planteamiento del sistema es esencial cuando el *software* debe interrelacionarse con otros elementos, tales como *hardware*, personas y bases de datos. La ingeniería y el análisis del sistema abarca los requisitos globales del sistema con una pequeña cantidad de análisis y diseño a nivel superior.

3.5.1.2 Análisis de los requisitos del software.

El proceso de recopilación de requisitos se centra e intensifica esencialmente para el *software*. Para poder comprender la naturaleza de los programas que hay que construir, el ingeniero de *software* debe comprender el ámbito de la información del *software*, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del *software*, se documentan y se revisan con el cliente.

3.5.1.3 Diseño.

El diseño del *software* es un proceso multipaso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del *software*, el detalle procedimental y la caracterización de la interfaz. El proceso del diseño traduce los requisitos en una representación del *software* que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del *software*.

3.5.1.4 Codificación.

El diseño se traduce en una forma legible para la máquina. El proceso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

3.5.1.5 Prueba.

Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del *software*, asegurando que todas las secuencias se han probado, y en las funciones externas, realizando prueba que aseguren que la entrada definida produce los resultados que realmente se requieren.

3.5.1.6 Mantenimiento.

El *software*, indudablemente, sufrirá cambios después de que se entregue al cliente (una posible excepción es el *software* empotrado o a la medida). Los cambios ocurrirán debido a que se hayan encontrado errores, a que el *software* se

deba adaptar a un cambio de entorno externo, o debido a que el cliente requiera aplicaciones funcionales o del rendimiento. El mantenimiento del *software* aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

El ciclo de vida clásico es el paradigma más antiguo y más ampliamente usado en la ingeniería del *software*. Sin embargo, con el paso de unos cuantos años, se han producido críticas al paradigma, incluso para seguidores activos, que cuestionan su aplicabilidad a todas las situaciones. Entre los problemas que se presentan algunas veces, cuando se aplica el paradigma del ciclo de vida clásico, se encuentran:

1. Los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
2. Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida lo requiere y tiene dificultades en acomodar posibles incertidumbres que puedan existir al comienzo de muchos proyectos.
3. El cliente debe tener paciencia. Hasta llegar a las etapas finales del desarrollo del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso.

3.5.2 Construcción de prototipos

En este modelo los prototipos que se hacen son generalmente al vapor ya que requieren de fundamento por parte de los usuarios o personas que solicitan la realización de un sistema

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo de *software* a construir. El modelo tomará una de las tres siguientes formas:

1. Un prototipo en papel o un modelo basado en PC que describa la interacción hombre – máquina, de forma que facilite al usuario la comprensión de cómo se producirá tal interacción.
2. Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.
3. Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

Como en todos los métodos de desarrollo de *software*, la construcción de prototipos comienza con la recolección de los requisitos. El técnico y el cliente se

reúnen y definen los objetivos globales para el sistema, identifican todos los requisitos conocidos y perfilan las áreas donde será necesario una mayor definición.

Luego se produce un "diseño rápido". El diseño rápido se enfoca sobre las representaciones de los aspectos del *software* visibles al usuario (por ejemplo, métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente / usuario y se utiliza para refinar los requisitos del *software* a desarrollar. Se produce un proceso interactivo en el que el prototipo es afinado para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer.

Idealmente, el prototipo sirve como un mecanismo para identificar los requisitos del *software*. Si se va a construir un prototipo que funcione, el realizador intenta hacer uso de fragmentos de programas existentes o aplica herramientas (por ejemplo, generadores de informes, gestores de ventanas, etc.) que faciliten la rápida generación de programas que funcionen.

En la construcción de prototipos también existen una serie de problemas para el desarrollo del *software*:

1. El cliente ve funcionando lo que parece ser una primera versión del *software*, ignorando que el prototipo se ha hecho con plastilina y alambres ignorando que, por la prisa en hacer que funcione, no hemos considerado los aspectos de calidad o de mantenimiento de *software* a largo plazo. Cuando se le informa que el producto debe ser reconstruido, el cliente se vuelve loco

y solicita que se le hagan cuantas mejoras sean necesarias para hacer del prototipo un producto final que funcione. El gestor de desarrollo se demora demasiado a menudo.

2. El técnico de desarrollo, frecuentemente, impone ciertos compromisos de implementación con el fin de obtener un prototipo que funcione rápidamente. Puede que utilice un sistema operativo o lenguaje de programación inapropiados, simplemente por que ya está disponible y es conocido, puede que implemente ineficientemente un algoritmo, sencillamente para demostrar su capacidad. Después de algún tiempo, el técnico puede haberse familiarizado con esas elecciones y haber olvidado las razones por las que eran inapropiadas. La elección menos ideal forma parte ahora del sistema.

En la construcción de prototipos para tener éxito la clave está en definir al comienzo las reglas del juego, esto es que el desarrollador y el cliente están de acuerdo que el prototipo se construya para servir sólo como un mecanismo de definición de los requisitos.

3.5.3 El modelo en espiral

El modelo en espiral para la ingeniería del *software*, ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis

de riesgo, que falta en esos paradigmas. El modelo representado es representado mediante una espiral, define cuatro actividades principales:

1. Planificación: determinación de objetivos, alternativas y restricciones.
2. Análisis de riesgo: análisis de alternativas e identificación / resolución de riesgos.
3. Ingeniería: desarrollo de productos del siguiente nivel.
4. Evaluación del cliente: valoración de los resultados de la ingeniería.

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del *software*, cada vez mas completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente. Se pueden usar simulaciones y otros modelos para definir mas el problema y refinar los requisitos.

El cliente evalúa el trabajo de ingeniería y sugiere codificaciones. En base a los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada ciclo alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de seguir o no seguir. Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

Sin embargo, en la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral, y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema, y , al final, al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería, que se puede llevar a cabo bajo el enfoque del ciclo de vida o de la creación de prototipos.

El paradigma del modelo en espiral para la ingeniería de *software* es actualmente el enfoque más realista para el desarrollo de *software* y de sistemas de gran escala. Utiliza un sistema evolutivo para la ingeniería de *software*, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporando dentro del marco de trabajo interactivo que hace un enfoque más realista. El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas la etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos. (Pressman, 1993: 24 a 35).

El modelo en espiral fue utilizado en el desarrollo del sistema HISTOCLIN ver. 10. por que se consideró un método en el cual realmente se obtiene un verdadero avance y realmente se desarrolla lo que el usuario quiere.

CAPÍTULO CUATRO

En este capítulo se encuentra el análisis y diseño del HISTOCLIN versión 1.0 así como toda la documentación de éste. El análisis y el diseño se realizó utilizando el enfoque que tiene Coad y Yourdon, por lo que se utiliza su notación. La técnica de Coad y Yourdon esta orientada para el uso del análisis, diseño y programación orientada a objetos. También se utiliza el paradigma de programación en espiral.

4.1 Análisis y diseño orientado a objetos

Como se menciona anteriormente existen varios tipos de paradigmas de programación pero uno de los que está tomando una mayor fuerza últimamente, por el tipo de aplicaciones que se están generando todo esto motivado por la creación de nuevos lenguajes de programación, que día con día son mas potentes y eficientes, ahorrándole trabajo innecesario a los programadores, este paradigma es el orientado a objetos.

Coad y Yourdon escriben [COA90 ,pág. 1] escriben:

El análisis orientado a objetos se basa en conceptos que una vez aprendimos en la guardería: objetos y atributos, clases y miembros, todo y parte. Nadie sabe por que hemos tratado tanto tiempo en aplicar estos conceptos en el análisis y la especificación de los sistemas de información,

quizás hemos estado demasiado ocupados "siguiendo el flujo" durante nuestros análisis estructurados diarios, para ponernos a considerar otras alternativas.

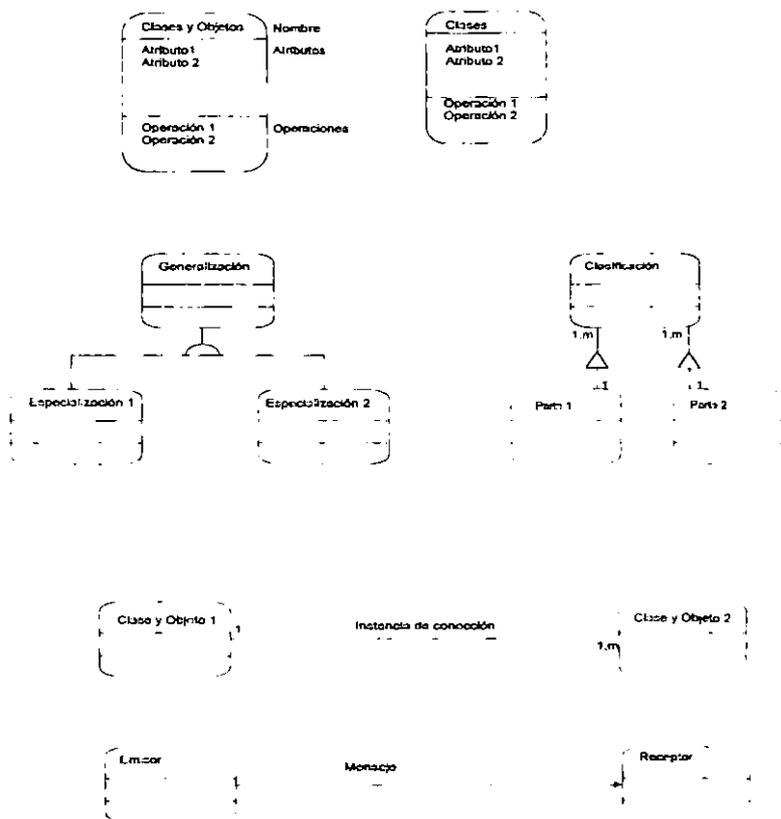
Para comprender el punto de vista de orientado a objetos, consideremos un ejemplo del mundo real una silla. Silla es un miembro (también se usa el término instancia) de una clase de miembros mucho mayor que denominaremos mueble. Se puede asociar un conjunto de atributos genéricos a cada objeto de la clase mueble. Por ejemplo, todo mueble tiene precio, dimensiones, peso, situación y color entre muchos atributos posibles. Se aplican estemos hablando de una mesa o de una silla, de un sofá o de una cómoda, dado que la silla es un miembro de la clase mueble, hereda todos los atributos definidos para la clase.

Cada objeto de la clase mueble puede ser manipulado de diferentes formas, puede ser vendido, modificado físicamente, son operaciones que también son conocidos como servicios o métodos que modifican uno o más atributos del objeto.

La encapsulación significa que todos los atributos, operaciones y otros objetos están empaquetados bajo un sólo nombre y puede ser reutilizada como especificación o como componente de programa. (Pressman, 1993: 2251 a 261).

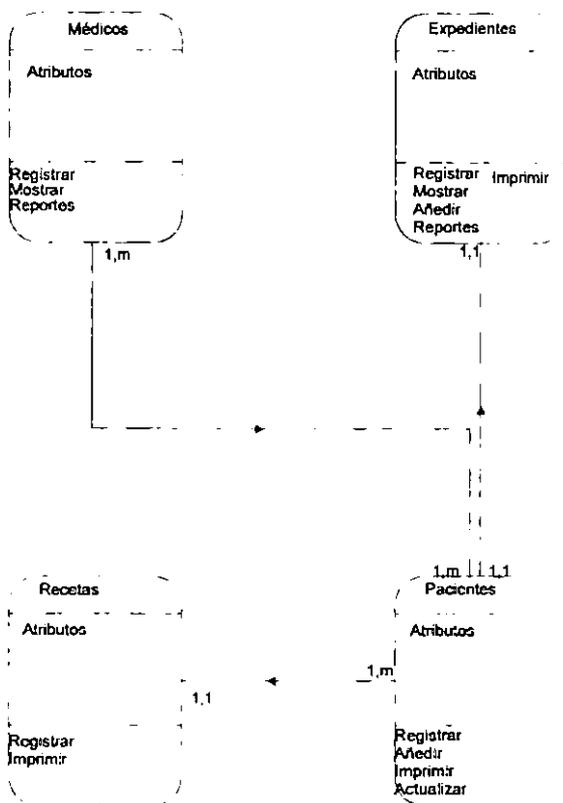
4.2 Notación

La notación o tipo de esquemas que sugieren Peter Coad y Edward Yourdon para la elaboración del análisis orientados a objetos es la siguiente: (Coad y Yourdon, 1990: 196,197)



4.3 Análisis

A continuación se muestra el esquema del análisis orientado a objetos, bajo la metodología de Peter Coad y Edward Yourdon:

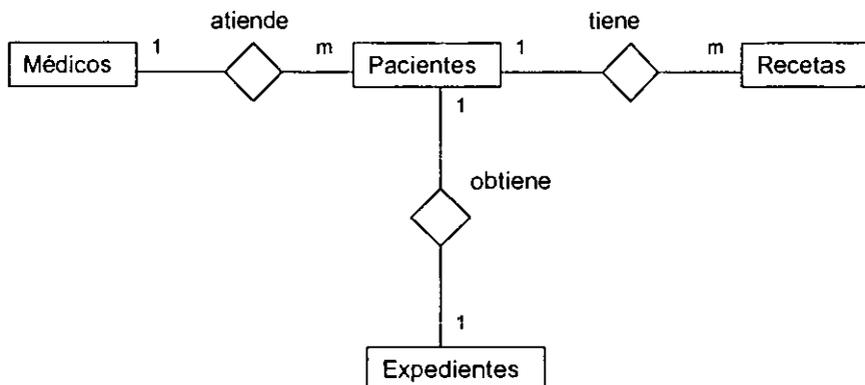


Los atributos de cada objeto son mostrados mas adelante en las tablas de la base de datos.

4.4 Modelado de datos

La modelización de datos se usa ampliamente en aplicaciones para bases de datos, como es el caso de HISTOCLIN ver. 1.0 . Esta técnica proporciona una amplia visión de los datos y de las relaciones que gobiernan los datos

Para HISTOCLIN versión 1.0 el esquema de modelado de datos queda de la siguientes forma:



Expedientes	Pacientes	Recetas	Médicos
<u>Num_registro</u>	Nombre	<u>Num_registro</u>	<u>Num_registro</u>
Interrogatorio	<u>Num_registro</u>	Num_receta	Nombre
Padecimiento_Act	Edad	Fecha	Dirección
Sintomas_gral	Estado Civil	Rx	Telefono
Peso_act	Ocupación		RFC
Peso_ant	Sexo		

Peso_max	Lugar_Naci		
Peso_ideal	Residencia		
Estatura	Dirección		
Pulso	Fecha_vis		
Temperatura	Fecha_act		
Respiración			
Tensión_art			
Inspección_gral			
Cabeza			
Cuello			
Torax			
Abdomen			
Dorso_extre			
Digestivo			
Circulatorio			
Respiratorio			
Urinario			
Genital			
Nervioso			
Endocrino			
Hemato			
Osteo			
Psicoso			
Ex_previos			
Diag_ant			

4.5 DISEÑO

4.5.1 Atributos y características de la Base de Datos

A continuación se dan a conocer todas las propiedades de algunos de los objetos utilizados para HISTOCLIN versión 1.0.

Comenzamos con la base de datos que componen al sistema.

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Base de datos: C:\WINDOWS\ESCRIT~1\Prueba.mdb

Propiedades

AccessVersion: 07.53	AllowBreakIntoCode: Verdadero
AllowBuiltinToolbars: Verdadero	AllowFullMenus:
AllowShortcutMenus: Verdadero	AllowSpecialKeys:
AllowToolbarChanges: Verdadero	AppTitle: Histoclin
Build: 3512	Def. actualizable: Verdadero
DefaultPartner: {guid {00000000-0000-0000-0000-0000}	DesignMasterID: {guid {B94B8A43-F9FE-11D1-BCFE-89B}
LastUpdater: {guid {B94B8A43-F9FE-11D1-BCFE-89B}	MostRecentSyncPartner {guid {B94B8A6A-F9FE-11D1-BCFE-89B}
Orden de combinación: General	Registros afectados: 0
ReorderTables: T	Replicable: T
ReplicableBool: Verdadero	ReplicaID: {guid {B94B8A43-F9FE-11D1-BCFE-89B}
RepVersion: 0	StartupForm: Panel de control
StartupShowDBWindo Verdadero	StartupShowStatusBar:
Tiempo de espera de la Versión : 3.0	60 Transactions: Verdadero

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Los siguiente son las relaciones de las tablas que componen HISTOCLIN versión 1.0., que como se menciona anteriormente, Microsoft Access hace uso del modelo relacional.

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Relaciones: Todo

Relaciones

PacientesExpedientes

Pacientes	Expedientes
Num_registro	Num_registro
:	Uno a uno
Atributos:	Único, No exigir

PacientesMédicos

Pacientes	Médicos
Num_registro	Num_registro
Atributos:	No exigir
Atributos:	Uno a varios

PacientesRecetas

Pacientes	Recetas
Num_registro	Num_receta
Atributos:	No exigir
Atributos:	Uno a varios

4.5.2 Diseño y características de las tablas

En la utilidad que tiene el Microsoft Access el documentador arroja las siguientes características y propiedades de cada tabla.

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Elementos del Panel de control

Propiedades

ActivarFiltro: Falso

Def. actualizable:

31/05/98 11:35:00 PM

RecordCount: 17

ReplicableBool:

5/06/98 3:43:34 PM

CollsGuid: s_GUID

Verdadero Fecha de creación:

Replicable: F

Falso Última actualización:

Columnas

Nombre	Tipo	Tamaño
s_GUID	Id de réplica	16
SwitchboardID	Número (largo)	4
ItemNumber	Número (entero)	2
ItemText	Texto	255
Command	Número (entero)	2
Argument	Texto	50

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Expedientes

Propiedades

ActivarOrden: Falso

Def. actualizable:

1/06/98 6:20:12 PM

RecordCount: 0

ReplicableBool:

5/06/98 3:43:49 PM

CollsGuid: s_GUID

Verdadero Fecha de creación:

Replicable: F

Falso Última actualización:

Columnas

Nombre	Tipo	Tamaño
s_GUID	Id de réplica	16
Num_registro	Número (largo)	4
Interrogatorio	Memo	-
Padecimiento_Act	Memo	-
Sintomas_gral	Texto	100
Peso_act	Texto	3
Peso_ant	Texto	3
Peso_max	Texto	3
Peso_ideal	Texto	3
Estatura	Número (largo)	4
Pulso	Texto	10
Temperatura	Texto	10
Respiración	Texto	50
Tensión_art	Texto	10
Inspección_gral	Texto	50
Cabeza	Texto	200
Cuello	Texto	50
Torax	Texto	100
Abdomen	Texto	130
Dorso_extre	Texto	100
Digestivo	Texto	70
Circulatorio	Texto	50
Respiratorio	Texto	50
Urinario	Texto	50
Genital	Texto	50
Nervioso	Texto	50
Endocrino	Texto	50
Hemato	Texto	50
Osteo	Texto	50
Psicoso	Texto	50
Ex_previos	Texto	50
Diag_ant	Texto	50

Relaciones

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Expedientes

PacientesExpedientes

Pacientes	Expedientes
Num_registro	Num_registro
Atributos:	Único, No exigir
RelationshipType:	Uno a uno

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Médicos

Propiedades

ActivarOrden: Falso

Def. actualizable:

1/06/98 6:11:30 PM

RecordCount: 2

ReplicableBool:

5/06/98 3:43:44 PM

CollsGuid: s_GUID

Verdadero Fecha de creación:

Replicable: F

Falso Última actualización:

Columnas

Nombre	Tipo	Tamaño
s_GUID	Id de réplica	16
Num_registro	Número (largo)	4
Nombre	Texto	50
Dirección	Texto	50
Telefono	Texto	10
RFC	Texto	14

Relaciones

PacientesMédicos

Pacientes	Médicos
Num_registro	Num_registro
Atributos:	No exigir

RelationshipType: Uno a varios

C:\WINDOWS\ESCRIT~1\Prueba.mdb
Tabla: Pacientes

Propiedades

ActivarOrden: Falso
Def. actualizable:
1/06/98 6:16:17 PM
RecordCount: 2
ReplicableBool:
5/06/98 3:43:54 PM

CollsGuid: s_GUID
Verdadero Fecha de creación:
Replicable: F
Falso Última actualización:

Columnas

Nombre	Tipo	Tamaño
s_GUID	Id de réplica	16
Nombre	Texto	50
Num_registro	Número (largo)	4
Edad	Texto	3
Estado Civil	Texto	20
Ocupación	Texto	50
Sexo	Texto	15
Lugar_Naci	Texto	30
Residencia	Texto	30
Dirección	Texto	50
Fecha_vis	Fecha/Hora	8
Fecha_act	Fecha/Hora	8

Relaciones

PacientesExpedientes

Pacientes	Expedientes
Num_registro	Num_registro

Atributos: Único, No exigir
RelationshipType: Uno a uno

PacientesMédicos

Pacientes	Médicos
Num_registro	Num_registro
Atributos:	No exigir
Atributos:	Uno a varios

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Pacientes

PacientesRecetas

Pacientes	Recetas
Num_registro	Num_receta
Atributos:	No exigir
Atributos:	Uno a varios

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Tabla: Recetas

Propiedades

ActivarOrden: Falso
 Def. actualizable:
 1/06/98 6:30:17 PM
 RecordCount: 1
 ReplicableBool:
 8/06/98 4:53:47 PM

CollsGuid: s_GUID
 Verdadero Fecha de creación:
 Replicable: F
 Falso Última actualización:

Columnas

Nombre	Tipo	Tamaño
s_GUID	Id de réplica	16
Num_registro	Número (largo)	4
Num_receta	Número (largo)	4
Fecha	Fecha/Hora	8
Rx	Memo	-

Relaciones

PacientesRecetas

Pacientes	Recetas
Num_registro	Num_receta
Atributos:	No exigir
RelationshipType:	Uno a varios

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Base de datos: C:\WINDOWS\ESCRIT~1\Prueba.mdb

Propiedades

AccessVersion: 07.53	AllowBreakIntoCode: Verdadero
AllowBuiltInToolbars: Verdadero	AllowFullMenus:
AllowShortcutMenus: Verdadero	AllowSpecialKeys:
AllowToolbarChanges: Build: 3512	Verdadero AppTitle: Histoclin
DefaultPartner: {guid	Def. actualizable: Verdadero
{00000000-0000-0000-0000-0000	DesignMasterID: {guid
BCFE-89B	{B94B8A43-F9FE-11D1-
LastUpdater: {guid	MostRecentSyncPartner {guid
{B94B8A43-F9FE-11D1-BCFE-89B	{B94B8A6A-F9FE-11D1-
BCFE-89	
Orden de combinación:	General Registros afectados: 0
ReorderTables: T	Replicable: T
ReplicableBool: Verdadero	Verdadero ReplicaID: {guid
BCFE-89B	{B94B8A43-F9FE-11D1-
RepVersion: 0	StartUpForm: Panel de control
StartUpShowDBWindo Verdadero	Verdadero StartUpShowStatusBar:
Tiempo de espera de la Versión : 3.0	60 Transactions: Verdadero

Permisos de usuario

admin

Permisos de grupo

Admins

Users

C:\WINDOWS\ESCRIT~1\Prueba.mdb

Relaciones: Todo

Relaciones

PacientesExpedientes

Pacientes

Num_registro

Expedientes

Num_registro

:

Atributos:

Uno a uno

Único, No exigir

PacientesMédicos

Pacientes

Num_registro

Médicos

Num_registro

Atributos:

Atributos:

No exigir

Uno a varios

PacientesRecetas

Pacientes

Num_registro

Recetas

Num_receta

Atributos:

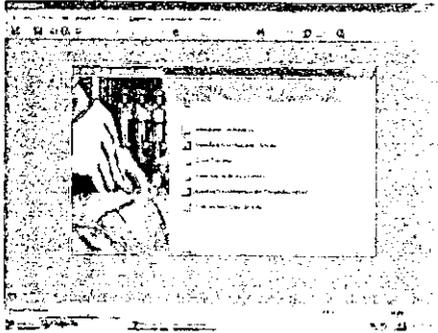
Atributos:

No exigir

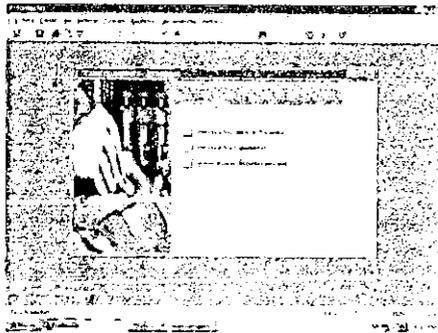
Uno a varios

4.5.3 Diagrama de estados del programa

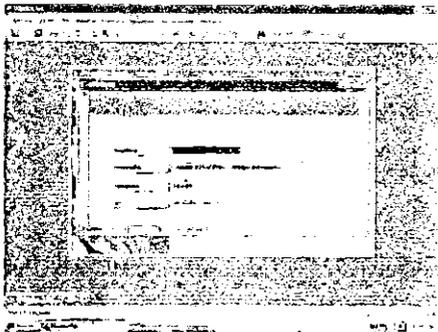
Inicio



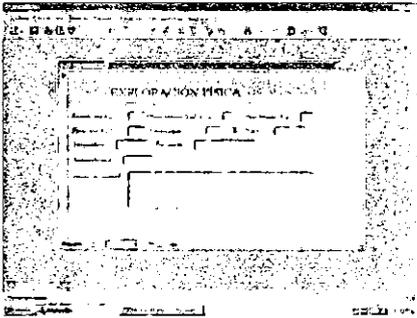
Menú Principal



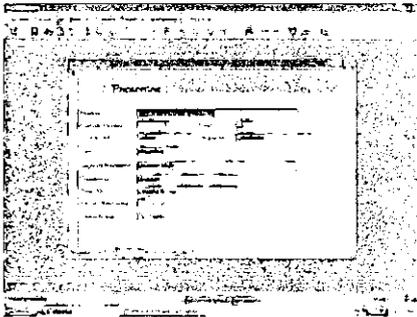
Submenú
Expedientes



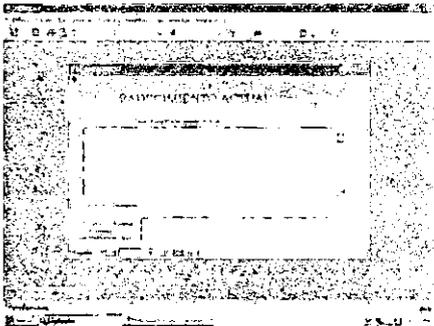
Módulo de Médicos



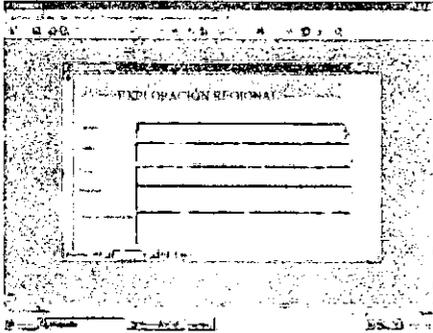
Módulo de Pacientes



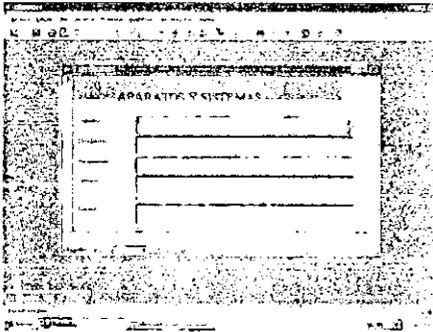
Módulo de Expedientes



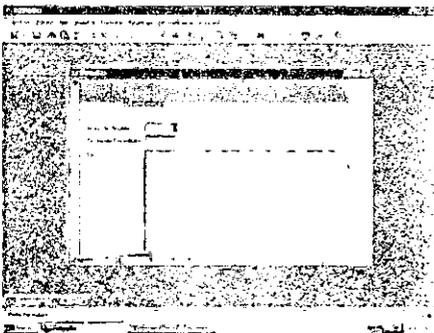
Módulo de Expedientes



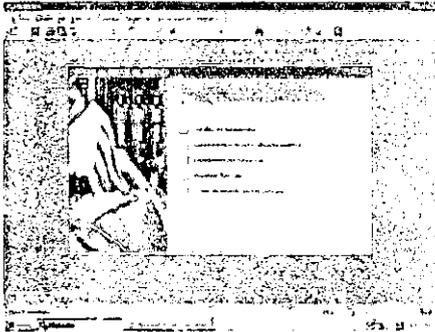
Módulo de
Expedientes



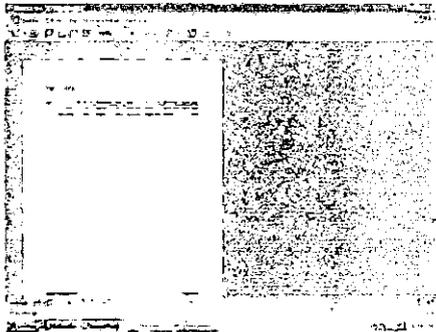
Módulo de
Expedientes



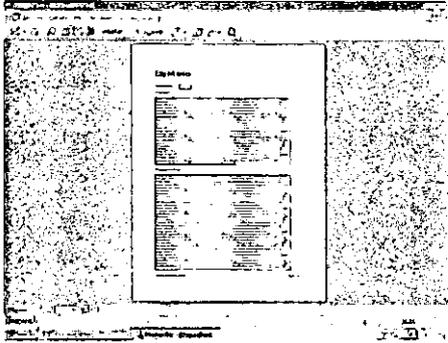
Módulo de Recetas



Submenú de Reportes



Reportes



Ejemplo de reporte de expedientes

4.5.4 Pruebas al sistema

En el capítulo tres se mencionan cuales son las fases que se componen para el desarrollo de un sistema, dependiendo del tipo de paradigma que se va a usar aunque prácticamente en todos se tiene la fase de pruebas y mantenimiento al sistema.

Mucha gente cree que un sistema estará listo y sin errores en terminando el ciclo del sistema, pero el *software* siempre va a contar con errores, por eso que esta fase es esencial para los sistemas. En ocasiones no se podrán eliminar todos los errores porque hay algunos que se dan en la mitad del ciclo de vida del sistema u otros que sean sin importancia.

Existen básicamente dos técnicas para probar un sistema:

Caja blanca

La prueba de caja blanca se usa para garantizar el óptimo diseño procedimental del sistema, como en los siguientes casos:

1. Garantizar que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo.
2. Verificar todas las decisiones lógicas en sus fases de verdadero y falso

3. Ejecutar todos los ciclos y que se cumplan con veracidad.

Caja negra

Esta prueba es el complemento de la prueba de caja blanca y se concentra en los requisitos funcionales del sistema. Los errores que intenta encontrar la prueba de caja negra son los siguientes:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores de estructuras de datos o en accesos a bases de datos externas.
4. Errores de inicialización y de terminación.

Por las características del sistema las pruebas que se realizaron son de caja negra, ya que se hizo el sistema HISTOCLIN versión 1.0 con el menor código de programación posible y explotando las características de Microsoft Access.

CONCLUSIONES

En el desarrollo de sistemas la ejecución y terminación de cada una de las etapas del método que se eligió para la elaboración del programa es fundamental para la culminación de éste. Desde la recopilación de la información necesaria que envuelven al problema hasta la etapa de pruebas y mantenimientos del sistema.

El hacer una aplicación para ambiente gráfico trajo consigo grandes ventajas para los usuarios novatos e inexpertos en el uso de equipos de cómputo, ya que por ejemplo en Microsoft Windows 95 la interfaz gráfica es mas amigable y lo que trata de simular en pantalla es un escritorio común y corriente por que se usan carpetas para guardar la información, lo que hacemos de manera cotidiana al guardar documentos en un archivero. Inclusive se cuenta con un bote de basura y por si fuera poco también con un maletín electrónico para llevar consigo toda la información que desee.

Hablando del Microsoft Access 97 encontramos que cuenta con una serie de ventajas como son las siguientes:

- ✓ La ayuda, es realmente un elemento que cumple con su función, es bastante amplia y cuenta con muchos expertos que ahorran mucho tiempo para dar respuesta a las dudas que puedan surgir.

- ✓ Es actualmente la única base de datos que usa el portafolio de Microsoft Windows 95 para la depuración de datos entrada por entrada. Esto le permite a los usuarios que viajan, actualizar sólo las partes de sus bases de datos que hayan cambiado mientras ellos estaban viajando, en vez de copiar toda la base de datos.
- ✓ Microsoft Access 97 cuenta con una excelente vinculación con los demás componentes del Microsoft Office 97, es decir, puede integrarse una tabla de Access 97 a un documento de Word 97 , y hacerle cambios directamente en el procesador de palabras. También se puede trabajar con hojas de calculo de Excel 97 desde dentro de Access 97.
- ✓ Cuenta con un lenguaje de programación, el cuál es Visual Basic para Access, con lo que se pueden desarrollar programas casi de cualquier necesidad.

En el sistema operativo para el que esta hecho Microsoft Windows 95 es el mas utilizado en todo el mundo y en seguida se mencionan algunas razones de su éxito:

- ✓ Nueva interfaz de trabajo *drag and drop*, en donde los elementos del escritorio ofrecen una gran versatilidad.
- ✓ Soporte de la arquitectura *Plug & Play*, el cual facilita la instalación de nuevos dispositivos para la computadoras.

- ✓ Reconfiguración dinámica de componentes, sin necesidad de emplear complicados procesos de instalación.
- ✓ Sistema operativo de 32 bits, con mejor manejo de memoria y capacidad de correr varias aplicación a la vez, es decir, le da prioridad a la multitarea.
- ✓ La memoria se maneja en forma separada para cada aplicación, con lo que se evitan un poco mas las caídas o congelamientos de la computadora.
- ✓ Nuevo sistema de archivos que ofrece un mayo velocidad y permite nombres de archivos mas largos.
- ✓ Servicios de red y comunicaciones mejorados, permitiendo compartir recursos más fácilmente.

Al haber mencionado algunas de las ventajas y características que tiene el sistema operativo y Microsoft Access 97. HISTOCLIN versión 1.0 se hace acreedor de estas ventajas nativas de sistema operativo y del administrador de bases de datos.

HISTOCLIN versión 1.0 ayuda a los médicos a llevar un mejor registro de sus historias clínicas de los pacientes, así se evitaran el enorme papeleo que les

ocasiona el redactar cada una de las historias clínica y recetas necesarias para cada paciente.

El uso de un menú para la elección de la alternativa deseada facilita el uso del programa, aunque también se puede hacer uso de manera independiente al ejecutar el formulario, consulta o informe deseado.

Al tener una base de datos de pacientes y con la vinculación que se tiene con Microsoft Word 97 fácilmente se pueden realizar, por ejemplo invitaciones, circulares, etc. para pacientes haciendo uso de los campos que tiene la tabla de pacientes, así evitar el molesto llenado de invitaciones de manera individual.

En HISTOCLIN versión 1.0 se utilizó el menor código de programación posible, sin embargo, se hizo mayor uso de asistentes y el generador de eventos, haciendo el desarrollo del programa de una manera mas rápida pero trayendo consigo una desventaja enorme, el tamaño del programa es muy grande.

El utilizar Microsoft Access 97 al principio de la tesis se veía como una tarea sencilla, pero para hacer uso de este administrador de base de datos no es tan fácil, por que es demasiado grande el programa que muchas veces te pierdes en su complejidad.

A manera de resumen las ventajas que se tiene el utilizar HISTOCLIN versión 1.0 son las siguientes:

- ✓ Contar con información ordenada de los pacientes.
- ✓ Realizar reportes de una manera fácil y rápida.
- ✓ Realizar consultas en específico y rápida.
- ✓ Tener estadísticas de sus pacientes.
- ✓ Uso sencillo del sistema.
- ✓ Interfaz amigable e intuitiva.
- ✓ Interrogar correctamente y mas afondo a los pacientes, por medio de los esquemas o pantalla que tienen información util para el interrogatorio y captación de la información.
- ✓ Establecer un historial por paciente.

También es importante resaltar las desventajas que tiene HISTOCLIN versión 1.0:

- ✓ No es un programa ejecutable, es decir, necesita de otro sistema (Microsoft Access 97) para su funcionamiento.
- ✓ Enlazarlo con programas que manejen un sistema hospitalario, para poder ofrecer un solución integral.

- ✓ HISTOCLIN versión 1.0 ocupa mas espacio en disco que otro programas de su tipo (versión MS-DOS).
- ✓ Guardar una fotografia digitalizada de los pacientes, para hacer búsquedas por foto y así tener una mayor referencia del paciente.

BIBLIOGRAFIA

1. Barker F. Scoot con Barker Diana

Microsoft Access 97 Paso a paso

Prentice-Hall Hispanoamericana

2. Burch y grudnidski

Análisis y diseño de sistemas de información

Primera edición 1996

Ed. Limusa

3. Coad Peter and Edward Yourdon

Object-Oriented Analysis

Second Edition

U. S. A. 1990

Yourdon Press

4. Enciclopedia Microsoft® Encarta® 98 ©

1993-1997 Microsoft Corporation. Reservados todos los derechos.

5. Hansen Gary W. & Hansen James V.

Diseño y administración de bases de datos

España, 1997

Segunda edición

Ed. Prentice Hall

6. Kendall & Kendall

Análisis y diseño de sistema

México 1997

Tercera edición

Ed. Prentice Hall

7 M. en C. Morales López Felipe

Bases de datos

Instituto Tecnológico de Morelia

Enero 1993

8 Pressman Roger S.

Ingeniería de Software "Un enfoque práctico"

España 1993

Tercera edición

Mac Graw Hill