

47
21



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**AUTOMATIZACION DE LOS TRAMITES PARA
LA PRESTACION DE SERVICIOS DEL CENTRO
DE COMPUTO DE LA UNIVERSIDAD
AUTONOMA METROPOLITANA UNIDAD
XOCHIMILCO**

T E S I S

**Que para obtener el titulo de
INGENIERO EN COMPUTACION**

p r e s e n t a n

**GATICA OLVERA RAUL BALDEMAR
PARRA NAVARRETE ADRIAN ELEAZAR
RIVERA HUERTA RAUL**



Director de Tesis: M. en I. JUAN CARLOS ROA BEIZA

México, D. F.

1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mis padres y hermanos, por todo el apoyo que me han dado a lo largo de mi carrera.

A la Universidad, por haberme dado los conocimientos suficientes para llegar hasta este punto.

A todas aquellas personas con quienes compartí buenos y malos momentos a través de este largo camino.

Raúl Baldemar Gatica Olvera.

Este trabajo de tesis lo dedico a:

Dios, por permitirme existir y hacer realidad muchos sueños.

"Acuérdate de tu creador en los días de tu juventud antes que vengan los días malos y lleguen los años de los cuales digas no tengo en ellos contentamiento".

A mis padres por regalarme todos esos momentos de felicidad, comprensión y apoyo sin esperar nada a cambio. Por haber sabido dirigirme con sabiduría y ser una luz firme en todos los caminos de penumbra y altibajos de la vida. Y sobre todo por el amor que siempre me han demostrado.

A Dalía, Mario, Diana y Bebe por estar siempre unidos y hacer más emotivo cada momento que pasamos juntos.

A ti Maribel, porque siempre me has impulsado y enseñado otras formas de sentir la vida.

A ti por todo el cariño y paciencia que me tienes. Te amo.

A la Universidad, profesores y compañeros que con gran esfuerzo y esmero compartieron conmigo sus conocimientos; cuyos frutos abren las puertas del éxito.

Adrián Eleazar Parra Navarrete.

A mis padres y hermanos:

Esto es el resultado del amor que me han brindado durante todos estos años de mi vida y por lo cual los quiero hacer parte de ello; ya que sin su apoyo no hubiera podido lograr terminar mis estudios profesionales. A todos y cada uno de ustedes les debo mucho.

Muchas gracias. ¡¡ Los quiero !!

Raúl Rivera Huerta



ÍNDICE



ÍNDICE TEMÁTICO.

INTRODUCCIÓN

CAPITULO I. FUNDAMENTOS TEÓRICOS.

I.1. Sistemas automáticos de identificación	1
I.1.1. Reconocimiento óptico de caracteres	1
I.1.2. Código de barras	8
I.1.2.1. Estándares	12
I.1.2.2. Tipos	14
I.1.2.3. Procedimientos para su implementación	31
I.1.2.4. Beneficios que ofrece el sistema	32
I.2. Redes de computadoras	33
I.2.1. Conceptos generales	34
I.2.1.1. Estándares y topologías	43
I.2.1.2. Protocolos	57
I.2.1.3. Interconexiones	60
I.3. Ingeniería de software	65
I.3.1. Definición de requisitos	71
I.3.2. Especificación del software	78
I.3.2.1. Especificaciones de interfaces	79
I.3.2.1. Especificaciones operacionales	80
I.3.2.3. La especificación de abstracciones de datos.	81
I.3.2.4. Especificaciones prácticas de software	82
I.3.3. La interfaz del usuario.	85
I.3.3.1. Diseño de interfaces del usuario	86
I.3.3.2. Interfaces iniciadas por la computadora	87
I.3.3.3. El uso de gráficas.	88
I.4. Bases de datos.	89
I.4.1. Conceptos generales	89
I.4.1.1. Ventajas	95
I.4.1.2. Porque utilizar Base de Datos	96
I.4.1.3. Desventajas	98
I.4.1.4. Entidades y atributos	99
I.4.1.5. Arquitectura de un sistema de B.D.	101
I.4.2. Bases de Datos Distribuidas	109

TESIS SIS-UAM

1.4.2.1. Estructura de las B.D. distribuidas	112
1.4.3. Bases de Datos relacionales.	119
1.4.4. Normalización	123
1.4.4.1. Objetivos	126
1.4.5. Diseño e implementación de una B.D.	127

CAPITULO II. HERRAMIENTAS DE DISEÑO

II.1. Servidores de red	130
II.1.1. Servidor HP9000 serie 827	134
	143
II.2. Sistema operativo UNIX versión 9.04	160
II.3. Red Ethernet	161
II.3.1. Arquitectura	179
II.3.2. Protocolo TCP/IP	184
II.3.3. Interconexiones	189
II.3.4. Administración	194
II.3.5. Pathway	197
II.4. Manejador de Base de Datos ORACLE	197
II.4.1. Definición	197
II.4.1.1. Características de ORACLE	203
II.4.1.2. Componentes de ORACLE	214
II.4.2. Estructuras de la Base de Datos	216
II.4.2.1. Estructura Lógica de la Base de Datos.	225
II.4.2.2. Estructura Física de una Base de Datos	227
II.5. Interfaces	227
II.5.1. Bar code and Magnetic Stripe Reader	227

CAPITULO III. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

III.1. Antecedentes y Planteamiento del Problema	233
III.2. Requerimientos del Usuario	235
III.3. Recopilación de Información	236

III.4. Análisis de la información obtenida	244
III.5. Propuesta de solución	247
CAPITULO IV. DISEÑO E IMPLEMENTACIÓN	
IV.1. Notaciones de diseño	249
IV.2. Diagramas de flujo de datos	251
IV.3. Modelos entidad/relación	259
IV.4. Diccionario de datos	267
IV.5. Metodología de la programación	293
IV.6. Implementación, Pruebas y Depuración de Programas	300
IV.6.1. Mantenimiento y Depuración del Sistema.	302

CONCLUSIONES	304
MANUAL DE USUARIO	
MANUAL TÉCNICO	
GLOSARIO	
BIBLIOGRAFÍA	



INTRODUCCIÓN

INTRODUCCIÓN

La **UNIVERSIDAD AUTÓNOMA METROPOLITANA**, plantel **XOCHIMILCO**, es un organismo del estado con personalidad jurídica y patrimonio propio, creada por decreto del Congreso de la Unión, con 21 años de experiencia.

Está encargada de preparar profesionalmente a la población estudiantil que cuenta con los requisitos fundamentales para desarrollar una carrera dentro de esta sociedad, que requiere día a día de personal altamente preparado que se encuentre en un nivel competitivo, suficiente para salir adelante, así como organizar y desarrollar actividades de investigación humanística y científica para perseverar y difundir la cultura.

Actualmente cuenta con 17 carreras, divididas en tres áreas:

- Ciencias Sociales y Humanidades
- Ciencias Biológicas y de la Salud y
- Ciencias y Artes para el Diseño,

Las cuales requieren del servicio propio de un Centro de Cómputo.

El avance tecnológico ha permitido que la computadora pueda ser utilizada en una gran cantidad de organizaciones concentrando la función informática en **CENTROS DE CÓMPUTO**, quienes alojan equipo y programas que procesan la información además de contar con personal especializado, de cuya capacidad dependerá la satisfacción de las necesidades informáticas que existan en la organización.

Actualmente, los trámites para la prestación de algún servicio en el **CENTRO DE COMPUTO UAM- XOCHIMILCO**, es realizado manualmente o semiautomatizado, lo que no cumple con las características de este tipo de lugares, esto, trae como consecuencia la pérdida de tiempo tanto para los usuarios, como para los prestadores de servicio.

El automatizar es una etapa de desarrollo de constante cambio y evolución, bien sea para la tecnología informática, o bien para la administración del centro de cómputo, ya que si se llegara a adoptar una actitud pasiva o a mantener una estabilidad, se corre el riesgo de que el centro de cómputo y todas sus actividades se vuelvan insuficientes y obsoletas, por lo cual se crea la necesidad de **AUTOMATIZAR** los trámites de prestación de servicios del centro de cómputo de este plantel.

Teniendo una buena base administrativa se asegura un óptimo y oportuno funcionamiento integral de toda la institución; el **Manejador de Bases de Datos ORACLE**, es un instrumento ideal

para el tipo de aplicación que se presentará al término de este trabajo de TESIS, ya que su capacidad de almacenamiento a grandes cantidades de información a la que el usuario tiene acceso, y sus características como la estandarización y el mejor control de información, lograrán que los objetivos de este trabajo se cumplan; para esto hay que estudiar los recursos con los que se cuentan que podrían facilitar el trabajo; se sabe de antemano que las credenciales tanto de alumnos como de profesores, están registradas bajo un código de barras que accesa a la B.D. que agilizará la automatización pues evitará que se invierta tiempo en la inserción de datos vía teclado y agilice el proceso. (Esto se abarca en el capítulo I de la tesis).

Es conveniente plantear las necesidades como problemas a resolver en forma sistemática, por tal motivo tanto el OBJETIVO GENERAL como los OBJETIVOS ESPECÍFICOS, son plasmados a continuación, ya que estos fueron los puntos básicos que se tomaron para que al capítulo tras capítulo se llegara satisfactoriamente a la solución de el problema actual.

OBJETIVO GENERAL.

Diseñar un sistema confiable para administrar el Centro de Cómputo de la UAM - XOCHIMILCO, utilizando el código de barras como sistema de adquisición de datos que permita la agilización de los trámites para la prestación de servicios que proporciona dicho Centro de Cómputo.

OBJETIVOS ESPECÍFICOS.

- 1.- El sistema deberá tener como dispositivos de entrada de datos el lector de código de barras y teclado y como dispositivos de salida la pantalla así como la impresora.
- 2.- El sistema debe llevar el control de reservaciones y reasignaciones de equipo y salas de cursos por día y hora, lleva el control de ordenes servicios de reparación y mantenimiento de equipo de cómputo, permite el manejo de inventario de equipo y usuarios del centro de cómputo.
- 3.- El sistema generará estadísticas acerca de demandas de servicios de cómputo en fechas, horarios y equipos; para ayudar en la toma de decisiones en la administración del centro de cómputo.

Todo esto, procura poner en el mismo nivel la demanda que existe por parte de los usuarios y el servicio que se les otorga, ahorrando así horas hombre, que pueden invertirse en formar profesionistas mejor capacitados.



CAPITULO I



FUNDAMENTOS TEÓRICOS

CAPITULO I. FUNDAMENTOS TEÓRICOS.

I.1. SISTEMAS AUTOMÁTICOS DE IDENTIFICACIÓN.

Los sistemas de identificación se aplican actualmente para las personas y objetos bajo la forma de registros magnéticos, ópticos, sonoros e impresos.

En general se trata de un elemento codificado portador de la información y un elemento lector capaz de reconocer información; ésta alimenta por lo regular a una computadora donde la identificación es decodificada, verificada, comparada y aceptada para luego tomar una decisión lógica.

Los sistemas modernos son automáticos, lo que agiliza su proceso, evita errores y aumenta su confiabilidad y eficiencia. Estos mismos sistemas se utilizan también para la identificación de objetos especialmente cuando están destinados a una actividad comercial.

I.1.1. RECONOCIMIENTO ÓPTICO DE CARACTERES.

Se trata también de caracteres impresos cuya forma es la información que se desea procesar; son leídos automáticamente por un haz de luz y decodificados por algoritmos matemáticos a una forma digital, analógica o ASCII.

La lectura es por contacto o a distancia, el haz es fijo o móvil, y visible o no (infrarrojo), la fuente de luz puede ser policromática (incandescente) o coherente: láser, de estado sólido (diodos fotoemisores) o gaseoso (helio-neón). Los sistemas están siendo desplazados por el Código de barras, para su uso comercial masivo.

La asignación y aplicación, de un número de código a cada producto (por ejemplo en el supermercado) no es un sistema automático de identificación ya que tanto la marcación como la lectura de cada producto es manual y el sistema de reconocimiento óptico de caracteres en cuestión, se refiere solamente a los sistemas que leen automáticamente la información.

SCANNERS.

El reconocimiento óptico de caracteres (OCR) se realiza por lo general con la ayuda de dispositivos electrónicos llamados **scanners**. Existen dos tipos principales de scanners: los manuales y los de página (o escritorio).

CAPITULO I

Los manuales tienen un ancho de 2 y 5 pulgadas, así que se requiere hacer varias pasadas para leer un documento completo, luego se usa un software especial para unir múltiples pasadas en un sólo y uniforme documento. A veces los modelos manuales se utilizan para OCR, pero la inevitable vibración de la mano humana hace difícil realizar la casi perfecta lectura requerida para OCR, los scanners manuales son más prácticos para capturar figuras pequeñas que puedan ser leídas en una sola lectura.

Los scanners de página o de escritorio pueden manejar una página completa a la vez, estos se dividen a su vez en scanners de: alimentación, flatbed y overhead.

En los scanners de alimentación se pone la página a ser capturada y existen rodillos que la introducen a la máquina, los rodillos son la única parte móvil de este tipo de scanner, el sensor óptico y la circuitería se quedan fijos mientras la página fluye. Tienen la desventaja de que no pueden capturar páginas de libros o material engrapado.

Los scanners flatbed son mecánicamente los más complejos y usualmente los más caros, se operan de la misma forma que una fotocopidora: el documento a ser leído se coloca en una superficie de vidrio, abajo de este existe una cabeza óptica que se mueve a lo largo del documento para capturar la imagen, este diseño puede manejar fácilmente libros y otro material pegado o engrapado.

Este tipo de scanner tiene ventajas importantes, se les prefiere por el alineamiento preciso de dibujos detallados y se pueden copiar imágenes de libros y revistas sin tener que destruirlas.

Los scanners overhead son parecidos a proyectores, en estas unidades el documento se pone en una superficie plana y se captura desde una cabeza que está posicionada arriba. Estos scanners generalmente tienen una profundidad nominal de 3/4 de pulgada, lo cual significa que pueden leer ciertos objetos tridimensionales de bajo perfil.

RESOLUCIONES Y ESCALAS.

La resolución de un scanner es el número de puntos que pueden muestrear por cada pulgada lineal del documento, resolución típicas son 75, 100, 150, 200, 300, 400 y 600 dpi (dots per inch), pero debido a que la mayoría de las impresoras láser actuales están limitadas a 600 dpi, esta resolución es la más alta que la mayoría de los scanners proveen.

Resoluciones más altas resultan en archivos de disco más grandes para las imágenes capturadas debido a que se almacenan más bits de información, muchos scanners ofrecen ahora

resolución seleccionable permitiendo al usuario cualquier resolución estándar hasta el límite del scanner, esto permite crear archivos más pequeños para imágenes simples capturadas en baja resolución.

Los scanners más simples son dispositivos de dos niveles llamados de dos tonos. Estas unidades son usadas principalmente para dibujos simples que puedan ser representados simplemente en blanco y negro, intentar capturar fotografías y otras ilustraciones de medio tono resulta en la pérdida de la mayoría de los detalles.

Con los scanners de gray scale es posible capturar fotografías con una buena calidad. Casi todos los scanners de gray-scale son de las siguientes tres categorías: 16, 64 y 256 niveles, conocidos como scanners de 4, 6 y 8 bits respectivamente, correspondiendo al número de bits requerido para almacenar cada pixel de información. Estos scanners dividen el color entre 16, 64 o 256 tonos discretos de gris, obviamente, mientras más tonos soporte el scanner, más precisa será la imagen digitalizada.

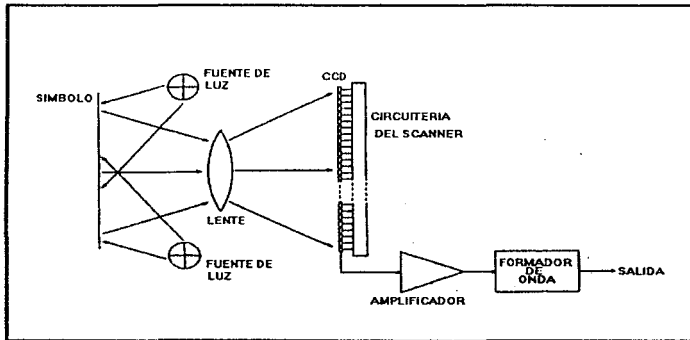


Figura 11.2.3 Diagrama de bloques de un scanner tipo CCD.

ANATOMÍA DE UN SCANNER.

La sig. figura muestra un scanner manual visto desde arriba, con la tapa superior retirada.

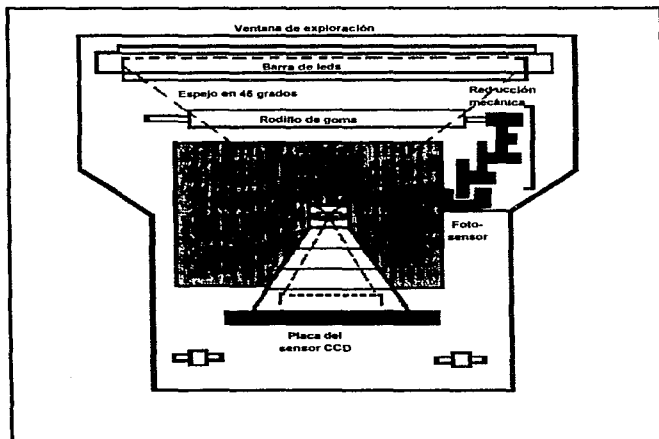


Figura I.1.1.1 Vista superior de un Scanner

De un modo general todos los scanners manuales están contruidos así, con algunas variaciones en los componentes utilizados.

En la anterior gráfica tenemos las ventana de exploración, que recibirá la luz reflejada por la imagen al ser iluminada por la barra de leds, esta luz será reflejada por el espejo montado en ángulo de 45 grados en relación al plano de la imagen explorada, dirigiéndola hacia el objetivo, montado en una cámara cerrada. A través del objetivo de la parte de la imagen proveniente de la ventana alcanza el sensor CCD que es, por así decirlo, el alma de este tipo de scanner. Este componente que físicamente se asemeja a un circuito integrado DIP (Dual Line Package) de 22

pins, contiene una ventana de cuarzo larga (en casi toda su longitud) que permite a la luz alcanzar el "mosaico" semiconductor que hará la exploración de la imagen, controlado por los circuitos electrónicos asociados.

Naturalmente la imagen que llega al sensor CCD corresponde a una pequeña parte de la imagen que está siendo explorada, de forma que precisamos mover el scanner hasta completar la exploración.

Durante el movimiento del scanner un rodillo cubierto de goma queda en contacto con la superficie explorada y gira, transfiriendo su rotación a través de la reducción mecánica al disco estriado.

Este disco es montado de tal forma que sus ranuras interrumpen el haz de luz del fotosensor, creando así un tren de pulsos cuando el scanner está en movimiento. Este tren de pulsos sirve para sincronizar el circuito de exploración montado en el CCD.

La señal generada por el sensor CCD es entregada al circuito electrónico, compuesto de varios circuitos integrados SMD, donde es procesada y enviada a la tarjeta de interface del micro, a través de un cable flexible.

En la placa de interface se ejecuta el acceso directo a la memoria (DMA), transfiriendo la "imagen electrónica" hacia la memoria de la microcomputadora, bajo el comando del software propio que acompaña al scanner. En general se puede ver la imagen a medida que se va formando en la pantalla simultáneamente a la exploración, permitiendo así controlar la calidad de imagen.

Scanner más recientes disponen de un led que indica si la velocidad a la que se está moviendo el scanner es la adecuada. Mientras la velocidad esté dentro del límite aceptable, el led permanece encendido. Si la velocidad está sobrepasando el límite, el led comienza a guñar y finalmente se apaga cuando la velocidad es excesiva. Las velocidades de exploración, para un scanner manual, varían de 20 mm/seg a 100 mm/seg, dependiendo de la densidad de exploración.

Los scanner poseen generalmente a un costado un botón que cuando es presionado habilita la exploración, permitiendo que se pueda ubicar convenientemente el scanner e interrumpir la exploración en cualquier momento, incluso después de su activación por software.

FUNCIONAMIENTO.

La mayoría de los scanners mandan luz a la página y miden cuanto luz vino de regreso. El blanco refleja la mayoría de la luz, el negro absorbe la mayoría de ella y varias formas de gris

CAPITULO I

reflejan luz en proporción a su densidad. La mayoría de los scanners miden la luz reflejada con un dispositivo de carga acoplada o CCD (Charge-Couple Device), el cual envía diferentes señales a la PC basadas en la cantidad de luz recibida. Las diferentes señales son enviadas desde el scanner a un coprocesador o al propio procesador de la PC para su procesamiento. De aquí en adelante lo que le suceda a las señales depende principalmente del software de aplicación.

La operación de un scanner típico es como sigue: El documento es iluminado por una luz fluorescente o incandescente, la luz reflejada pasa por un lente hacia un fotosensor también llamado dispositivo de carga acoplada CCD. El CCD es un arreglo lineal de elementos fotoeléctricos o detectores de luz. Cada elemento da un voltaje proporcional a la cantidad de luz que se refleja en él, un punto negro en el documento absorbe la mayor parte de la luz resultando en una salida de alto voltaje. Niveles de gris (o colores), entre estos extremos similarmente causa voltajes proporcionales a ser generados por los elementos CCD.

Los voltajes de salida analógica de los elementos CCD deben entonces ser convertidos a valores digitales, en un scanner simple de dos tonos un voltaje de umbral (usualmente ajustable) aproximadamente a la mitad del voltaje entre blanco y negro se selecciona como el punto de decisión, voltajes abajo del umbral se considera negro (0) y aquellos arriba del umbral significa blanco (1), este tipo de scanner utiliza por lo tanto un bit por pixel. En un scanner de gray-scale un convertidor analógico digital (ADC) convierte el voltaje de salida de cada elemento CCD en un patrón de bits apropiados que representan la intensidad de la luz reflejada dentro del número de bits por pixel soportados por el scanner, por ejemplo un ADC con resolución de cuatro bits dará valor entre 0000 (negro) y 1111 (blanco). Una vez que los voltajes analógicos son convertidos en valores digitales, la información resultante es almacenada localmente para su posterior procesamiento o bien es mandada a la PC para su procesamiento y/o almacenamiento.

Cada lectura de todos los elementos CCD representa un renglón completo de píxeles, sucesivos renglones son entonces leídos hasta que el documento entero o la imagen ha sido capturada. El CCD puede ser avanzado al siguiente renglón de píxeles por uno de los siguientes métodos: manualmente moviendo el scanner (de tipo manual), automáticamente moviendo la cabeza del scanner (del tipo flatbed) o automáticamente moviendo un espejo reflejante (scanner overhead).

En la siguiente gráfica se muestra un diagrama de bloques de mano.

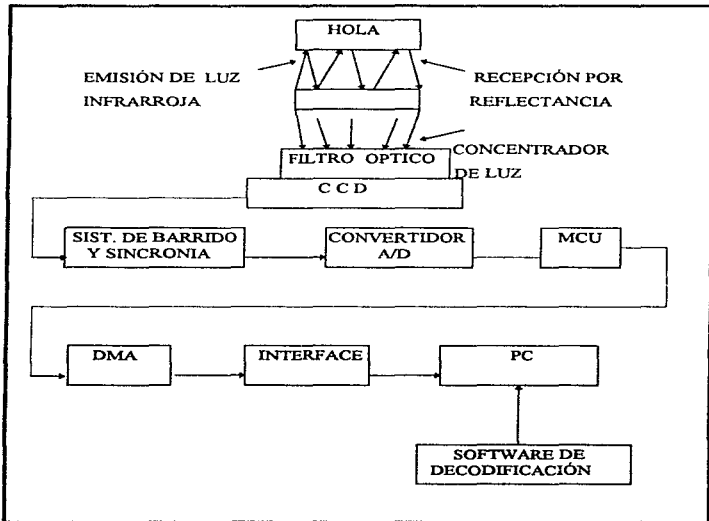


Figura L1.1.2 Diagrama de bloques de una scanner de mano.

CAPÍTULO I

I.1.2. CÓDIGO DE BARRAS

Código de Barras es la tecnología de identificación automática más avanzada disponible, aplicable a las personas y los objetos, exitosamente en uso en casi todo el mundo desde hace aproximadamente 20 años. Su objetivo es la identificación y localización repetitiva de productos a nivel industrial y comercial.

El sistema consta de series de líneas y espacios de distintos anchos, que almacenan información con distintos ordenamientos que se denominan simbologías.

La enorme aceptación ganada por estos sistemas se debe tanto a su exactitud, precisión y confiabilidad para la recolección automática y sistematizada de información impresa, como a su capacidad de establecer lazos de intercambio y comunicación de la información únicos entre el industrial y distribuidor de productos en gran escala, para consumo masivo.

EQUIPO DE LECTURA.

Los lectores son dispositivos utilizados para extraer información codificada en marcas ópticas conocidas como código de barras y convertirlos en datos digitales compatibles para el procesamiento en la computadora, pudiendo ser localmente almacenada para ser descargada más tarde o puede interactuar con algún programa de aplicación residente en el lector.

A continuación se muestra un diagrama de bloques de como se lleva a cabo el proceso de lectura del código de barras.

Considerando el anterior diagrama de bloques el lector de código de barras puede ser considerado como dos elementos separados: el **dispositivo de entrada** y **decodificador**.

El dispositivo de entrada es una unidad que emplea técnicas electro-ópticas para rastrear o examinar códigos. El rastreador móvil es transportado por el operador, por un mecanismo interno de rastreador o por el movimiento de símbolos al dispositivo de entrada.

En la figura I.1.2.3 se muestra un diagrama de bloques de un dispositivo de entrada manual.

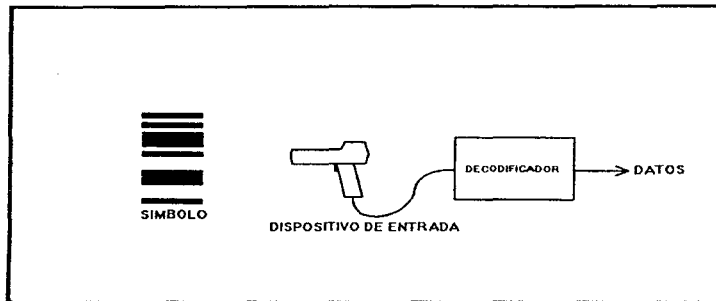


Figura I.1.2.1 Diagrama de bloques del proceso de lectura de código de barras.

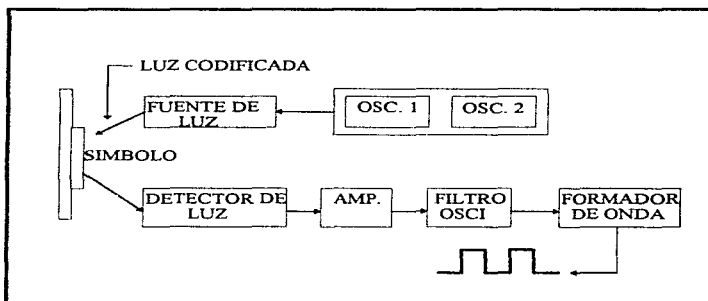


Figura I.1.2.2 Diagrama de bloques de un dispositivo de entrada para código de barras.

CAPÍTULO I

Un dispositivo de entrada de hecho es un sistema activo: ilumina el símbolo a ser leído con energía luminosa, entonces examina la cantidad de luz reflejada por el área localizada en el símbolo. La salida eléctrica del dispositivo de entrada es la representación de la luz reflejada por el símbolo.

La luz reflejada es mandada directamente a un fotodetector, el cual genera una pequeña corriente que es proporcional a la cantidad de luz recibida. Un amplificador posteriormente incrementa la señal del fotodetector para su fácil manejo. De tal forma de como se mueva el rastreador en el símbolo el voltaje analógico también variará. El voltaje analógico en el amplificador es proporcional a la luz reflejada por el símbolo.

El voltaje a la salida de un dispositivo de entrada puede ser analógica o digital. Si la salida es analógica entonces se necesitará de una etapa de conversión analógica a digital mediante un dispositivo llamado formador de onda (**waveshaper**).

El decodificador es la parte que analiza lo que recibe de la lectura del código o símbolo, produciendo la entrada a dispositivos periféricos. El resultado de la decodificación es transmitido a la computadora cargando localmente o remotamente la información para las aplicaciones residentes.

El proceso de la codificación se realiza con una aplicación de software o con un microprocesador; las tareas fundamentales que se realizan en la decodificación son las siguientes:

- a) Determinación del dispositivo de entrada así como el de salida.
- b) Determinación de la simbología, para decidir el tipo de técnica a utilizar y la distancia adecuada para la lectura.
- c) Determinación del ancho, angosto y tamaño del símbolo.
- d) Decodificar el símbolo de acuerdo con los estándares ya establecidos.
- e) Al leer el símbolo de derecha a izquierda utilizando un algoritmo para confirmar la decodificación y detectar si existe algún error.
- f) Confirmación y validación: confirmación de zonas mudas y chequeo de caracteres.
- g) Transmisión correcta de la secuencia de caracteres, emitiendo junto con estos los de chequeo.

Los decodificadores se dividen de acuerdo a su operación y comunicación en:

Decodificadores de línea. Decodifican directamente al equipo, el cual generalmente es una computadora.

Decodificadores portables. Contienen una pantalla y una batería recargable para facilitar la portabilidad. Este puede transmitir remotamente o ir almacenado los datos para después transferirlos a la computadora, utilizando memorias CMOS con capacidad de 16 kb a 512 kb.

Decodificadores inalámbricos portables. Son los más recientes, portables y pueden ser remotos, pero con la diferencia de que pueden trabajar en frecuencias VHF y UHF entre los 1200 a 4800 baudios.

Los dispositivos que son usados para leer un sistema de código de barras pueden ser lápices ópticos (**wands**) o scanners.

Los lápices ópticos son dispositivos de contacto, es decir, deben de tener contacto con la superficie donde se encuentra impreso el código de barras en el orden indicado para su lectura.

Los scanners son dispositivos de no contacto que leen el código a distancia, algunas veces a distancia de algunas pulgadas para scanners construidos por medio de diodos que utilizan tecnología láser, hasta 23 pulgadas de distancia. Los scanners que utilizan la tecnología CCD (Charge Couple Device) son capaces de leer códigos hasta una distancia de 10 pies.

En la figura I.1.2.3. (Pag. 4) se muestra un diagrama de bloques de un scanner con tecnología CCD.

El scanner lee el código impreso de la siguiente forma: primero lo ilumina con una luz de frecuencia comprendida entre el color rojo y el infrarrojo, barriendo el símbolo horizontalmente, esta luz es absorbida por las barras (oscuras) y reflejada por los espacios (claros); el mismo scanner recibe la luz reflejada, transformando en digital, la siguiente representación analógica:

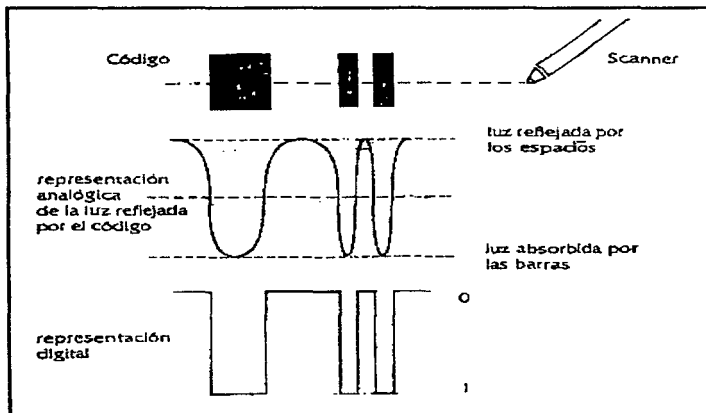


Figura I.1.2.4 Representación digital de la lectura de código de barras.

I.1.2.1 ESTÁNDARES.

La asociación nacional que agrupa a los distribuidores y/o industriales interesados en implementar un sistema de código de barras, se hace miembro del EAN (Asociación Internacional de Numeración de Artículos) y obtiene una identificación para el país de 2 o 3 dígitos llamada FLAG que permitirá reconocer internacionalmente el país de origen de cada producto.

El industrial solicita a su asociación nacional, encargada de la asignación de códigos, un conjunto de números que identificará a su empresa y será único para todos sus productos; luego podrá asignar él mismo otros conjuntos de números únicos para cada producto o forma de

presentación del mismo definiendo así una serie única de números para cada uno de sus productos llamada **CODIGO** que incluye:

" **PAÍS + EMPRESA + PRODUCTO + CONTROL** "

Este código se compone de un conjunto de barras verticales (símbolo para su lectura automática) y un conjunto de números impresos (código para su identificación individual por el hombre).

El industrial puede ahora utilizar este mismo código para la identificación de cada producto dentro de su empresa, en un sistema interno de producción, administración, contabilidad, venta y tráfico.

En los casos de exportaciones a otros países, se utiliza el código del fabricante y su país para la comercialización en todo el mundo, excepto en E.U. y Canadá donde se utiliza el código UPC (Universal Product Code) (Código de Producto Universal) que deberán llevar impresos correctamente los envases de productos que se exporten a esos dos países.

El industrial diseña sus envases ubicando correctamente el código para sus productos; y compra al fabricante de embalaje sus materiales para envasamiento con el código ya correctamente impreso y verificado, o etiquetas autoadhesivas codificadas que coloca en cada producto ya envasado (cuando la producción es pequeña).

Todos estos procesos requieren de normas y controles de calidad muy estrictos y nunca deben improvisarse ya que un embalaje mal impreso sólo afecta al producto, mientras que un código mal impreso impide la comercialización automática del producto, o lo que es mucho peor, puede producir costosas devoluciones de material de embalaje o productos envasados, de consecuencias muy onerosas especialmente en las exportaciones.

El distribuidor, por ejemplo el supermercado, adopta el código de cada producto para identificarlo dentro de su sistema interno de compras, administración, contabilidad, tráfico y ventas, para lo cual cuenta con un sistema central de computación en los sectores mencionados directamente conectado a la caja registradora.

Las cajas registradoras disponen de una ventana lectora llamada scanner, cada una abierta en la mesa donde un haz de luz, generalmente color rojo y de tipo coherente o láser, barre constantemente en 3 o más direcciones a gran velocidad explorando y analizando los objetos tridimensionalmente, a este proceso se le llama búsqueda (**scanning**).

AIM (Automatic Identification Manufacturers) Internacional:

CAPITULO I

Es una asociación comercial que representa a los fabricantes y vendedores de equipos, sistemas y abastecimientos para identificación automática, incluyendo: código de barras, identificación por radiofrecuencia, cintas magnéticas, reconocimiento óptico de caracteres, reconocimiento de voz y sistemas de visión. Esta es una industria joven de alta tecnología cuyos desarrollos tecnológicos se originaron inicialmente en los Estados Unidos.

AIM en los Estados Unidos ha servido como organización educativa y de promoción para la tecnología por medio de programas educativos incluyendo paneles de oradores, artículos, desarrollos publicitarios y el patrocinio de conferencias. La organización AIM ofrece a los usuarios potenciales de tecnología la confianza de viabilidad. Las tecnologías de identificación automática son básicamente periféricas a la computación, necesitan sistemas de computación para funcionar. Previamente al esfuerzo de desarrollo internacional de AIM los principales fabricantes de estas tecnologías ya estaban expandiendo sus esfuerzos de venta para incluir vendedores, distribuidores y representantes en otros países.

En 1983 AIM creó el Comité Técnico de Simbología (TSC) para proveer asistencia técnica al desarrollo de estándares genéricos para código de barras, el TSC creó 5 Especificaciones de Simbología Uniforme (USS) para describir las 5 simbologías de uso mas común en aplicaciones comerciales: USS-39, USS-12/5, USS-Codabar, USS-128 y USS-93.

1.1.2.2. TIPOS

EL CÓDIGO EAN 13

Es un sistema de codificación constituido por series de barras y espacios, paralelos, de ancho variable donde por lo general las barras son oscuras y los espacios claros; consta de una cantidad fija de barras (30 en total) y espacios (29 en total) que codifican información.

Permite codificar teóricamente en 1000 países u organizaciones adheridas a 10000 industrias distintas y a cada una de ellas 100000 productos o formas de presentación de los mismos, lo cual representa una enorme cantidad de posibles combinaciones.

Siendo éste un sistema común a varios países y productos dentro y fuera del mercado común europeo dispone, como es lógico, de un indicativo nacional llamado Flag, para cada país que identifica al organismo nacional de codificación que asignará los códigos localmente.

CARACTERÍSTICAS EAN 13.

El carácter numérico es un número de un dígito (0 1 2 3 4 5 6 7 8 9); con estos números se forma el código EAN 13 que precisa 13 caracteres.

De los trece caracteres que forman el código, 12 serán simbolizados e impresos por barras y espacios, para que el scanner pueda leerlos, y un carácter no será representado de esta manera.

Cada carácter numérico se representa por 2 barras + 2 espacios, ubicados alternativamente, o sea 4 elementos para cada carácter; el ancho y ubicación de los elementos diferencia a un carácter de otro. Se codifica cada módulo/barra = 1, cada modulo/espacio = 0. El ancho de cada carácter es fijo y mide 7 módulos (módulo es la unidad de menor ancho que forma los elementos).

Por lo tanto, los 4 elementos que forman un carácter también tendrán un ancho total de 7 módulos, es así que cada barra y/o espacio podrán tener un ancho como mínimo de 1 módulo, y como máximo de 4 módulos, siendo así un código de estructura compleja.

Estos criterios sólo se aplican a los 12 caracteres numéricos que se codifican en el sistema EAN 13, y no se aplican a los separadores, zonas mudas, ni al carácter cuya posición es la número 13.

Definimos la ubicación o posición de cada carácter en el código mirándolo de frente, posición número 1 la primera a la derecha del código y la posición número 13, la última a la izquierda del código.

Los elementos que integran el código EAN 13 son la barra o espacios y miden de 1 a 4 módulos de ancho cada uno.

La complejidad aparente de la codificación tiene por objeto permitir al decodificador del computador la identificación del código, seleccionar el sentido correcto en la lectura bidireccional del scanner, activar los mecanismos electrónicos de verificación y chequeo, detectar errores y evitar lecturas erradas.

A continuación se muestran las principales características del código EAN 13:

- Caracteres: 13 en total, numéricos solamente, asignados a:
posición número 13: identificación del país (Flag); este carácter es determinado por la secuencia de otros caracteres y no se le codifica con barras o espacios.
Posición número 12,11: identificación del país (Flag).
Posición número 10,9,8,7: identificación del fabricante del producto.
Posición número 6, 5, 4, 3, 2: identificación del producto.

CAPITULO I

Posición número 1: código de verificación (su valor es calculado).

- Separadores:

Izquierdo ancho fijo: 3 módulos (2 barras con 1 espacio en medio, codificado: 101).

Derecho, ancho fijo: 3 módulos (2 barras con 1 espacio en medio, codificado: 101).

Central ancho fijo: 5 módulos (3 espacios con 2 barras intercaladas, codificado: 01010).

Altura estándar de los separadores (HS): 24.50 mm (son un poco mas altos que las barras).

- Zonas mudas:

Izquierda: ancho mínimo 11 módulos, codificado: 00000000000

Derecha: ancho mínimo 7 módulos, codificado: 0000000.

Superior: 1 módulo como mínimo, por encima del código.

Inferior: 1 módulo, entre el código y la línea de interpretación.

- Codificación: continúa bidireccional.

- Línea de interpretación: al pie del código, los caracteres en posición número 1 al número 12.

En la zona muda izquierda: posición número 13.

- Uso del código controlado por: EAN y la organización nacional de codificación comercial propia de cada país.

- Módulo (M): ancho estándar teórico (para $f_m=1$): 0.33 mm.

- Longitud: fija total 113 módulos, entre señal de encuadre (AE).
95 módulos, entre extremos separadores (AS).

- Densidad: media.

- Tamaño estándar (para $f_m=1$), 37.29 X 26.26 mm, entre las señales de encuadre, incluyendo las 4 zonas mudas (a la derecha, izquierda, arriba y abajo del código).

- Altura del simbolo (barra o espacio) HB = 22.85 mm.

- Factores de magnificación (f_m):

$f_m = 2$ muy recomendado

$f_m = 1$ recomendado

$f_m = 0.8$ poco recomendado.

A continuación se muestra un esquema general del código EAN 13.

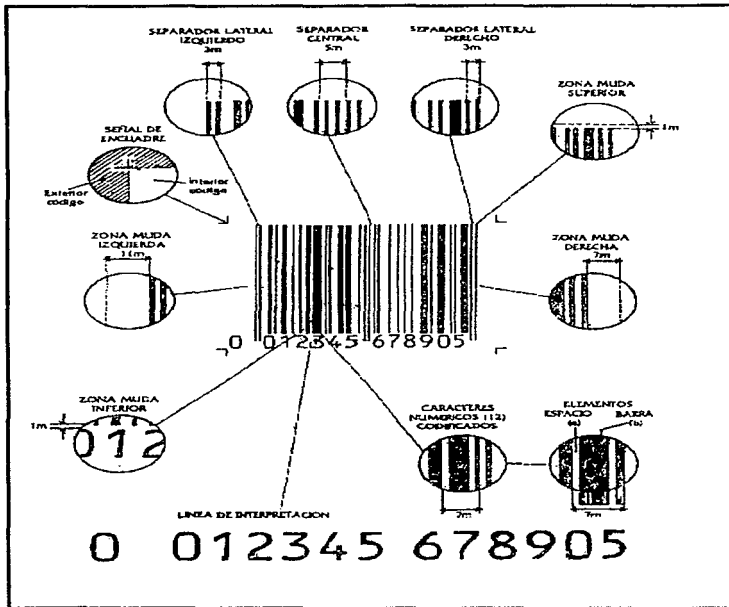


Figura L1.2.2.1 Características del código EAN 13.

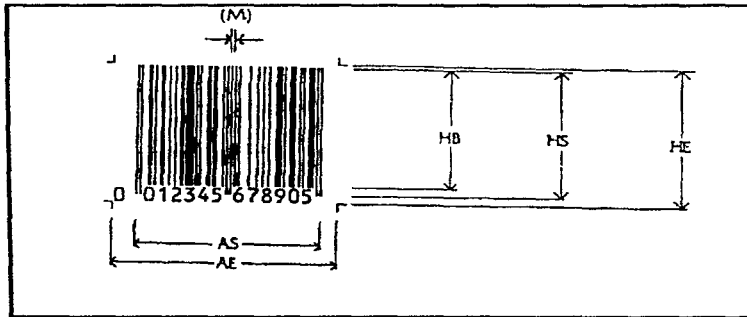


Figura I.1.2.2.2. Características del código EAN 13.

CÓDIGO TRUNCADO EAN

Cuando se dispone de un espacio reducido en el envase, que no es posible disponer de un código de tamaño adecuado y/o las características del sustrato y el sistema de impresión (escala patrón) no lo permiten; ni siquiera utilizando la versión reducida EAN 8, la última opción de todas es recortar la longitud de las barras o truncar el código.

Al reducir el largo del símbolo disminuye proporcionalmente la posibilidad de lectura omnidireccional para el scanner. Lo que quiere decir que el producto deberá ser maniobrado, rotado y deslizado por el operador, hasta lograr una lectura del scanner perdiendo mucho tiempo que es lo opuesto al objetivo del código de barras. A mayor truncamiento menor posibilidad de lectura.

Cuando se trunca un código debe tomarse un símbolo base con el factor de magnificación mayor que sea posible, evitando siempre truncar códigos de magnificación menor de 100% ($f_m=1.0$).

Para determinar el mínimo truncamiento posible, se utiliza la tabla de valores I.1.2.2.1 y se procede a realizar una primera reducción en la longitud de las barras (reducción 1), que dependerá del factor de magnificación del código base. Si esto no resulta aun suficiente se procederá a hacer un segundo truncamiento (reducción 2) teniendo presente que probablemente el código impreso NO podrá ser leído omnidireccionalmente.

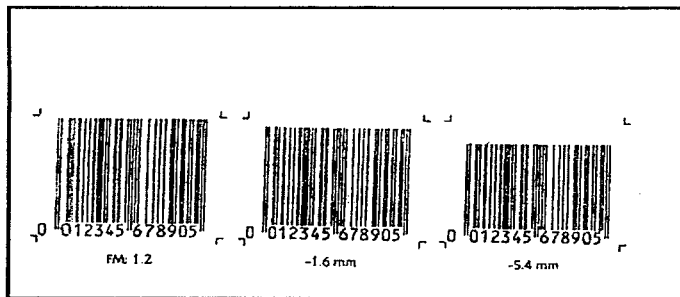


Figura I.1.2.2.3 Ejemplo de truncamiento del código EAN.

CÓDIGO EAN 8

Es la versión reducida del sistema EAN, que se utiliza exclusivamente cuando el tamaño y/o forma del envase no deja suficiente lugar disponible para imprimir el código EAN 13. Si bien es una versión reducida del código EAN 13, no es exactamente otra forma de almacenar la misma información, siendo un código EAN 8 totalmente independiente del EAN 13, no se da el mismo caso que el UPC E que es la forma reducida del UPC A en forma casi automática.

Ventajas de esta versión:

Es más pequeña, ocupa menos lugar.

Es igual de confiable y estable que el EAN 13, (a igual factor de magnificación).

CAPITULO I

Es preferible utilizar el código EAN 8 antes que truncar el EAN 13, principalmente si ha sido reducido.

Factor de magnificación símbolo base (fm)	Reducción proporcional de las barras (de arriba abajo)		
	Reducción 1 (mm) de: a:	Reducción 2 (mm)	Reducción máxima total
0.8 - 0.95	No truncar	No truncar	No
1.0	No truncar	3.8	3.8 3.8
1.1	0.8 0.8	3.8	4.6 4.6
1.2	1.5 1.6	3.8	5.3 5.4
1.3	2.3 2.4	3.8	6.1 6.2
1.4	3.0 3.2	3.8	6.8 7.0
1.5	3.8 4.0	3.8	7.6 7.8
1.6	4.6 4.8	3.8	8.4 8.6
1.7	5.3 5.6	3.8	9.1 9.4
1.8	6.1 6.4	3.8	9.9 10.2
1.9	6.9 7.2	3.8	10.7 11.0
2.0	7.6 8.0	3.8	11.4 11.8

Tabla I.1.2.2.1 Relaciones de truncamiento para el código EAN 13.

Desventajas del código EAN 8:

Capacidad de codificación más limitada.

El uso de esta versión no es optativo ni libre y debe ser asignado por la institución local de codificación.

Características EAN 8:

El carácter numérico es el número de 1 dígito (0 1 2 3 4 5 6 7 8 9) y al igual que en el código EAN 13 a partir del cual se le obtiene, con estos números se forma el código EAN 8 que precisa 8 caracteres.

Los 8 caracteres que forman el código serán representados e impresos por barras y espacios, para que el scanner pueda leerlos.

Cada carácter numérico se representa por 2 barras más 2 espacios, ubicados alternativamente, o sea 4 elementos para cada carácter.

El ancho y ubicación de los elementos diferencia a un carácter de otro. Se codifica cada módulo/barra = 1, y cada módulo/espacio = 0.

El ancho de cada carácter es fijo y mide 7 módulos (módulo es la unidad de menor ancho que forma los elementos). Por lo tanto los elementos que forman a un carácter tendrán un ancho total de 7 módulos, es así que cada barra y/o espacio podrán tener un ancho como mínimo de 1 módulo, y como máximo de 4 módulos siendo así un código de estructura compleja.

Estos criterios, sólo se aplican a los caracteres numéricos que se codifican en el sistema EAN 8, y no se aplican a los separadores ni a las zonas mudas.

Definimos la ubicación o posición de cada carácter en el código mirándolo de frente, posición número 1 la primera a la derecha del código y posición número 8 la última a la izquierda del código.

Características del código EAN 8.

- Caracteres. 8 en total, numéricos solamente, asignados a:
 - Posición número 8,7 identificación del país. (Flag).
 - Posición número 6,5,4 identificación del fabricante (o del producto).
 - Posición número 3,2 identificación del producto.
 - Posición número 1 dígito de verificación (su valor es calculado).

CAPITULO I

- **Separadores:**
 - Izquierdo, ancho fijo de 3 módulos (2 barras con 1 espacio en medio, codificado 101).
 - Derecho, ancho fijo de 3 módulos (2 barras con 1 espacio al medio, codificado 101).
 - Central, ancho fijo de 5 módulos (3 espacios con 2 barras intercaladas, codificado 01010).

Altura estándar de los separadores (HS) es de 19.88 mm. (son un poco más altos que las barras).
- **Zonas mudas:**
 - Izquierda, ancho mínimo 7 módulos, codificado 0000000.
 - Derecha, ancho mínimo 7 módulos, codificado: 0000000.
 - Superior 1 módulo, como mínimo, por encima del código.
 - Inferior 1 módulo, entre el código y la línea de interpretación.
- **Codificación es continua, bidireccional.**
- **Línea de interpretación.** Al pie del código los caracteres en posición del número 1 al 8.
- **Uso del código controlado por EAN y la organización nacional de codificación de cada país.**
- **Módulo M:** ancho estándar teórico de 0.33 mm.
- **Longitud:** fija total 81 módulos, entre señales de encuadre (AE), y 67 módulos entre extremos separadores (AS).
- **Tamaño estándar:** 26.73 x 21.64 mm, entre las señales de encuadre, incluyendo las 4 zonas mudas.
- **Altura del carácter numérico (barra o espacio):** HB=18.23 mm.

A continuación se muestra un ejemplo del código EAN 8:



Figura 1.1.2.2.4 Ejemplo de código EAN 8.

CÓDIGO UPC A (Universal Product Code)(Código Universal de Producto).

Este tipo de código es uno de los más populares, fue creado y adoptado por la industria norteamericana en 1973 para su lectura en las cajas registradoras (punto de venta); existen dos versiones: UPC A y UPC E.

Características UPC A

Está formado por 12 caracteres en total. Numéricos solamente, es decir:

Carácter número 12 categoría del producto (medicinal, alimenticio, etc).

Carácter número 11, 10, 9, 8, 7 identificación del fabricante del producto.

Carácter número 5, 4, 3, 2 identificación del producto.

Carácter número 1 dígito de verificación del código.

Los caracteres número 1 y 12 se imprimen con barras más largas que las demás. El carácter numérico es el número de 1 dígito (0 1 2 3 4 5 6 7 8 9), con el cual se forma el código UPC A que precisa de 12 caracteres.

CAPITULO I

Los 12 caracteres que forman el código, serán representados e impresos por barras y espacios para que el scanner pueda leerlos.

Cada carácter numérico se representa por 2 barras + 2 espacios, ubicados alternativamente, o sea 4 elementos para cada carácter; el ancho y ubicación de los elementos diferencia a un carácter de otro.

El ancho de cada carácter es fijo y mide 7 módulos (módulo es la unidad de menor ancho que forma los elementos). Por lo tanto, los 4 elementos que forman a un carácter también tendrán un ancho de 7 módulos, es así que cada barra y/o espacio podrán tener un ancho como mínimo de 1 módulo, y como máximo de 4 módulos, siendo así un código de estructura compleja.

Estos criterios sólo se aplican a los 12 caracteres numéricos que se codifican en el sistema UPC A, y no se aplican los separadores ni en las zonas mudas.

Definimos la ubicación o posición de cada carácter en el código, mirándolo de frente, posición número 1 la primera a la derecha del código y posición número 12, la última a la izquierda del código.

Los primeros 6 dígitos están separados de los 6 segundos dígitos por unas barras de seguridad centrales. Las dos partes del símbolo están protegidas por dos barras izquierdas y dos barras derechas ambas de seguridad. Estas barras de seguridad pueden ser el patrón de inicio/final.

Los unos binarios que son contabilizados en la mitad izquierda, siempre suman impares y la suma de los unos de la mitad derecha siempre suman pares, de tal manera que por medio de un estudio del dato de paridad el rastreador (scanner), pueda decodificar en forma independiente la parte derecha o izquierda del símbolo, identificando inmediatamente que mitad fue decodificada.

Características generales.

• Separadores:

Izquierdo, ancho fijo: 3 módulos (2 barras con 1 espacio en medio, codificado= 101).

Derecho, ancho fijo: 3 módulos (2 barras con 1 espacio en medio, codificado= 101).

Central, ancho fijo: 5 módulos (3 espacios con 2 barras intercaladas, codificado= 01010).

Altura estándar de los separadores (HS): 24.50 mm (son un poco más altos que las barras).

• Zonas mudas:

Izquierda, ancho mínimo 11 módulos (codificado 0000000000).

Derecha, ancho mínimo 7 módulos (codificado 0000000).

Inferior, 1 módulo, entre el código y la línea de interpretación.

- Codificación es continua y bidireccional.
- Línea de interpretación: Al pie del código de los caracteres del número 2 al 11.
En la posición muda izquierda posición número 12; en la zona muda derecha posición número 1.
- Módulo M, ancho estándar teórico 0.33 mm.
- Longitud, 113 módulos entre señales de encuadre(AE) y 95 módulos entre extremos separadores.
- Tamaño estándar, 37.29 x 25.93 mm, entre las señales de encuadre, incluyendo las zonas mudas.
- Altura del símbolo (barra o espacio), HB= 22.85 mm excepto los caracteres en la posición número 1 y 12 que son más largos, al igual que los separadores.

- Factores de magnificación (fm):
fm= 2.0 muy recomendado.
fm= 1.0 recomendado.
fm= 0.8 poco recomendado.
- Compatibilidad, puede ser leído en el sistema EAN que interpretará un carácter = 0 más a la izquierda, en la posición número 13.
En la figura I.1.2.2.5 se observa un ejemplo del código UPC A.

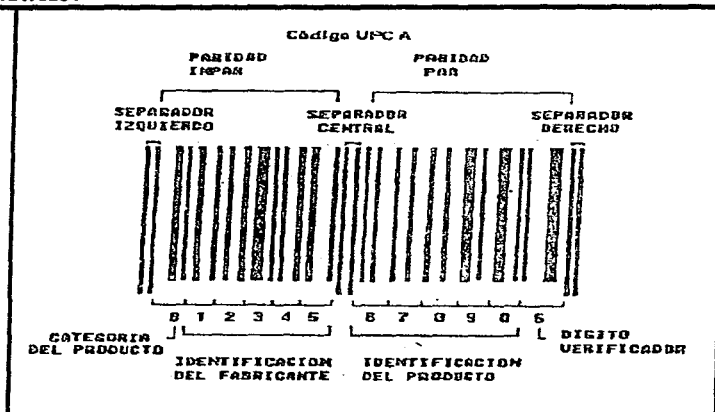


Figura I.1.2.2.5 Ejemplo y características del código UPC A.

CÓDIGO UPC E (código reducido)

Esta versión se llama "Cero Suprimido" (zero suppressed) ya que elimina por lo menos 4 ceros en el código. No siempre es posible su uso ya que esto depende del número del fabricante y el número del producto asignado. Es muy común creer que la versión UPC E se trata simplemente de quitarle los ceros al código UPC A, pero nada está más alejado de la realidad, ya que existen 4 formas de supresión de ceros, dependiendo de los tipos de números que le serán asignados al fabricante y producto, obedeciendo a normas muy estrictas de aplicación, que determinan en cada caso cuántos son los artículos que podrán disponer de un código reducido UPC E. Las cuatro combinaciones de supresión de ceros son las siguientes:

Si el número del fabricante termina en 00, precedido por 0, 1 ó 2; 1000 productos podrán ser codificados.

Si el número del fabricante termina en 00, precedido por 3 al 9; 100 productos podrán ser codificados.

Si el número del fabricante termina en 0; 10 números de producto podrán ser asignados.

Si el número del fabricante no termina en 0; sólo 5 productos podrán utilizar la versión reducida.

Los caracteres de identificación del fabricante y del producto se codifican por un método especial que permite eliminar dígitos cuyo valor es igual a cero, la supresión de los mismos depende de su ubicación en la versión estándar UPC A.



Figura L1.2.2.6 Ejemplo del código UPC E.

CÓDIGO 39 (3 de 9)

Este Código fue creado en 1974 y adoptado por el Departamento de Defensa (LOGMARS), Administración de Servicios (GSA) e Industria automovilística (ALAG) en Norteamérica, aproximadamente en 1982, adoptando el SCS (shipping container system), luego llamado UNIFORM CONTAINER SYMBOL para depósito y distribución; diseñado originalmente para ser impreso sobre cartón corrugado.

CAPITULO I

USS 39 es la especificación del código 39; USS (uniform symbol specification), fue desarrollado por el Comité de Simbología Técnica de AIM establecido en 1983 por AIM inc., este comité se compone de personal de varias empresas que son elegidos por los asociados.

USS 39 (código 39) es una simbología de código de barras con un conjunto de caracteres alfanuméricos completos: un único carácter inicial/final y siete caracteres especiales. El nombre 39 deriva de su estructura de elementos anchos de un total de nueve elementos. Los nueve elementos consisten en cinco barras y cuatro espacios cada uno.

El código 39 cuenta con las siguientes características:

Conjunto de 26 letras mayúsculas, 10 dígitos y 7 caracteres especiales, expandible a los 128 caracteres del código ASCII con las dos marcas y final.

Longitud del símbolo: variable.

Carácter de chequeo: opcional.

Carácter de cabecera: 2 por símbolo.

Otras características: Habilidad de concatenación.

Densidad máxima de 9.8 caracteres por pulgada, cuando la impresión utiliza 7.5 mm por dimensión.

Cada carácter del código 39 consiste de 5 barras y 4 espacios, 3 de los nueve elementos son anchos y 6 delgados. Cada carácter puede contener 2 ó 0 barras anchas. El carácter asterisco es utilizado exclusivamente como marca de inicio/final.

En la figura I.1.2.2.7 se muestra un ejemplo del código 39.

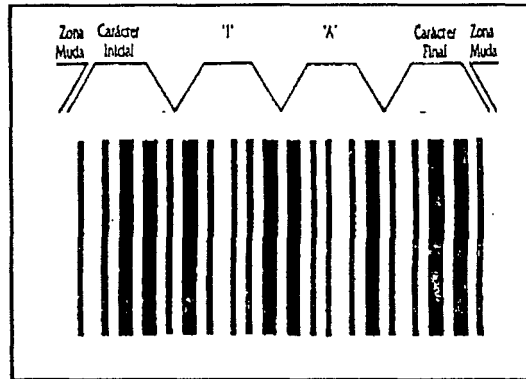


Figura I.1.2.2.7 Ejemplo y características del código 39.

CODABAR.

Originalmente desarrollado en 1972, y fue adoptado por la Comisión Americana de Bancos de Sangre como el estándar para identificar las bolsas de sangre; en la actualidad se utiliza comúnmente en las bibliotecas, librerías, otras aplicaciones médicas y en aplicaciones de servicios de paquetería aérea.

El codabar es una simbología discreta de longitud variable, con verificación bidireccional y el número puede ser extendido. Contiene un conjunto de 20 caracteres los números del 0 al 9 y los caracteres \$, ;, /, +, -. Hay cuatro diferentes caracteres de inicio y final: A, B, C y D.

Cada carácter consiste de siete elementos, cuatro barras y tres espacios con dos o tres elementos que pueden ser anchos (uno binario), y el resto angosto (cero binario). Los caracteres

CAPITULO I

de datos están delimitados por caracteres de inicio y final. La dimensión de las barras y espacios de cada carácter del Codabar está determinada por ANSI.

El aspecto más peculiar de Codabar es que la anchura de las barras y espacios para impresos puede tomarse en 18 diferentes valores dependiendo del carácter particular.

Ignoremos por un momento las 18 diferentes anchuras y consideremos que todos los elementos pueden ser anchos o angostos. Codabar utiliza tres esquemas diferentes de codificación de caracteres:

- 1) Los dígitos del cero al nueve (0-9), los caracteres "\$" y "-" son impresos con una barra ancha y un espacio ancho; todos los demás caracteres o elementos son angostos.
- 2) Los cuatro caracteres especiales ("S", "/", ".", "+") son codificados con tres barras anchas y sin espacios anchos.
- 3) Los cuatro caracteres de inicio/terminación (A,B,C y D) son codificados con una barra ancha y dos espacios anchos.

Cuando se imprime en este estilo cada carácter tiene la misma anchura. Muchas de las dimensiones publicadas difieren sólo por algunos diez milésimos de pulgada, la cual es obviamente insignificante cuando se considera la típica tolerancia de impresión, la mayoría de impresores de Codabar usan un menor número de dimensiones. En la figura I.1.2.2.8 se observa un ejemplo del código utilizado por Codabar.

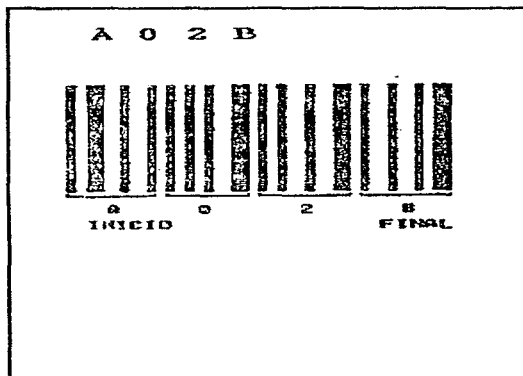


Figura I.1.2.2.8 Ejemplo de código Codabar.

I.1.2.3. PROCEDIMIENTOS PARA SU IMPLEMENTACIÓN.

La información se procesa y almacena con base en un sistema digital binario donde todo se resume a sucesiones de unos y ceros. El sistema central de decisiones lógicas es un computador electrónico del tipo estándar, disponible ya en muchas empresas comerciales y generalmente compatible con las distintas marcas y modelos de preferencia en cada país. Estos equipos permiten también interconectar entre sí distintas sucursales o distribuidores centralizando toda la información.

La incorporación del código de barras y sus sistemas electrónicos en una empresa, es equivalente al aprendizaje de un idioma nuevo, requiere de tiempo, estudio y práctica. Por lo tanto es muy recomendable tener en cuenta los siguientes conceptos generales:

- Asesorarse lo mejor posible en todas las áreas.

CAPITULO I

- Informarse sobre situaciones similares ya existentes.
- Contactar todas las personas y áreas, dentro de la empresa, que serán afectadas de una u otra manera, por el nuevo sistema a implementar.
- El responsable del desarrollo deberá compenetrarse también en los detalles con los proveedores de equipos y sistemas, instaladores, asesores, asociación local de codificación, proveedores de productos, impresores, convertidores, fabricantes de envases y todas aquellas áreas que directa o indirectamente se relacionen al código de barras y su utilización.
- Darle al proyecto e implementación todo el tiempo necesario, evitando siempre las improvisaciones y las soluciones inadecuadas o provisorias.
- Asignar al equipo de trabajo, personal con amplia experiencia en sistemas e interpretación del comportamiento comercial, que sea capaz de organizar el aprovechamiento de futura información.

1.1.2.4. BENEFICIOS QUE OFRECE EL SISTEMA.

- Permite la marcación única del producto desde la fuente primaria de producción hasta el consumidor, a quien permite saber exactamente qué acción está realizando, evitando también posibles equivocaciones.
- Información precisa de los tiempos y ciclos de producción, inspección, almacenamiento, transporte y venta.
- Informaciones estadísticas en general.
- Mínimo de errores en la información, ya que incluyen sistemas de autoverificación y/o caracteres de control dentro de sí mismos, que eliminan los errores de lectura.
- Sobredimensionado vertical, que permitirá leer un código aún cuando solamente un 5 % de su altura permanezca legible, ante la eventual destrucción del código impreso.
- Velocidad y eficiencia en la recepción, venta y cobranza, especialmente en los módulos de prestación de servicios.
- Se elimina la necesidad del remarcado de precios y/o la codificación manual e individual producto por producto, especialmente en los países de alta inflación.
- Información en tiempo real de inventario, venta y reposición de productos.

- Eliminación de errores humanos de marcación, interpretación, facturación al cliente y pérdida desconocida.
- Adaptable a la mayoría de los sistemas de embalaje, impresión y materiales de envasamiento existentes.
- Permite realizar inventarios de productos rápidamente y además saber que tanto se tiene en almacén.
- Fácilmente adaptable y compatible a muchos de los distintos sistemas y marcas de computadoras disponibles en cada país.

I.2. REDES DE COMPUTADORAS.

El surgimiento de las microcomputadoras a fines de la década de los 70, atrajo la atención de millones de usuarios que vieron en ellas la capacidad de cómputo a bajo costo, con autonomía y versatilidad. El mejoramiento de sus características, velocidad y capacidad del microprocesador, memoria principal y secundaria, les permitieron, no sólo capturar el mercado de estudiantes y usuarios particulares, sino también competir favorablemente contra minicomputadoras y equipos grandes en aplicaciones de negocios, en empresas de todos tamaños, al grado de desplazarlos paulatinamente de las salas de cómputo. Se estima que el mercado de las computadoras personales tipo PC, modelo con que la empresa IBM participa en el mercado de los equipos desde 1981, crece anualmente más del 50%, mientras que el de los equipos mayores crece en tasas menores al 15%.

Sin embargo, para alcanzar la amplia aceptación que han tenido las microcomputadoras, se han debido superar algunas deficiencias que habían persistido pese al mejoramiento de sus características individuales. Por ejemplo, es altamente incosteable conectar de manera dedicada cierto tipo de periféricos a cada microcomputadora de que se dispone. Igualmente ineficiente resulta compartir información, paquetes o herramientas de programación entre varios equipos, ya que ello implica duplicarlos.

La solución de éstos y otros inconvenientes fue la estructuración de redes de microcomputadoras, es decir, la interconexión y operación de este tipo de equipos en un ambiente homogéneo, que les permitiera compartir recursos e información de manera eficiente. Las redes

CAPITULO I

además, han traído consigo ventajas adicionales respecto a los otros sistemas computacionales, como se verá más adelante.

El concepto de red se ha ampliado enormemente, incluyendo equipos minicomputadores y equipos grandes, rebasando el ámbito de una sala de cómputo hasta alcanzar con sus nodos diversos puntos de una ciudad o un país; utilizando como medio de conexión, desde líneas telefónicas convencionales hasta las comunicaciones via satélite. Actualmente la Tecnología de Redes ha creado sus propias herramientas de programación y comunicaciones y se ha convertido de hecho en la rama de la computación con mayor desarrollo.

1.2.1. CONCEPTOS GENERALES.

¿Cómo surgen las Redes?

Como se vio, las microcomputadoras han ido desarrollándose fuertemente desde fines de los 70's. Esta explosión, particularmente en las PCs ha llevado a muchas empresas a poseer más de un equipo, lo que ha provocado problemas como los que a continuación se describen:

- **Compartición de datos y programas:** Si bien una micro hace mucho más productivo el trabajo de una persona, cuando alguien necesita de ciertos datos (archivos) que se encuentran dentro del disco duro de la PC del vecino, la solución más común es tomar un diskette y pedirle los datos al vecino. Esto ocasiona muchas inconveniencias, tales como el propio tiempo de tomar los datos y copiarlos de un equipo a otro, el hecho de que un archivo se encuentre en varios equipos, lo que dificulta su actualización y genera desperdicio en disco.
- **Compartición de dispositivos:** Con el número creciente de micros, los precios de los periféricos, entre ellos las impresoras, han bajado considerablemente, por lo que no es costoso que en cada micro se conecte una impresora ¿Porqué entonces la necesidad de compartir una impresora? En general si bien a las empresas no les impacta mucho el adquirir una impresora por cada micro (refiriéndose a las populares impresoras de matriz), lo que sí resulta costoso es el adquirir una impresora láser para cada equipo, de manera que estos dispositivos caros, es normal querer compartirlos.
- **Falta de estandarización en el software:** Entre más micros se tienen en una empresa, es más probable que los paquetes de software que se utilizan sean cada vez más diversos. Esto se refiere a que en vez de que todo el personal utilice un sólo tipo de procesador de palabras, es

común que si en una empresa existen 20 o 30 micros, algún usuario utilice el procesador que a el más le guste. Lo anterior provoca la imposibilidad de intercambiar archivos, costos mayores de entrenamiento y mayor inversión en software.

- **Poca Seguridad:** En su concepción inicial, una micro fue pensada como un instrumento de trabajo personal, de forma que aspectos tales como alta seguridad no se contemplaban. Sin embargo, conforme las micros han ido evolucionando, las aplicaciones y datos que ahí se trabajan, son cada vez más estratégicos para las empresas.
- **Falta de integración con equipos mayores:** Tipicamente, los grandes usuarios tienen además de muchas micros, una o más minis y/o mainframes, y con este tipo de usuario, las aplicaciones críticas (o la mayoría de ellas) son ejecutadas en dichos equipos mayores. De manera que en un esquema tradicional, las micros viven separadas de los equipos grandes, creando problemas asociados a esto, tales como la recaptura de la información, o la discordancia de ésta entre una micro y los equipos grandes. La problemática anterior originó que muchas organizaciones que contaban con un número considerable de computadoras en operación, con frecuencia alejadas unas de otras, decidieran interconectarlas para tener así la capacidad de extraer y correlacionar información referente a toda la organización.

OBJETIVOS DE LAS REDES

Puesto en una forma más general, la idea que se trata de transmitir consiste en compartir recursos, y el objetivo es hacer que todos los programas, datos y equipo estén disponibles para cualquiera de la red que así lo solicite, sin importar la localización física del recurso y del usuario. En otras palabras, el hecho de que el usuario se encuentre a 1000 Km de distancia de los datos, no debe evitar que éste los pueda utilizar como si fueran originados localmente. Otro aspecto de compartir recursos es el relacionado con la compartición de la carga. Este objetivo se puede resumir diciendo que es un intento por terminar con las restricciones geográficas intrínsecas de las comunicaciones.

Un segundo objetivo consiste en proporcionar una alta fiabilidad, al contar con fuentes alternativas de suministro. Por ejemplo, todos los archivos podrían duplicarse en dos o tres máquinas, de tal manera que si una de ellas no se encuentra (como consecuencia de una falla de hardware), podría utilizarse alguna de las otras copias. Además la presencia de múltiples CPU

CAPÍTULO I

significa que si una de ellas deja de funcionar, las otras pueden ser capaces de encargarse de su trabajo, aunque se tenga un rendimiento global menor.

Otro objetivo es el ahorro económico. Las computadoras pequeñas tienen una mejor relación costo/rendimiento, comparada con la ofrecida por las máquinas grandes. Estas son, a grandes rasgos, diez veces más rápidas que el más rápido de los microprocesadores, pero su costo es miles de veces mayor. Este desequilibrio ha ocasionado que muchos diseñadores de sistemas construyan sistemas constituidos por poderosas computadoras personales, una por usuario, con los datos guardados en una o más máquinas que funcionan como servidor de archivo compartido.

Este objetivo conduce al concepto de redes con varias computadoras localizadas en un mismo edificio. A este tipo de red se le denomina LAN (Red de Área Local), ejemplificada en la Figura 1.2.1.1, en contraste con lo extenso de una WAN (Red de Área Extendida o Amplia), mostrada en la Figura 1.2.1.2, a la que también se le conoce como red de gran alcance.

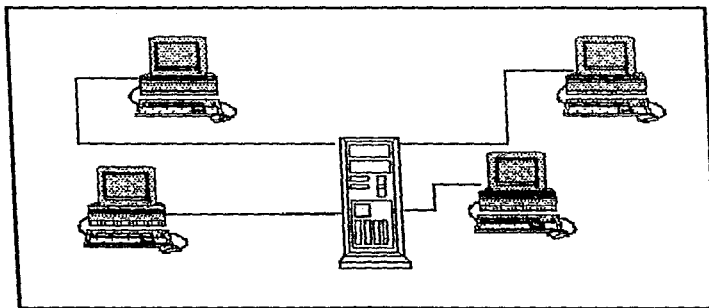


Figura 1.2.1.1 Redes de Área Local(LAN).

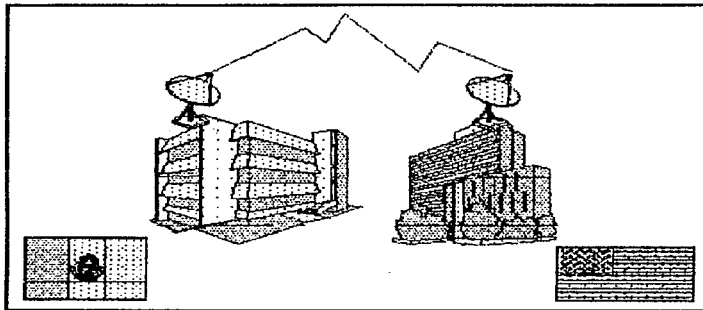


Figura 1.2.1.2 Redes de Área Extendida o Amplia (WAN).

Un punto muy relacionado es la capacidad para aumentar el rendimiento del sistema en forma gradual a medida que crece la carga, simplemente añadiendo más procesadores. Con máquinas grandes deberá reemplazarse con uno aún más grande, operación que por lo normal genera un gran gasto y una perturbación inclusive mayor al trabajo de los usuarios.

Otro objetivo del establecimiento de una red de computadoras no tiene nada que ver con la tecnología. Una red de computadoras puede proporcionar un poderoso medio de comunicación entre personas que se encuentran muy alejadas entre sí. Con el empleo de una red es relativamente fácil para dos o más personas, que viven en lugares separados, escribir un informe juntos.

VENTAJAS Y DESVENTAJAS DE LAS REDES.

El desarrollo alcanzado por las redes de microcomputadoras las ha hecho reunir las características deseables de los demás sistemas computacionales, evitando muchas de las deficiencias que ellos poseen, constituyéndose así en la solución más adecuada para una gran variedad de aplicaciones. Todo ello a cambio de muy pocas inconveniencias.

CAPITULO I

Por una parte, las redes han superado un conjunto de limitaciones que presentaban las micros por sí solas. Una de ellas es la dificultad para compartir programas y datos. En el caso de las micros, los archivos debían ser transferidos de un equipo a otro mediante discos flexibles, lo cual, para grandes volúmenes de información es poco práctico e implica el riesgo de contagio de virus. En un ambiente de redes, las bases de datos, paquetes y programas de uso común, son instalados en el servidor de la red, donde pueden ser utilizados fácilmente por los interesados y las transferencias de archivos se hacen internamente, de una sección a otra del área de almacenamiento.

Las redes permiten también aprovechar mejor otro tipo de recursos como impresoras, sobre todo las de calidad, cuyo precio es considerablemente alto. En el caso de microcomputadoras aisladas, estos periféricos deben estar cambiando de lugar, provocándose desperfectos y acortamiento de su vida útil.

Una deficiencia de las micros que también es superada con las redes es la estandarización de los programas. En micros individuales, cada usuario debe tener su propio procesador de palabras, hoja de cálculo o manejador de bases de datos, tal vez el mismo paquete pero con diferente versión. Todo ello dificulta el intercambio de información. Mediante las redes, se utiliza un solo paquete de cada tipo y por lo tanto, el intercambio de datos es completamente natural.

En la comparación entre equipos macro y mini y redes de microcomputadoras, éstas también pueden tener puntos a favor. El primero y más importante es el precio. Los equipos mini cuestan en promedio, cuatro o cinco veces más que los equipos con procesador 80486 O PENTIUM, que son los más poderosos. La proporción con el precio de los equipos grandes es aún mayor.

Otra ventaja relacionada con la anterior, es que gracias a los estándares (tanto "de jure" como "de facto"), existe una gran cantidad de fabricantes de equipo y componentes así como de desarrolladores de herramientas de programación. Ello hace que los usuarios no dependan de un sólo proveedor que imponga las condiciones y limite las soluciones de los problemas a las herramientas que vende. Por el contrario, se ha dado origen a una fuerte competencia entre distintos proveedores para ofrecer productos de mejor calidad y precio, de lo que se han beneficiado los usuarios.

Una ventaja adicional es que mediante las redes ha surgido la posibilidad de descentralizar los servicios de cómputo. En lugar de una sola área responsable de diseñar y desarrollar sistemas,

que invariablemente se convertía en un cuello de botella, ahora los usuarios finales disponen de una gran cantidad de herramientas informáticas que les permiten desarrollar sus propias aplicaciones, apegados a sus necesidades, haciendo su trabajo más productivo.

En cuanto a las desventajas de las redes de microcomputadoras con respecto a los equipos grandes, podría decirse que existen aplicaciones en las que, por su magnitud y demanda de recursos, las primeras por sí solas no tienen todavía un desempeño adecuado. En estos casos, las soluciones de conectividad permiten la incorporación de equipos macro como servidores de archivos o de bases de datos manteniendo la transparencia para el usuario, en el manejo de la información. Además, de acuerdo con las tendencias tecnológicas, se espera que en un futuro cercano, prácticamente no existan diferencias notables entre la capacidad de proceso de las micros y de otro tipo de equipos.

Realmente, el mayor inconveniente de las redes es la posible falta de compatibilidad que puede existir entre productos de distinto fabricante. Pese a los esfuerzos de distintas organizaciones para definir estándares, la adopción de éstos, no garantiza en todos los casos la interoperabilidad de los productos, ya sean equipos o herramientas de programación. Existen características en cada producto, fuera de los estándares, que son definidas por cada fabricante y que pueden afectar la operación de la red, o de una aplicación en particular.

En el caso de sistemas de un solo fabricante, como macros y minis, la responsabilidad de verificar la interoperabilidad entre los productos, que van a componer el sistema, la asume el proveedor. En las redes el cliente o usuario debe hacer una elección cuidadosa de los elementos que van a componer la red, o de los que se vayan a agregar a una ya existente, sobre todo si estos provienen de distinto fabricante, a fin de evitar problemas de compatibilidad. De lo contrario no se tendrá el rendimiento esperado y se perderán las ventajas que éste tipo de esquema ofrece.

ELEMENTOS DE UNA RED DE COMUNICACION

Los elementos con que cuenta una red pueden variar dependiendo de las funciones específicas para las que este diseñada. sin embargo existen elementos comunes a todas ellas que son los que se enuncian a continuación:

Estación de Trabajo: Son las computadoras o terminales, desde las cuales el usuario puede utilizar la red.

CAPITULO I

Servidor (Server): Es un dispositivo que proporciona una función especial a todos los usuarios de la red. Entre los tipos de servidores más importantes se encuentran los siguientes:

- **Servidor de archivos:** Provee área de almacenamiento y acceso a programas y archivos de datos compartidos.
- **Servidor de impresión:** Controla las colas de impresión y da acceso a la o las impresoras conectadas a él.
- **Servidor de Base de Datos:** Equipo dedicado al almacenamiento y organización de base de datos y a la recuperación de los datos solicitados en las consultas.
- **Servidor de Comunicaciones:** Equipo dedicado a atender las comunicaciones entre estaciones de trabajo remotas y los demás dispositivos de la red.

En realidad los equipos que operan como servidores son computadoras semejantes a las que se utilizan como estaciones de trabajo, aunque generalmente con mayor capacidad, en los cuales se ejecuta un programa que les permite desempeñar la función especial que desarrollan. Cuando el equipo únicamente desempeña esa función se llama servidor dedicado. Cuando además se utiliza como estación de trabajo se llama servidor no dedicado.

Existen microcomputadoras muy poderosas (basadas en microprocesadores INTEL 80386, 80486 y PENTIUM) que pueden desempeñar eficientemente varias funciones de servidores a la vez. A dichos equipos se les denomina servidores de redes.

REDES DE AREA LOCAL

Las Redes de Area Local (LAN, Local Area Network) se describen a veces como aquellas que "cubren una área geográfica limitada...", donde todo "nodo de la red puede comunicarse con todos los demás, y ... no requiere un nodo procesador central". una definición complementaria como la que ha dado Lee A. Bartman, sugiere que una LAN "es una red de comunicación que puede ofrecer intercambio interno entre medios de voz, datos de computadora, procesamiento de palabras, facsímil, videoconferencias, transmisión televisiva de video, telemetría y otras formas de transmisión electrónica de mensajes." Una definición más restrictiva que se encuentra con frecuencia ha sido repetida por Robert Bowerman: las LAN "están diseñadas para compartir datos entre estaciones de trabajo uniusuario". Una LAN debe ser local en extensión geográfica, aunque el término "local" podría referirse a cualquier cosa, desde una oficina o un

edificio grande hasta una instalación educativa o industrial de múltiples edificios. Un atributo claro de una LAN es la conectividad, la posibilidad de cualquier punto dado (nodo, conexión) de comunicarse con cualquier otro punto. Parte del poder de una LAN es la capacidad de integrar comunicaciones electrónicas multimedia (datos, video, voz, etc).

El Comité 802 de la IEEE (The Institute of Electrical and Electronics Engineers, Inc.) hizo un buen intento para describir las redes de área local en 1982. La caracterización del IEEE de una LAN fue resumida en forma conveniente en las EDP Solutions de DataPro. El intento por definir el significado de LAN se apega muy de cerca al resumen de DataPro, aunque se han hecho algunas modificaciones. Según el Comité 802 del IEEE "Una red de local es un sistema de comunicaciones de datos que permite a un número de dispositivos independientes comunicarse entre sí".

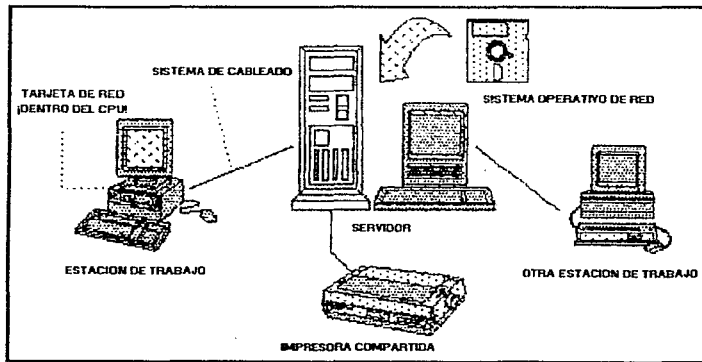


Figura L2.1.3 Elementos de una Red de Comunicaciones(LAN).

CAPITULO I

Una LAN puede clasificarse además como:

- **Intraintitucionales**, de propiedad privada, administradas por el usuario: De esta categoría se excluyen a empresas de servicios comunes, tales como sistemas telefónicos públicos y sistemas comerciales de televisión por cable.
- **Integradas** a través de la interconexión via medio estructural continuo: pueden operar múltiples servicios en un mismo juego de cables. Que soportan comunicaciones de datos a baja y alta velocidad: las LAN no están sujetas a las limitaciones de velocidad impuestas por empresas de servicios comunes tradicionales y pueden ser diseñadas para soportar dispositivos cuya velocidad va de 75 bits/segundo con base en cualquier tecnología a cerca de 140 Mbits/segundo en el caso de LAN de fibras ópticas disponibles en el mercado.
- **Disponibles en el mercado** (al alcance del comprador): el mercado de las LAN sigue siendo volátil y muchos sistemas siguen siendo diseñados por pedido. Incluso los productos ya anunciados pueden encontrarse aún en fase de prueba. Como LAN es más un concepto que un producto, el término "disponibles en el mercado" debe interpretarse de la siguiente manera: los componentes de las LAN que ofrecen conexiones de un medio físico son los que pueden conseguirse en el mercado.

PRINCIPALES ATRIBUTOS DE UNA LAN :

- Algunos estudios afirman que el 80% de los requerimientos de procesamiento en las aplicaciones más comunes se resuelven en un entorno de 70 metros de la ubicación del usuario, y otro 10%, dentro de los 800 metros. Si nos atenemos a estas cifras podremos entender lo siguiente: el 90% de los requerimientos de procesamiento, puede ser resuelto dentro de una LAN. Esto, de por sí, sería una gran ventaja de la utilización de las LAN.
- Es indudable que el poder compartir recursos, trae mayores posibilidades desde el punto de vista de las aplicaciones así como también, disminuye los costos por usuario conectado.
- **Compatibilidad de equipos.** En una LAN que tenga cierta flexibilidad a nivel de las interconexiones, es posible juntar equipo de diferente tecnología, proveedor, aplicación, etc.
- **Procesamiento Distribuido.** La posibilidad de tener unidades redundantes, no depender de un único elemento central, dispone de cierto grado de independencia a nivel de usuario, poder

procesar en el lugar en donde se originan los datos y se toman las decisiones finales, etc. son beneficios que trae consigo la LAN.

- Aplicaciones Complementarias o de Valor Añadido. Las comunicaciones entre terminales, el acceso a las bases de datos y documentación útil, el soporte de correo electrónico, etc. son otros beneficios relacionados al uso de las LAN.
- Ventajas Comparativas con otro tipo de Conexión. Velocidades mayores, menor tasa de error, distancias mayores, transmisión simultánea de información de distinta naturaleza.
- Distribución Física del Hardware. Las LAN permiten optimizar la disposición de los equipos, mejorando la interrelación entre el hombre y la máquina, los requerimientos ambientales, reduciendo costos de instalación, volviendo estéticamente mejores los lugares de trabajo.
- Simplicidad y flexibilidad de modificaciones de configuración. En muchas LAN, las altas y bajas de elementos de la red no afectan al resto de los usuarios ni implican cambios en el software de control.

I.2.1.1. ESTANDARES Y TOPOLOGIAS.

ESTANDARES APLICABLES A LAS REDES.

Los estándares benefician a los compradores de computadoras y productos de comunicaciones, pero obligan a los fabricantes a considerar la calidad en vez de las artimañas. Los debates sobre estándares nunca han sido tomados con entusiasmo por los fabricantes (en especial grandes corporaciones), ya que pierden el control sobre los compradores: como consecuencia de los estándares muchos fabricantes pueden producir equipo compatible y complementario, reduciendo así la posibilidad que una o unas cuantas compañías puedan monopolizar una sección de la industria.

Como muchos fabricantes pueden producir equipo compatible donde existen estándares, el personal de mercadotecnia se ve forzado a resaltar la calidad y los servicios con valor agregado en lugar de diferencias de protocolo específicas. esto genera competencia y precios más bajos para los consumidores de productos.

En particular, el modelo Open System Interconnection (OSI) de la International Organization for Standardization (ISO) ofrece un marco de referencia general para estándares

CAPITULO I

aplicables a las LAN, en tanto que el Comité 802 del IEEE intenta ofrecer estándares que se pueden utilizar para guiar la manufactura de componentes y software para LAN.

NATURALEZA Y CARACTER DE LOS ESTANDARES.

Los estándares no podrán, y no deben, poner fin al debate de acerca de qué es lo "mejor" en relación con alguna tecnología específica. Los comités creadores de estándares están compuestos en general por científicos de la computación de universidades y representantes de fabricantes con intereses especiales concernientes a la tecnología en consideración.

Por tanto, es probable que un borrador final de un estándar sea un compromiso entre diversos intereses especiales, mediado por la necesidad de contar con definiciones técnicas que tengan algún sentido. Los estándares contemporáneos no están diseñados para ser cancelados en piedra. En cambio, están diseñados para ser aumentados y modificados de modo que nuevas tecnologías puedan coexistir con dispositivos anteriores.

Los estándares que nos interesan son aquellos que tienen que ver con medios, acceso y transmisión de datos en redes LAN. Un interés secundario será en aquellos estándares futuros que definan vías de acceso y puentes de enlace. Parte del proyecto 802 de IEEE tiene el fin de intentar estandarizar los dos primeros estratos (físico y de enlace) del modelo OSI. De los varios subestratos (control de enlace lógico, control de acceso a medios), tres métodos de acceso a comunicaciones fueron los primeros respaldados: CSMA/CD, Token Bus y Token Ring.

REDES EN EL CONTEXTO DEL MODELO OSI/ISO

Para entender algunos de los conceptos implicados en la planificación de una red y para dar credibilidad al proyecto mismo, resulta útil dar un vistazo rápido a alguno de los estándares disponibles. Existen varias organizaciones promotoras de estándares en América del Norte y Europa que aspiran a racionalizar sistemas electrónicos. Entre estas organizaciones se encuentran la International Standards Organization (ISO) y The Institute of Electrical and Electronics Engineers (IEEE). Estándares de cualquier tipo para redes de origen reciente, situación que ha conducido a tener una variedad casi caótica de productos para redes.

En 1977 la ISO formó un comité para estudiar la compatibilidad de equipo para redes, trabajo que condujo eventualmente a la publicación del modelo Open System Interconnection. En este contexto, "sistema abierto" se refiere a un modelo de red abierto a equipo de fabricantes de la

competencia. El modelo de referencia OSI es útil para cualquier persona involucrada en la compra o manejo de una red local, porque ofrece un marco teórico, mediante el cual se pueden entender problemas y oportunidades de conexión en redes. El modelo OSI divide los aspectos de conexión en redes en funciones o estratos. Estos estratos se representan en la Tabla I.2.1.1.1

MODELO DE REFERENCIA OSI/ISO OPEN SYSTEM INTERCONNECTION

ESTRATO	FUNCION
Estrato 7 Aplicación	Funciones de usuario final y aplicación final.
Estrato 6 Presentación	Traducción de datos para ser usados en el estrato 7, como conversión de protocolo.
Estrato 5 Sesión	Ofrece el establecimiento de una conexión de sesión entre dos entidades del estrato 6.
Estrato 4 Transporte	Transferencia transparente de datos entre entidades de sesiones.
Estrato 3 Red	Ofrece el medio para establecer, mantener y poner fin a conexiones de redes de sistemas.
Estrato 2 Enlace	Define la estrategia de acceso para compartir el medio físico.
Estrato 1 Físico	Definición de las características eléctricas y mecánicas de la red.

Tabla I.2.1.1.1 Modelo OSI. Este modelo es un intento por definir tareas funcionales en una red, aunque se pueden combinar varios estratos en una misma pieza de equipo o software.

DESCRIPCIÓN DEL MODELO O S I

1. El modelo OSI tiene siete capas. Los principios aplicados para el establecimiento de siete capas fueron los siguientes:

CAPITULO I

2. Una capa se creará en situaciones en donde se necesita un nivel diferente de abstracción.
3. Cada capa deberá efectuar una función bien definida.
4. La función que realizará cada capa deberá seleccionarse con la intención de definir protocolos normalizados internacionalmente.
5. Los límites de las capas deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.
6. El número de capas deberá ser lo suficientemente grande para que las funciones diferentes no tengan que ponerse juntas en la misma capa y, por otra parte, también deberá ser lo suficientemente pequeño para que su arquitectura no llegue a ser difícil de manejar.

CAPA FÍSICA.

La Capa Física se ocupa de la transmisión de bits a lo largo de un canal de comunicación. Su diseño debe asegurar que cuando un extremo envía un bit con valor 1, éste se reciba exactamente como un bit con ese valor en el otro extremo, y no como un bit de valor 0. Preguntas comunes aquí son cuántos volts deberán utilizarse para representar un bit de valor 1 o 0; cuántos microsegundos deberá durar un bit; la posibilidad de realizar transmisiones bidireccionales en forma simultánea; la forma de establecer la conexión inicial y como interrumpirla cuando ambos extremos terminan su comunicación; o bien, cuántas puntas terminales tiene el conector de la red y cual es el uso de cada una de ellas. Los problemas de diseño a considerar aquí son aspectos mecánico, eléctrico, de procedimiento de interfase y el medio de transmisión física. Se puede considerar que el diseño de la capa física cae dentro del dominio del ingeniero en electrónica o en computación.

CAPA DE ENLACE.

La tarea primordial de la Capa de Enlace consiste en, a través de un medio de transmisión común y corriente, transformarlo en una línea sin errores de transmisión para la capa de red. Esta tarea la realiza al hacer que el emisor trocee la entrada de datos en tramas de datos (típicamente constituidas por algunos cientos de bytes), y las transmita en forma secuencial y procese las tramas de asentamientos, devueltas por el receptor. Como la capa física básicamente acepta y transmite un flujo de bits sin tener en cuenta su significado o estructura, recae sobre la capa de enlace la creación o reconocimiento de los límites de la trama. Esto puede llevarse a cabo

mediante la inclusión de un patrón de bit especial al inicio y al término de la trama. Si estos patrones de bits pueden aparecer entre los datos, deberá tenerse un cuidado especial para evitar cualquier confusión al respecto.

La trama puede destruirse por completo debido a una ráfaga de ruido en la línea, en cuyo caso el software de la capa de enlace, perteneciente a la máquina emisora, deberá retransmitir la trama. Sin embargo, múltiples transmisiones de la misma trama introducen la posibilidad de duplicar la misma. Por ejemplo, el duplicado de una trama podrá enviarse, si el acuse de recibo que regresa el receptor se hubiera destruido. Corresponde a esta capa resolver los problemas causados por daño, pérdida o duplicidad de tramas. La capa de enlace ofrece diferentes clases de servicios a la capa de red, cada uno de ellos con distinta calidad y precio.

Otro de los problemas que aparecen en la capa de enlace es el referente a cómo evitar que un transmisor muy rápido sature con datos a un receptor muy lento. Se deberá emplear un mecanismo de regulación de tráfico que permita que el transmisor conozca el espacio de memoria que en ese momento tiene el receptor. Frecuentemente, y por conveniencia, los procedimientos de regulación de flujo y control de errores se tratan en forma conjunta.

Otra dificultad aparece cuando la línea tiene la capacidad de utilizarse para transmitir datos bidireccionalmente. El problema radica en que los asentamientos para el tráfico de A a B compiten por el uso de la línea con las tramas de datos del tráfico que va de B hacia A. Para resolver éste problema se ha inventado una solución inteligente consistente en el envío de aquellos superpuntos (piggy backing).

CAPA DE RED.

La Capa de Red se ocupa del control de la operación de la subred. Un punto de suma importancia en su diseño, es la determinación sobre cómo encaminar los paquetes del origen al destino. Las rutas podrán basarse en tablas estáticas que se encuentran "cableadas" en la red y que difícilmente podrán cambiarse. También, podrán determinarse al inicio de cada conversación, por ejemplo en una sesión de terminal.

Por último, podrán ser de tipo dinámico, determinándose en forma diferente para cada paquete, reflejando la carga real de la red.

CAPITULO I

Si en un momento dado hay demasiados paquetes presentes en la subred, ellos mismos se obstruirán mutuamente y darán lugar a un cuello de botella. El control de tal congestión dependerá también de la capa de red.

Como los operadores de la subred esperan alguna remuneración al esfuerzo que realizan, en muchas ocasiones se introduce una función de contabilidad en la capa de red. El software debe saber, por lo menos, cuántos paquetes o caracteres o bits se enviaron a cada cliente, con objeto de producir información de facturación. Cuando un paquete cruza una frontera nacional, con precios distintos en cada lado, el cálculo de la cuenta puede llegar a complicarse.

También pueden surgir otros problemas cuando un paquete tenga que desplazarse de una red a otra para llegar a su destino. El direccionamiento utilizado en la segunda red puede ser diferente al empleado en la primera. La segunda podrá no aceptar el paquete en su totalidad, por ser demasiado grande. Los protocolos podrán ser diferentes, etc. La responsabilidad, para resolver problemas de interconexión de redes heterogéneas recaerá, en todo caso, en la capa de red.

CAPA DE TRANSPORTE.

La función principal de la Capa de Transporte consiste en aceptar los datos de la capa de sesión, dividirlos, siempre que sea necesario, en unidades más pequeñas, pasarlos a la capa de red y asegurar que todos ellos lleguen correctamente al otro extremo. Además, todo este trabajo se debe hacer de manera eficiente, de tal forma que aisle la capa de sesión de los cambios inevitables a los que está sujeta la tecnología del hardware.

Bajo condiciones normales, la capa de transporte crea una conexión de red distinta para cada conexión de transporte solicitada por la capa de sesión. Si la conexión de transporte necesita un gran caudal, ésta podría crear múltiples conexiones de la red con objeto de mejorar dicho caudal. Por otra parte, si la creación o mantenimiento de la conexión de una red resulta costoso, la capa de transporte podrá multiplexar varias conexiones de transporte sobre la misma conexión de red para reducir dicho costo. En todos los casos, la capa de transporte se necesita para hacer los trabajos de multiplexión transparente a la capa de sesión.

La capa de transporte determina que tipo de servicio debe dar a la capa de sesión, y en último término a los usuarios de la red. El tipo más popular de conexión de transporte corresponde al canal punto a punto sin error, por medio del cual se entregan los mensajes en el mismo orden en que fueron enviados. Sin embargo, el transporte de mensajes aislados sin

garantizar el orden de distribución y la difusión de mensajes a destinos múltiples es otra posibilidad del servicio de transporte. El tipo de servicio se determina cuando se establece la conexión. La capa de transporte es una capa de tipo origen/destino o extremo a extremo. Es decir, un programa en la máquina origen lleva una conversación con un programa parecido que se encuentra en la máquina destino, utilizando las cabeceras de los mensajes y los mensajes de control. Los protocolos, de las capas inferiores, son entre cada máquina y su vecino inmediato, y no entre las máquinas origen y destino, las cuales podrán estar separadas por muchos Procesadores de Intercambio de Mensajes (IMP).

Algunas terminales son multiproceso, lo cual implica que múltiples conexiones estarán entrando y saliendo de cada uno de ellos. Se necesitará alguna forma para decir qué mensaje pertenece a qué conexión. La cabecera de transporte es el lugar en donde puede colocarse esta información.

Además de multiplexar varios flujos de mensaje en un canal, la capa de transporte debe ocuparse del establecimiento y liberación de conexiones a través de la red. Esto requiere algún mecanismo de denominación, de tal forma que un proceso en una máquina tenga una manera de describir con quién desea conversar. También debe haber un mecanismo para regular el flujo de información, de manera que una terminal muy rápida no pueda desbordar a otra más lenta. El control de flujo entre terminales es diferente a aquél entre IMP.

CAPA DE SESIÓN.

La Capa de Sesión permite que los usuarios de diferentes máquinas puedan establecer sesiones entre ellos. A través de una sesión se puede llevar a cabo un transporte de datos ordinario, tal y como lo hace la capa de transporte, pero mejorando los servicios que ésta proporciona y que se utilizan en algunas aplicaciones. Una sesión podrá permitir al usuario acceder a un sistema de tiempo compartido a distancia, o transferir un archivo entre dos máquinas.

Uno de los servicios de la capa de sesión consiste en gestionar el control de diálogo. Las sesiones permiten que el tráfico vaya en ambas direcciones al mismo tiempo, o bien en una sola dirección en un instante dado. Si el tráfico sólo puede ir en una dirección en un momento dado (en forma análoga a un solo sentido en una vía de ferrocarril), la capa de sesión ayudará en el seguimiento de quien tiene el turno.

CAPITULO I

La administración del testigo es otro de los servicios relacionados con la capa de sesión. Para el caso de algunos protocolos resulta esencial que ambos lados no traten de realizar la misma operación en el mismo instante. Para manejar estas actividades, la capa de sesión proporciona testigos que pueden ser intercambiados. Solamente el extremo con el testigo puede realizar la operación crítica.

Otro de los servicios de la capa de sesión es la sincronización. Considérense, por ejemplo, los problemas que podrán ocurrir cuando se trate de hacer una transferencia de archivo de dos horas entre máquinas en una red con un tiempo medio de una hora entre caídas. Después de abortar cada archivo, la transferencia completa tendrá que iniciarse de nuevo y, probablemente, se encontrará de nuevo con la siguiente caída de la red. Para eliminar este problema, la capa de sesión proporciona una forma para insertar puntos de verificación en el flujo de los datos, con objeto de que después de cada caída, solamente tengan que repetirse los datos que se encuentren después del último punto de verificación.

CAPA DE PRESENTACIÓN.

La Capa de Presentación realiza ciertas funciones que se necesitan bastante a menudo para buscar una solución general para ellas, más que dejar que cada uno de los usuarios resuelva los problemas. En particular y, a diferencia de las capas inferiores, que únicamente están interesadas en el movimiento fiable de bits de un lugar a otro, la capa de presentación se ocupa de los aspectos de sintaxis y semántica de la información que se transmite.

Un ejemplo típico de servicio de la capa de presentación es el relacionado con la codificación de datos conforme a lo acordado previamente. La mayor parte de los programas de usuario no intercambian ristas de bits binarios aleatorios, sino, más bien, cosas como nombres de personas, datos, cantidades de dinero y facturas. Estos artículos están representados por ristas de caracteres, números enteros, números de punto flotante, así como por estructuras de datos constituidas por varios elementos más sencillos. Las computadoras pueden tener diferentes códigos para representar ristas de caracteres (por ejemplo, ASCII y EBCDIC), enteros (por ejemplo, complemento a uno o complemento a dos), etc. Para facilitar la comunicación de computadoras con diferentes representaciones, la estructura de los datos que se va a intercambiar puede definirse en forma abstracta, junto con una norma de codificación que se utilice "en el cable". El trabajo de manejar estas estructuras de datos abstractas y la conversión de la

representación utilizada en el interior de la computadora a la representación normal de la red, se lleva a cabo a través de la capa de presentación. La capa de presentación está relacionada también con otros aspectos de representación de la información. Por ejemplo, la comprensión de datos se puede utilizar aquí para reducir el número de bits que tienen que transmitirse, y el concepto de criptografía se necesita usar frecuentemente por razones de privacidad y de autenticación.

CAPA DE APLICACIÓN.

La Capa de Aplicación contiene una variedad de protocolos que se necesitan frecuentemente. Por ejemplo, hay centenares de tipos de terminales incompatibles en el mundo. Considérese la situación de un editor orientado a pantalla que desea trabajar en una red con diferentes tipos de terminales, cada uno de ellos con distintas formas de distribuciones en pantalla, de secuencias de escape para insertar y borrar texto, de movimiento de cursor, etc.

Una forma de resolver este problema consiste en definir una terminal virtual de red abstracta, con el que los editores y otros programas pueden ser escritos y tratar con él.

Con objeto de transferir funciones de terminal virtual de una red a una terminal real, se debe escribir un software que permita el manejo de cada tipo de terminal. Por ejemplo, cuando el editor mueve el cursor de la terminal virtual al extremo superior izquierdo de la pantalla, dicho software deberá emitir la secuencia de comandos apropiados para que la terminal real ubique también el cursor en el sitio indicado. El software completo de la terminal virtual se encuentra en la capa de aplicación.

Otra función de la capa de aplicación es la transferencia de archivos. Distintos sistemas de archivo tienen diferentes convenciones para denominar un archivo, así como diferentes formas para representar las líneas de texto, etc. La transferencia de archivos entre dos sistemas diferentes requiere de la resolución de éstas y de otras incompatibilidades. Este trabajo, así como el correo electrónico, la entrada de trabajo a distancia, el servicio de directorio y otros servicios de propósito general y específico, también corresponden a la capa de aplicación.

TOPOLOGÍA.

Es la forma en que están conectados entre sí los componentes de la red. Dicho de otra manera, este término se refiere a la distribución física de las estaciones de trabajo, servidores,

CAPITULO I

impresoras y demás equipo que pertenece a la red. A la hora de establecer la topología de una red, el diseñador ha de plantearse tres objetivos principales:

- Proporcionar la máxima fiabilidad posible, para garantizar la recepción correcta de todo el tráfico (encaminamiento alternativo).
- Encaminar el tráfico entre el transmisor y el receptor a través del camino más económico dentro de la red.
- Proporcionar al usuario final un tiempo de respuesta óptimo y un caudal eficaz máximo.

TOPOLOGÍA JERÁRQUICA.

La estructura jerárquica es una de las más extendidas en la actualidad. El software que controla la red es relativamente simple, y la topología proporciona un punto de concentración de las tareas de control y de resolución de errores. En la mayoría de los casos, los nodos, HOSTS o estaciones de trabajo situados en el nivel más elevado de la jerarquía es el que controla la red. Muchos fabricantes incorporan a esta topología un cierto carácter distribuido, dotando a los nodos subordinados de un control directo sobre los nodos situados en niveles inferiores dentro de la jerarquía, lo cual reduce la carga de trabajo del nodo central.

Las redes con topología jerárquica se conocen también como redes verticales o en árbol.

TOPOLOGÍA HORIZONTAL (BUS).

Esta estructura es frecuente en las redes de área local. Es relativamente fácil de controlar el flujo de tráfico entre los distintos nodos, ya que el bus permite que todas las estaciones de trabajo reciban todas las transmisiones, es decir, una estación puede difundir la información a todas las demás. La principal limitación de una topología horizontal está en el hecho de que suele existir un sólo canal de comunicaciones para todos los dispositivos de la red. En consecuencia, si el canal de comunicaciones falla, toda la red deja de funcionar. Algunos fabricantes proporcionan canales completamente redundantes por si falla el canal principal, y otros ofrecen conmutadores que permiten rodear un nodo en caso de que falle.

TOPOLOGÍA EN ESTRELLA.

La topología en estrella es una de las más empleadas en los sistemas de comunicación de datos. Una de las principales razones de su empleo es histórica. La red en estrella se utilizó a lo largo de los años sesenta y principios de los setenta porque resultaba fácil de controlar; su software no es complicado y su flujo de tráfico es sencillo. Todo el tráfico emana del núcleo de la estrella que es el nodo central, este por lo general es una computadora que posee el control total de los nodos conectados a él. La configuración estrella es, por tanto, una estructura muy similar a la topología jerárquica, aunque su capacidad de procesamiento distribuido es limitada.

TOPOLOGÍA EN ANILLO.

La estructura en anillo es otra configuración bastante extendida. Se llama así por el aspecto circular del flujo de datos. En la mayoría de los casos, los datos fluyen en una sola dirección, y cada estación recibe la señal y la retransmite a la siguiente del anillo. La organización en anillo resulta atractiva porque con ella son bastante raros los embotellamientos, tan frecuentes en los sistemas en estrella o en árbol. Además, la lógica necesaria para poner en marcha una red de éste tipo es relativamente simple. Cada componente sólo ha de llevar a cabo una serie de tareas muy sencillas: aceptar los datos, enviarlos al nodo conectado al anillo o retransmitirlos al próximo componente del mismo. Sin embargo, como todas las redes, la red en anillo tiene algunos defectos. El problema más importante es que todos los componentes del anillo están unidos por un mismo canal. Si falla el canal entre dos nodos, toda la red se interrumpe. Por eso algunos fabricantes han ideado diseños especiales que incluyen canales de seguridad, por si se produce la pérdida de algún canal. Otros fabricantes construyen conmutadores que redirigen los datos automáticamente, saltándose el nodo averiado, hasta el siguiente nodo del anillo, con el fin de evitar que la falla afecte a toda la red.

TOPOLOGÍA EN MALLA.

La topología en malla se ha venido empleando en los últimos años. Lo que la hace atractiva es su relativa inmunidad a los problemas de embotellamiento y averías. Gracias a la multiplicidad de caminos que ofrece a través de los distintos nodos, es posible orientar el tráfico por trayectorias alternativas en caso de que algún nodo esté averiado u ocupado. A pesar de que la

realización de éste método es compleja y cara, muchos usuarios prefieren la fiabilidad de una red en malla a otras alternativas.

VENTAJAS Y DESVENTAJAS DE LAS TOPOLOGÍA.

VENTAJAS :

Topología Jerárquica.

- El software que controla la red es simple
- Proporciona un punto de concentración de tareas y control de errores.
- Reduce la carga de trabajo del nodo central.
- Provee un procesamiento distribuido para el control de la red.

Topología Horizontal (Bus).

- Es fácil de controlar el flujo de tráfico entre los nodos.
- No existen nodos centrales.
- La carga de trabajo tiende a ser más uniforme en todos los nodos.
- La conexión de nuevos nodos es fácil.

Topología en Estrella.

- Es fácil de controlar.
- Su software no es complicado.
- Su flujo de tráfico es sencillo.

Topología en Anillo.

- Evita embotellamientos.
- Evita colisiones.
- La tarea de cada nodo es sencilla.
- La lógica para poner en marcha esta red es sencilla.

Topología en Malla.

- Evita los problemas de embotellamiento.
- Ofrece trayectorias alternativas para solución de fallas.

DESVENTAJAS:

Topología Jerárquica

- Si la red es muy grande su estructura y el tiempo de acceso, se vuelven muy complejos.
- Si falla el nodo central falla la red se bloquea.
- También la falla de los nodos de los niveles más altos puede paralizar el tráfico de la red.

Topología Horizontal (Bus).

- Si falla el canal de comunicación que es, la red se paraliza.
- La comunicación puede ser muy lenta debido a que cada nodo recibe y retransmite el mensaje en redes con muchos nodos.

Topología en Estrella.

- Si el nodo central falla la red deja de funcionar.
- Por su estructura su alcance es limitado.

Topología en Anillo.

- Si algún nodo de la red falla está se paraliza.
- La transmisión de la información puede ser muy lenta.

Topología en Malla.

- Su estructura puede ser muy compleja.
- La inserción de nuevos nodos es algo complicada.

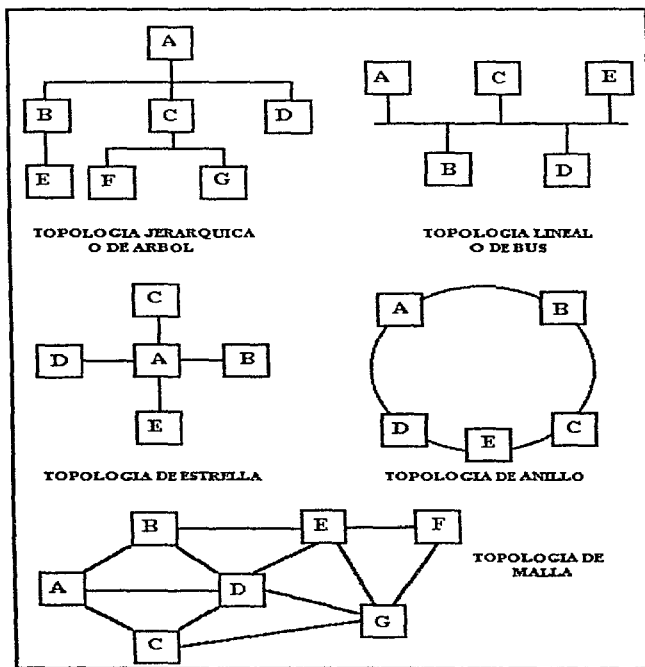


FIGURA. L2.1.1.2. TOPOLOGÍA DE REDES.

1.2.1.2. PROTOCOLOS.

Protocolo es el conjunto de reglas que siguen para empacar la información que va a ser enviada por las estaciones de trabajo y los servidores de la red. Se considera como lenguaje de la red. El protocolo es en realidad un elemento de la programación de las redes. Los protocolos involucran dos factores:

- a) Control de la transmisión.
- b) Acceso al medio.

El control puede estar centralizado o distribuido a través de todos los nodos. El acceso puede manejarse por demanda o siguiendo un orden específico. Estos dos tipos de control y de acceso producen cuatro tipos de protocolos:

POR ESCRUTINIO (POLLING ACCES PROTOCOL).

Este método de acceso se caracteriza por contar con un dispositivo controlador central, que es una computadora inteligente, como un servidor. Pasa lista a cada nodo de la secuencia predefinida solicitando su acceso a la red. si tal solicitud se realiza, el mensaje se transmite, de lo contrario, el dispositivo central se mueve a pasar lista al siguiente nodo.

POR CONMUTACIÓN DE CIRCUITOS (CIRUIT-SWITCHING).

Al igual que en el método anterior el control se encuentra centralizado, pero el acceso se va dando por demanda. Este tipo de protocolo se da en las topologías tipo estrella y jerárquica y un buen ejemplo lo constituye la red telefónica.

La conmutación de circuitos consiste en que el controlador central conecta un nodo que desea comunicación o nodo transmisor con otro llamado receptor. si el nodo receptor se encuentra ocupado, el controlador niega el acceso al transmisor, dedicando ambas lineas para formar un circuito cerrado entre ambos.

POR PASO DE IDENTIFICADOR (TOKEN PASSING).

Este protocolo, que se utiliza en redes ARCNET y TOKEN RING, se basa en un esquema libre de colisiones, dado que la señal o identificador (*token*) se pasa de un nodo o estación de trabajo al siguiente nodo. Con esto se garantiza que todas las estaciones tendrán la misma oportunidad de transmitir y que un solo paquete viajará a la vez en la red.

CAPITULO I

En este método, el acceso a la línea de comunicación está libre para transmitir mensajes, por lo que se pueden tener tiempos de respuesta predecibles aún con gran cantidad de actividad en la red.

Uno de los inconvenientes es que, al llegar a un nodo el *token* regenera el mensaje antes de pasarlo al siguiente. Esto origina una reducción en el rendimiento de la red pero se asegura una transmisión exitosa desde la primera vez que se envía el mensaje. Token Ring opera a una velocidad de transferencia de 4 ó 16 Mbps.

POR DETECCIÓN DE PORTADORA (CSMA/CD).

(Carrier Sense Multiple Acces / Collision Detection, Acceso Múltiple por Detección de Portadora / con Detección de Colisiones). Este tipo de protocolo utiliza una combinación de control distribuido con acceso por demanda. Es el método más común de acceso en las topologías de bus lineal.

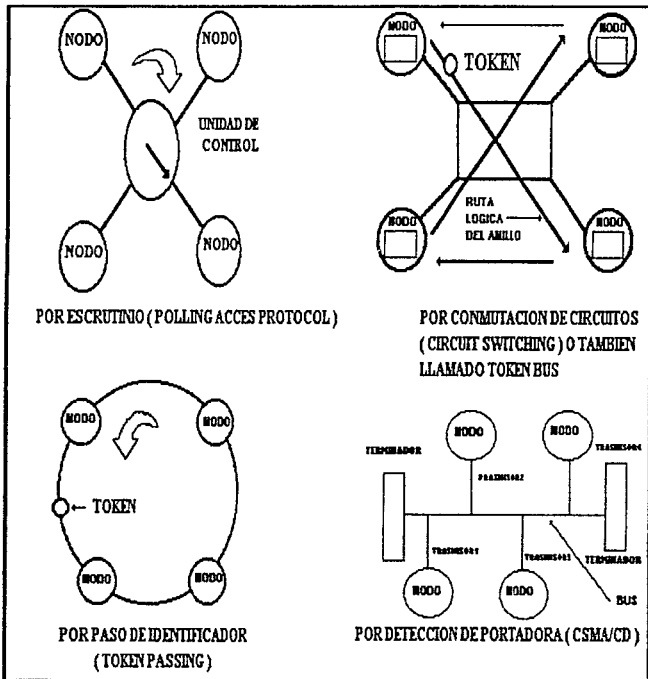
En este método todos los nodos tienen la misma posibilidad de transmitir información a través de la red. Para identificar si un nodo puede transmitir, escucha si existe en ese momento una transmisión en la línea común, de ser así, esperará a que la línea esté silenciosa.

Cuando la línea se encuentra silenciosa un nodo transmite su mensaje, si dos nodos detectan simultáneamente la línea silenciosa y comienzan a transmitir simultáneamente, se producirá una colisión, que distorsionará los mensajes, por ello cada nodo escucha la transmisión que realizó y al ver que esta fue alterada detecta inmediatamente que ocurrió una colisión.

Cuando ocurre una colisión los dos nodos que la provocaron lo detectan y entonces generan un tiempo de espera aleatorio, para iniciar nuevamente la transmisión, al ser estos tiempos aleatorios resulta poco probable una nueva colisión entre ellos.

Entre los ejemplos más importantes de protocolos de red se encuentran los siguientes:

- TCP/IP (TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL)
- NetBIOS (NETWORK BASIC INPUT-OUTPUT SYSTEM)
- IPX/SPX (INTERNET PACKET EXCHANGE/SEQUENCED PACKET EXCHANGE)
- SDLC (SYNCHRONOUS DATA LINK CONTROL)
- DECnet (DIGITAL EQUIPMENT CORPORATION NETWORK)



POR ESCRUTINIO (POLLING ACCES PROTOCOL)

POR CONMUTACION DE CIRCUITOS
(CIRCUIT SWITCHING) O TAMBIEN
LLAMADO TOKEN BUS

POR PASEO DE IDENTIFICADOR
(TOKEN PASSING)

POR DETECCION DE PORTADORA (CSMA/CD)

FIGURA 1.2.1.2.1 TIPOS DE PROTOCOLOS DE ACCESO A REDES.

I.2.1.3. INTERCONEXIONES.

Tarjeta (o adaptador) de red: Tanto las estaciones de trabajo como los servidores se encuentran conectados entre sí mediante cables especiales. Cada equipo cuenta con una tarjeta o adaptador de red, a la que se conectan dichos cables y está diseñada para controlar las comunicaciones de la estación de trabajo con los demás puntos de la red.

Sistema de Cableado: Dentro de éste, se incluyen los cables y elementos adicionales asociados a ellos como cajas de conexiones y conectores especiales, que se utilizan en la interconexión de los puntos de la red. Los tipos de cable más utilizados son básicamente tres y se describen a continuación:

PAR TRENZADO O TWISTED PAIR.

El medio de transmisión más antiguo, y todavía el más ampliamente utilizado, es el PAR TRENZADO. Este consiste en dos alambres de cobre aislados, en general de 1 mm de espesor. Los alambres se entrelazan en forma helicoidal. La forma trenzada del cable se utiliza para reducir la interferencia eléctrica con respecto a los pares cercanos que se encuentran a su alrededor.

Los pares trenzados se pueden utilizar tanto para la transmisión analógica como digital, y su ancho de banda depende del calibre del alambre y de la distancia que recorre; en muchos casos pueden obtenerse transmisiones de varios megabits/s, en distancias de pocos Km. Debido a su adecuado comportamiento y bajo costo, los pares trenzados se utilizan ampliamente.

CABLE COAXIAL.

El cable coaxial es otro medio típico de transmisión. Hay dos tipos de cable coaxial que se utilizan con frecuencia, uno de ellos es el cable de 50 ohms que se utiliza en la transmisión digital, en tanto que el otro tipo, el cable de 75 ohms, que se emplea en la transmisión analógica (por ejemplo, en el servicio de televisión por cable).

El cable coaxial consta de un alambre de cobre duro en su parte central, es decir, que constituye el núcleo, el cual se encuentra rodeado por un material aislante. Este material aislante está rodeado por un conductor cilíndrico que frecuentemente se presenta como una malla de tejido trenzado. El conductor externo está cubierto por una capa de plástico protector.

FIBRAS ÓPTICAS.

El cable hecho de fibra de vidrio o plástico, a través del cual se emite un haz luminoso que se va reflejando dentro del cable debido a los diferentes índices de refracción entre éste y su cubierta. Su instalación es considerablemente más costosa que en los casos anteriores, ya que requiere de cuidados y mantenimiento especial, aunque en condiciones adecuadas permite velocidades de transmisión mas elevadas y a mayores distancias. Se espera que con los avances técnicos se convierta en el cable del futuro.

En la figura 1.2.1.3.1 se muestran algunas características de las líneas de transmisión comentadas en párrafos anteriores.

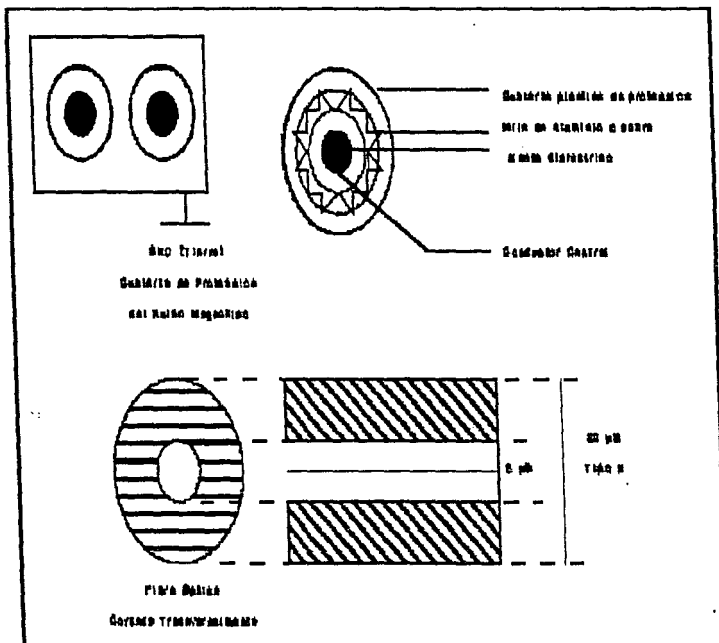


FIG. I.2.1.3.1 TIPOS DE LÍNEAS DE TRANSMISIÓN DE REDES .

FORMAS DE CONEXIÓN DE TERMINALES O NODOS

Tradicionalmente se hace referencia a dos formas de conectar un sistema central con una o varias terminales o sistemas secundarios: Punto a Punto y Multipunto.

PUNTO A PUNTO

Tal es el caso, cuando un enlace físico une sólo dos puntas de transmisión de datos, desde donde, por lo general, tanto se pueden enviar como recibir.

MULTIPUNTO

Se utiliza el nombre de multipunto cuando se hace referencia a un sistema central que conecta varias terminales o sistemas secundarios.

VENTAJAS Y DESVENTAJAS.

Ventajas de la conexión de Punto a Punto.

- Tiene bajo costo.
- Permite forma "conversacional" de comunicaciones.
- Permite fácil migración a fibras ópticas.
- Admite la utilización de diferentes medios físicos.
- Es de fácil implementación (en general es simple).

Ventajas Multipunto.

- Permite, normalmente, la conexión de más terminales por cada procesador central.
- Un mejor aprovechamiento de los recursos de los nodos con este tipo de conexión.
- Permite una mayor portabilidad de las aplicaciones.

Desventajas de Punto a Punto.

- No se aprovechan los recursos de los nodos al máximo.
- La transmisión de la información tiene solo una línea por lo cual se puede perder si falla.

Desventajas de Multipunto.

- Software y Hardware relativamente complejo.
- Exige la utilización de "intermediario".
- Exige la utilización de sondeo.

- Puede aumentar los tiempos de respuesta frente al punto a punto.

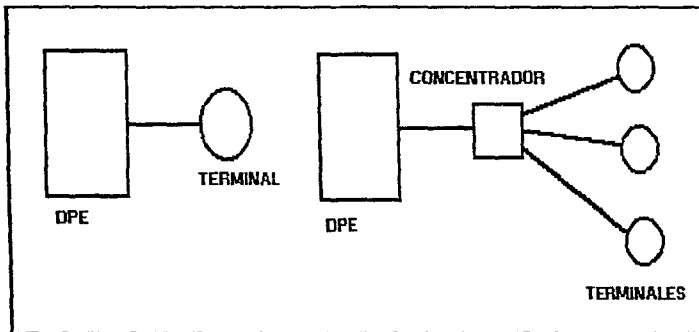


FIG. I.2.1.3 CONEXIONES PUNTO PUNTO Y MULTIPUNTO

CONEXIÓN ENTRE EL MUNDO ANALÓGICO Y DIGITAL. SEÑALES DIGITALES.

Cuando dos ETD (Equipo terminal de datos, estación de trabajo, nodo o host) emplean la línea telefónica para comunicarse entre sí, deben adaptar su señal a las características de un canal de éste tipo, pensado para trabajar con el mundo analógico. Sin embargo, los nodos "hablan" en lenguaje digital. Como se ve en la figura 1.10, el aspecto de una forma de onda digital es muy distinto al de una forma de onda analógica. Se parece en que es continua, se repite a sí misma y tiene carácter periódico, pero es muy diferente ya que ésta es discreta (presenta cambios muy abruptos en su voltaje). Las computadoras y las terminales usan símbolos digitales, binarios, porque los transistores semiconductores actúan como dispositivos discretos de dos estados.

Muchos sistemas actuales emplean transmisiones digitales, que ofrecen diversas ventajas frente a los canales analógicos. en cualquier caso, en las conexiones locales de equipos con los canales de las compañías telefónicas siguen predominando enlaces analógicos.

EL MÓDEM.

Los modems son dispositivos destinados principalmente a la conversión de señales digitales en analógicas y viceversa. Su nombre proviene de la contracción de modulación demodulación.

Pueden ser externos, independientes o residir dentro del gabinete del procesador central. Según el caso se les llama modulares o integrados.

Se distinguen por sincrónicos o asincrónicos, dependiendo del tipo de mensaje a transmitir. Pueden tener diagnósticos residentes y disponer de mecanismos de detección y corrección de errores. La rapidez de reacción de los circuitos del módem, es una variable que juega en los tiempos de respuesta de las terminales remotas.

Cuando es necesario, pueden proveer la sincronización de la señal. También pueden tener mecanismos de discado (dialing) y autorespuesta.

L3 INGENIERÍA DE SOFTWARE.

La falta de planeación es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de programación. Para evitar estos problemas se requiere de una planeación cuidadosa, tanto en el proceso de desarrollo como en el de la operación del producto.

Las fuentes que producen las ideas de productos de programación incluyen las necesidades del cliente generadas externamente, las necesidades internas de la organización, planes de mercadotecnia, y los planes o misiones organizacionales.

El primer paso en la planeación de un proyecto de programación es preparar, en la terminología del cliente, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado definitivo del problema debe de incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

La definición del problema requiere de un entendimiento cabal del dominio del problema y del entorno de éste. Las técnicas para obtener este conocimiento, por parte del planeador, son entrevistas con el cliente, observación de las tareas problemáticas, y desarrollo de las reales.

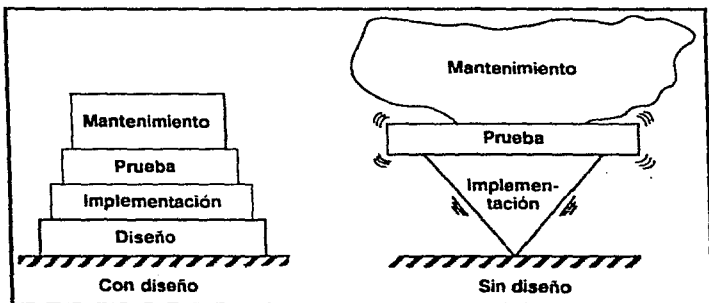


Figura L3.1 Estabilidad del sistema con y sin diseño.

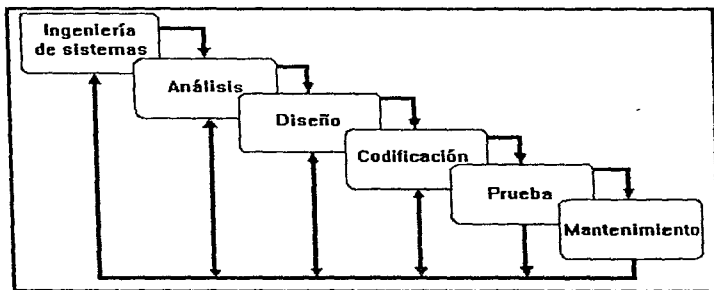


Figura L3.2 Ciclo de vida clásico.

En la figura 1.3.2 se ilustra el paradigma del ciclo de vida clásico para la ingeniería del software. Algunas veces llamado "modelo en cascada", éste ciclo de vida exige un enfoque sistemático, secuencial, del desarrollo del software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

En las figuras 1.3.3, 1.3.4 y 1.3.5 se ilustran los pasos genéricos que se deben realizar y las distintas representaciones del software que se derivan según se pasa del concepto a la realización.

Fase de definición. La fase de definición de la ingeniería del software, mostrada en la figura 1.3.3 empieza con la etapa de planificación del software. El propósito de la etapa de planificación del software es proveer de una indicación preliminar de la viabilidad del proyecto en relación al coste y a las restricciones de tiempo que ya se hayan establecido.

El próximo paso en la fase de definición es el análisis y la definición de los requerimientos del software. En este paso el elemento del sistema asignado al software se define en detalle.

El análisis y definición de los requerimientos del software es un esfuerzo conjunto guiado por el diseñador de software y el cliente. La especificación de requerimientos del software es un documento distribuible producido como resultado de esta etapa.

La fase de definición culmina con una revisión técnica de la especificación de requerimientos del software por el diseñador y el cliente. Una vez que se han definido los requerimientos, el plan de software es reevaluado para su corrección.

Fase de desarrollo. La fase de desarrollo (figura 1.3.4) traduce un conjunto de requerimientos en el elemento del sistema operacional que llamamos software.

El primer paso en la fase de desarrollo se centra en el diseño. Es decir, se desarrolla una estructura modular, se definen interfaces y se establece la estructura de datos.

A continuación se consideran los aspectos procedurales de cada componente modular del diseño de software. Cada descripción procedural detallada es añadida al documento de diseño después de revisarla.

Las tres etapas finales del desarrollo están asociadas a la prueba del software. Las pruebas de unidad intentan validar el rendimiento funcional de un componente individual del software. La prueba de integración constituye un medio de construcción de la arquitectura del software. La prueba de validación comprueba que se han cumplido todos los requerimientos.

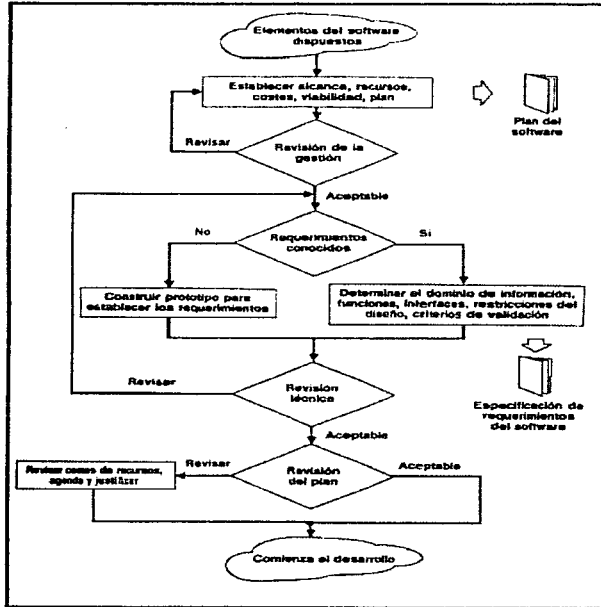


Figura 1.3.3 Fase de definición.

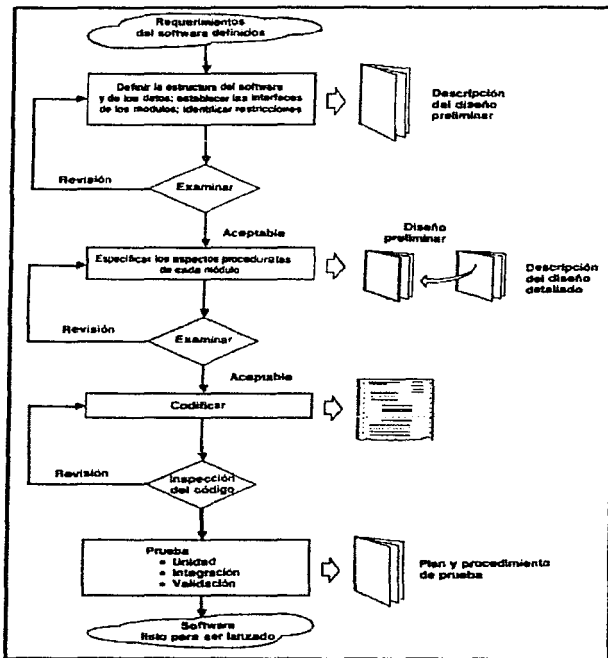


Figura I.3.4 Fase de desarrollo.

CAPÍTULO I

Fase de mantenimiento. El software no trivial se debe mantener. La fase de mantenimiento de la ingeniería del software, ilustrada en la figura I.3.5, comienza antes de la distribución del software y continúa a lo largo de su vida útil. Durante el mantenimiento del software se corrigen errores, se hacen adaptaciones y se implementan mejoras.

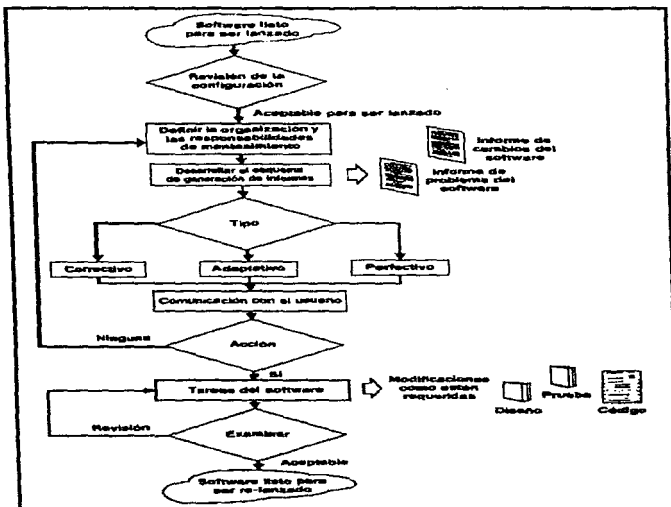


Figura I.3.5 Fase de mantenimiento.

L3.1 DEFINICION DE REQUISITOS.

El objetivo de la definición de requisitos es especificar total y consistentemente los requerimientos técnicos del producto de una manera concisa y sin ambigüedades, y se emplea para ello una definición formal.

La descripción precisa de los requisitos de un sistema de software se denomina documento de los requisitos del software. Este documento, no es un documento de diseño. Debe establecer lo que ha de hacer el sistema sin especificar cómo debe de hacerlo.

En principio, los requisitos establecidos en tal documento deben ser completos y consistentes. Es necesario especificar todo lo que ha de hacer el sistema y ningún requisito debe entrar en conflicto con los demás. Ian Sommerville propone la siguiente definición de requisitos:

- a) *Introducción.* Describir brevemente las necesidades del sistema.
- b) *Hardware.* Describir el hardware que se va a emplear y sus interfaces.
- c) *El modelo conceptual.* Visión de muy alto nivel que muestra los principales servicios del sistema y las relaciones entre ellos.
- d) *Requisitos funcionales.* Servicios proporcionados al usuario.
- e) *Requisitos de la base de datos.* Organización lógica e interrelaciones entre los datos del sistema.
- f) *Requisitos no funcionales.* Restricciones bajo las cuales operará el software.
- g) *Información para el mantenimiento.* Describir las partes más sujetas a cambios.
- h) *Glosario.* Definir los términos técnicos empleados en el documento.
- i) *Índice.* Puede ser deseable proporcionar al documento más de una clase de índice.

Puesto que los requisitos están sujetos a modificaciones, resulta esencial que el documento de los requisitos del software esté organizado de forma que se puedan introducir las modificaciones sin tener que escribir demasiado.

EL MODELO DEL SISTEMA.

Después de efectuar un análisis inicial de las necesidades del usuario, el siguiente paso, que constituye la primera etapa en la definición de requisitos, es producir un modelo conceptual del sistema de software. Este modelo es una visión de muy alto nivel del sistema en la que se identifican los servicios principales al usuario y se documentan sus relaciones.

CAPÍTULO I

Las notaciones más efectivas para describir un modelo conceptual de un sistema son las notaciones gráficas. La razón es que las imágenes y diagramas suelen ser comprensibles para los usuarios que no tienen conocimiento de la ingeniería de software.

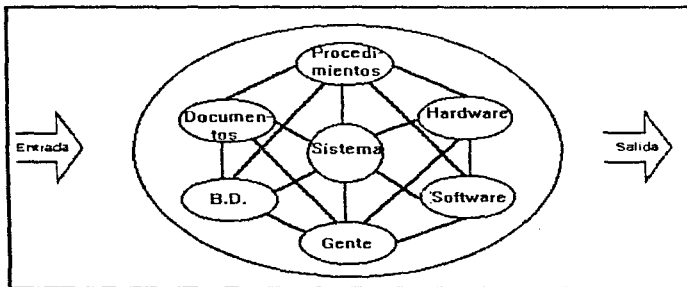


Figura I.3.1.1 Los elementos del sistema.

Los diagramas de flujo de datos especifican las fuentes y destino de los datos, sus almacenamientos, transformaciones, y los flujos entre ellos. Un almacén de datos es una estructura de datos conceptual, en el sentido de que no se consideran detalles de la instrumentación; sólo se hace incapié en las características lógicas de los datos.

El siguiente paso en el proceso de modelado conceptual es tomar cada servicio principal del usuario y establecer un modelo conceptual para ese servicio y, si es necesario, esos modelos se pueden refinar con más detalle.

Como diagramas de flujos, los de datos se pueden emplear a cualquier nivel de detalle, pueden ser representados según su jerarquía por medio de la especificación de la operación interna de los nodos funcionales, y para ello se utilizan diagramas adicionales.

La definición de un elemento del diccionario de datos incluye el nombre del dato, y atributos tales como el diagrama de flujo de datos donde se usa, su propósito, procedencia, subunidades, y cualquier nota aclaratoria necesaria. Cualquier nombre de datos que aparezca en el diagrama de flujo de datos, debe aparecer en el diccionario de datos.

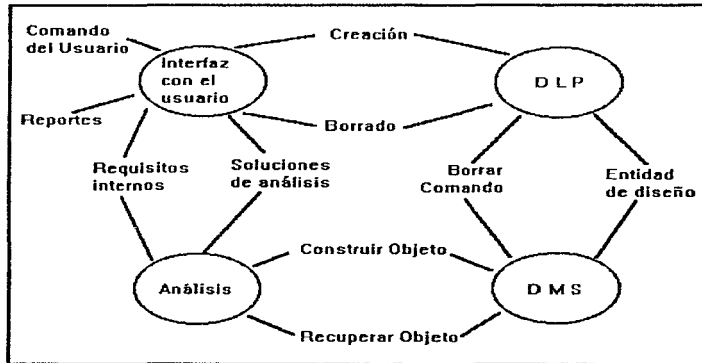


Figura I.3.1.2 Diagrama de flujo de datos

La razón de establecer un modelo conceptual del sistema es proporcionar un esquema de trabajo en el que se pueda basar una definición de requisitos más completa y detallada.

DEFINICION DE REQUISITOS FUNCIONALES.

La especificación de las características funcionales de un producto de programación es una de las actividades más importantes que se debe llevar a cabo en el análisis de requisitos.

Los requisitos funcionales del sistema son aquellos servicios que el usuario espera del sistema. Al inicio, los principios funcionales de un sistema deben ser completos y consistentes. Se entiende que todos los servicios requeridos por el usuario deben especificarse, y la consistencia significa que ninguna definición de requisitos debe contradecir a otra.

Existen distintas maneras de expresar los requisitos funcionales de un sistema. De todas ellas la más utilizada es el lenguaje natural, por la simple razón de que es el más expresivo y porque puede ser comprendido tanto por los usuarios como por los desarrolladores del sistema.

CAPITULO I

Hay dos problemas principales que pueden surgir cuando se utiliza el lenguaje natural para la definición de requisitos:

- a) No se distinguen con claridad los requisitos funcionales, los no funcionales y los objetivos del sistema.
- b) Cada párrafo puede incluir varios requisitos individuales en una sola proposición, lo que hace muy difícil la revisión de la consistencia y de la plenitud.

Podría parecer que el lenguaje natural no es una notación que ayude a expresar los requisitos del software. Sin embargo, es necesario utilizar este lenguaje cuando se formulan requisitos de alto nivel, porque la generalidad de estos no puede expresarse en una notación más restringida. Además, se puede emplear como suplemento y para documentar una definición más detallada de requisitos.

Las notaciones formales tienen la ventaja de ser concisas y no ambiguas, apoyan el razonamiento formal de las especificaciones funcionales y dan una base para la verificación de los resultados del producto de programación.

Tanto las notaciones orientadas a estados como las relacionales se utilizan para especificar las características funcionales del producto. Las notaciones relacionales se basan en los conceptos de entidades y atributos. A las entidades se les llama elementos de un sistema; los nombres se escogen de tal manera que indiquen la naturaleza de los elementos. Los atributos se especifican al aplicar funciones y relaciones a las entidades nombradas. Los atributos especifican las operaciones permitidas con las entidades, las relaciones entre ellas, y el flujo de datos entre las entidades.

El estado de un sistema es la información requerida para resumir el estado de las entidades del sistema en un momento dado; al darse el estado actual y un estímulo, se puede determinar el siguiente estado.

REQUERIMIENTOS DE LAS BASES DE DATOS.

Es esencial la identificación de las funciones e interfaces; se requiere de la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos.

La arquitectura lógica de una base de datos se define mediante un esquema que representa las definiciones de las relaciones entre las entidades de información. La arquitectura física de una base de datos depende de la configuración del hardware residente.

Antes de que pueda comenzar la evaluación de los requerimientos de una base de datos, el analista debe comprender los objetivos y ámbitos globales del sistema para el que se va a desarrollar la base de datos. Posteriormente se desarrolla un modelo de información completo y con gran detalle.

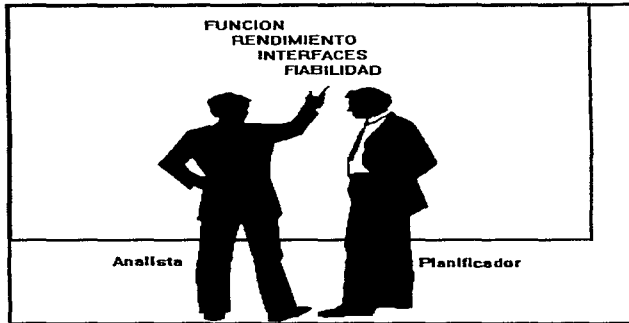


Figura I.3.1.3 Función del analista.

El modelo de información incluye un diccionario de datos que define todos los elementos de datos en términos de la información que se usa para desarrollar el elemento.

El siguiente paso en el análisis de una base de datos es definir las características lógicas y físicas de la base de datos. La organización lógica debe considerar los requerimientos de acceso, modificación, asociatividad de los datos, así como otros aspectos orientados al sistema. La organización física de una base de datos define la estructura de archivos, los formatos de los registros, las características de procesamientos dependientes del hardware y las características de DBMS.

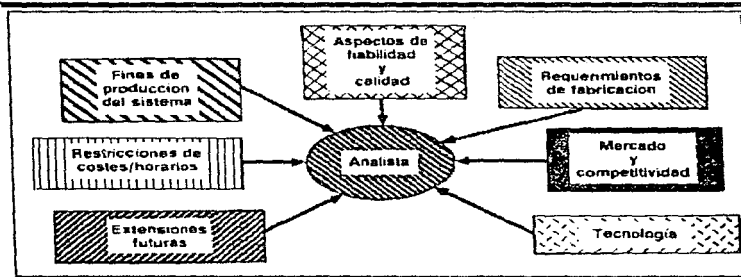


Figura I.3.1.4 Información que recibe el analista.

DEFINICION DE REQUISITOS NO FUNCIONALES.

Un requisito no funcional de un sistema es una restricción u obligación impuesta al servicio de éste. Aunque tanto los requisitos funcionales como los no funcionales están sujetos a cambios, los requisitos no funcionales se ven especialmente afectados por los cambios en la tecnología de hardware.

Los requisitos no funcionales dependientes del hardware pueden especificarse de modo que presuponga las capacidades de hardware que existirán a la terminación del proyecto, aunque quizá no sea ese el caso al comienzo del proyecto. Estos requisitos tienden a estar en conflicto y actuar recíprocamente con otros requisitos funcionales del sistema.

CONFIRMACION DE REQUISITOS.

Una vez establecidos, los requisitos del sistema han de confirmarse. Si no se realiza ninguna confirmación, los errores cometidos en la definición de requisitos se propagarán al diseño y a la aplicación del sistema, y puede ser necesario realizar costosas modificaciones en el sistema para corregir esos errores.

Las revisiones son la manera más efectiva de confirmar los requisitos. Durante la revisión, los usuarios y los desarrolladores del sistema estudian y examinan los requisitos, y se buscan anomalías e inconsistencias.

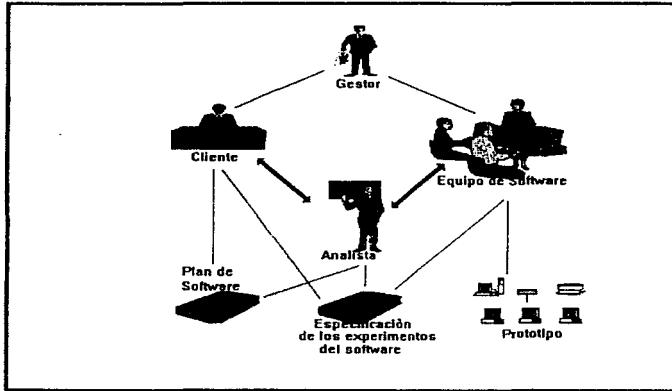


Figura L3.1.5 Interacción del analista.

El realismo de los requisitos puede demostrarse en algunos casos construyendo un simulador del sistema. La técnica de simulación de sistemas resulta especialmente útil para demostrar que se pueden cumplir los requisitos no funcionales.

Los otros pasos implicados en la confirmación de requisitos, la comprobación de la plenitud de los requisitos y la demostración de que el sistema cumple con las necesidades reales del usuario, sólo se pueden lograr con la cooperación del usuario del sistema, que debe examinar y comprender los requisitos y comprobar si especifican la clase de sistema que se desea.

1.3.2 ESPECIFICACION DEL SOFTWARE.

Durante esta etapa, se analiza la definición de requisitos y se diseñan los componentes del software para proporcionar los servicios al usuario. Este diseño se expresa de manera que a continuación se puedan realizar estos componentes en un lenguaje de programación.

Un programa que funciona es sólo una parte de una configuración del software que incluye todos los elementos mostrados en la figura 1.3 2.1. La documentación da la base para un buen desarrollo y lo que es más importante, provee de guías para la tarea de mantenimiento del software.

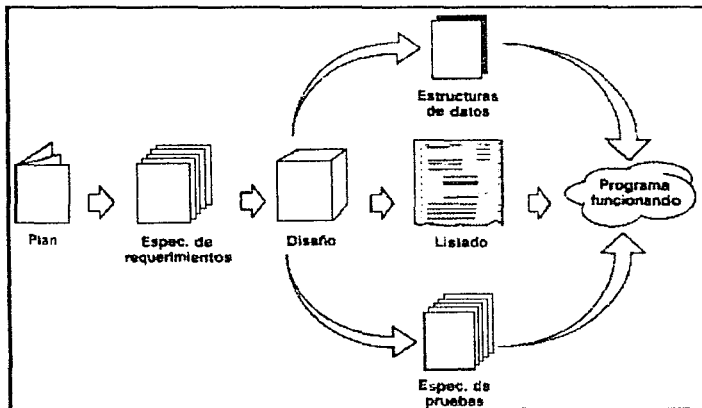


Figura 1.3.2.1 Configuración del software.

La especificación, independientemente del modo en que se realice, puede ser vista como un proceso de representación. Los requerimientos se representan de forma que conduzcan finalmente a una correcta implementación del software. La especificación del software está destinada al diseñador más que al usuario. Está compuesta de definiciones abstractas de los componentes del software, no de servicios al usuario.

La especificación formal ha de expresarse en una notación matemáticamente sólida. Esto implica que tanto la sintaxis como la semántica del lenguaje de especificación han de ser definidas de manera formal.

Son varias las ventajas derivadas del empleo de especificaciones formales de los componentes del software:

- a) Dada una especificación formal y una definición semántica del lenguaje de programación, puede demostrarse que un programa cumple con su especificación.
- b) Las especificaciones formales del software pueden estudiarse matemáticamente, para tener la posibilidad de contestar, por ejemplo, a las preguntas relacionadas con la equivalencia de especificaciones.
- c) Las especificaciones formales son procesables por computador y se pueden construir instrumentos de software para auxiliar a los diseñadores en el desarrollo, comprensión y depuración de las especificaciones del software.

L3.2.1 ESPECIFICACIONES DE INTERFACES.

La especificación de interfaces implica especificar las restricciones de entrada y salida que definen la función del componente de software.

En la práctica, los componentes de software suelen tener que utilizar entradas incorrectas, por lo que es necesario tener algún mecanismo para especificar la salida correspondiente a las entradas correctas e incorrectas.

Esto puede lograrse considerando el estado del componente al terminar. Si se proporcionó una entrada válida, se dice que el componente terminó en un estado normal (**N-estado**). Por otro lado, si la entrada es inválida, el componente termina en un estado de excepción (**E-estado**). Debe haber un E-estado por cada clase de entrada inválida.

La utilización de conjuntos como objeto fundamental en las especificaciones es muy conveniente, pues sus propiedades están estudiadas, y sus operaciones definidas. No obstante, con frecuencia es improbable considerar toda la información contenida en conjuntos, por lo que se deben incluir otros tipos de especificación.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

L3.2.2. ESPECIFICACIONES OPERACIONALES.

Cuando se utilizan especificaciones de interface, no se describe el cálculo que realiza la transformación de la entrada en salida. Por el contrario, especificación operacional define explícitamente esta transformación de entrada y salida, aunque pueda expresarse de una manera abstracta de alto nivel.

La función principal de las especificaciones operacionales sigue siendo la de definir lo que el programa debe hacer en lugar de efectuar realmente las operaciones.

Un área muy importante del desarrollo del software, que parece ignorarse en el desarrollo de la mayoría de los análisis sobre especificación del software, es la de la entrada y la salida. Los programas reales toman y producen información para su ambiente. Si una parte de cierto sistema es responsable de la entrada y la salida, también se debe especificar. La razón de evitar las operaciones de entrada y salida en la mayoría de los análisis sobre especificación formal es que resulta muy difícil establecer un modelo limpio de propósito general subyacente en los procesos de entrada y salida.

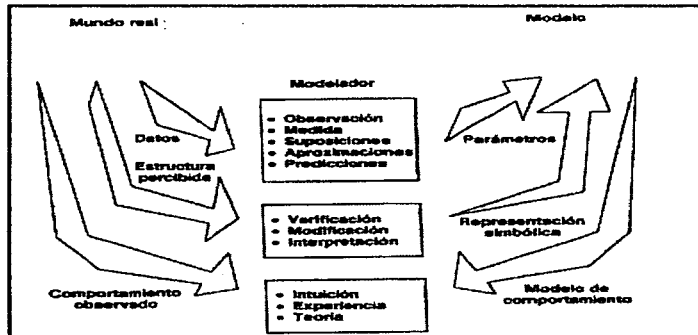


Figura L3.2.2.1 Modelización.

1.3.2.3 LA ESPECIFICACION DE ABSTRACCIONES DE DATOS.

La abstracción es la herramienta intelectual que permite interactuar con los conceptos independientemente de las instancias particulares de estos; durante la definición de los requerimientos y el diseño, la abstracción permite la separación de los aspectos conceptuales de un sistema de los que será más tarde instrumentado.

Durante el diseño de la programación, la abstracción permite organizar y dirigir los procesos del pensamiento al posponer las consideraciones estructurales y algorítmicas hasta que las características funcionales, las cadenas de datos y los almacenamientos hayan quedado definidos, las consideraciones estructurales son entonces estudiadas antes de ver los aspectos de la instrumentación. Este enfoque reduce la complejidad a la que nos enfrentamos en cualquier momento durante el diseño.

Cuando se representa un tipo de datos como un conjunto de operaciones y valores, se denomina tipo de datos abstracto. Los tipos de datos abstractos lo son en el sentido de que los detalles de la representación del dato y la forma de instrumentar a las funciones que los manipulan, quedan escondidos dentro del grupo que instrumenta dicho tipo de dato.

La abstracción de datos facilita al diseñador representar un objeto de datos a diferentes niveles de detalle, y especificar un objeto de datos en el contexto de las operaciones que se le puedan aplicar. El uso de los tipos de datos abstractos permite construir especificaciones estructuradas y también parece conducir a una metodología de aplicación que da como resultado un software más fácil de mantener.

Los detalles internos de la representación y manipulación de datos puede modificarse a voluntad, suponiendo que las interfaces de los procedimientos de manipulación permanezcan iguales, los otros componentes del programa no serán afectados por la modificación. Al utilizar una facilidad de abstracción de datos, las entidades de datos pueden definirse en términos de tipos predefinidos, tipos definidos por el usuario y otras abstracciones de datos, permitiendo así el desarrollo sistemático de abstracciones jerárquicas.

Un tipo abstracto puede ser definido en términos de otros tipos abstractos; con esto, se permite el diseño jerárquico de datos. Por supuesto, se deben definir los datos primitivos para evitar un ciclo infinito en la definición.

L3.2.4 ESPECIFICACIONES PRACTICAS DE SOFTWARE.

De acuerdo con Ian Sommerville, la especificación del sistema puede considerarse en tres etapas:

1. Construcción de las especificaciones operacionales para las abstracciones de más alto nivel del sistema, expresándolas en función de abstracciones más simples y tipos de datos abstractos.
2. Construcción de las especificaciones para los tipos de datos abstractos del sistema.
3. Construcción de especificaciones de interfaces para las abstracciones más simples del sistema que pueden especificarse con facilidad de esta manera.

El desarrollo de las especificaciones es un proceso iterativo; a medida que se desarrollan las especificaciones de las abstracciones de bajo nivel del sistema, pueden surgir errores en las especificaciones de nivel más alto y deben ser corregidas.

La definición de axiomas que establecerán el comportamiento de las operaciones del tipo de datos abstracto debe empezar por identificar las llamadas "operaciones constructoras". Estas operaciones son las que se usan para construir ejemplares del tipo de datos y, en la mayoría de los casos, constituyen las operaciones de creación y de adición de elementos a la estructura de datos. Al utilizar cada operación constructora, se define un axioma para todas las otras operaciones del tipo de datos.

Técnicas Gane-Sarson, Top-Down y Bottom-Up.

Gane-Sarson: Conocido como SSA o Structured System Analysis (Análisis Estructurado de Sistemas). Utiliza un lenguaje gráfico para construir los modelos de los sistemas, incorpora conceptos de bases de datos; sin embargo, SSA no proporciona una gran variedad de mecanismos estructurales. Hay cuatro elementos básicos en SSA: diagramas de flujo de datos, diccionarios de datos, representaciones lógicas de procedimientos y técnicas de estructuración de almacenamiento de datos.

En la figura 1.3.2.4.1. se muestra un diagrama de flujo de datos SSA. Los rectángulos abiertos indican almacenamiento de datos, las etiquetas en los arcos denotan elementos de datos, los rectángulos sombreados fuentes y destinos de datos, y los rectángulos restantes indican pasos de procesamiento.

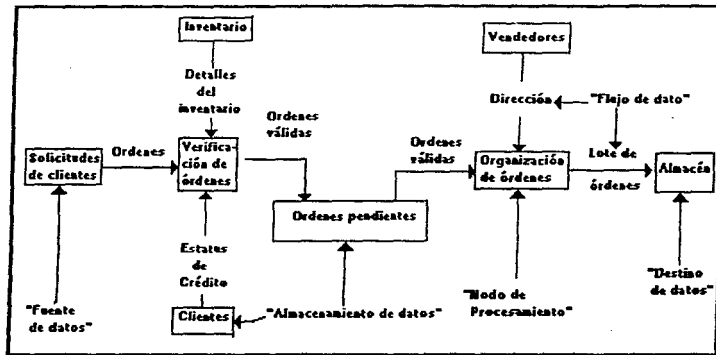


Figura L3.2.4.1 Diagrama de flujo de datos SSA.

Se utiliza un diccionario de datos para definir y registrar los elementos de datos y para especificar los detalles del procesamiento algorítmico se emplean las representaciones lógicas de los procesamientos, tales como las tablas de decisión y el idioma estructurado. Las representaciones lógicas del procesamiento se utilizan para especificar precisamente la secuencia de procesos en términos entendibles para el usuario y diseñador.

Existe el peligro de que los programadores, al usar SSA, se interesen en el nivel de diseño detallado para algunos nodos de procesamiento antes de completar el diagrama de flujo de datos y el diccionario. Esto puede recomendarse para algunos nodos críticos, pero en general, es mejor diferir las consideraciones de diseño detallado hasta concluir el análisis.

Una característica importante de SSA, es el empleo del modelo relacional para especificar flujos de datos y almacenamientos. Las relaciones están formadas con campos de los registros de datos llamados dominios.

CAPITULO I

Bottom-up: Cuando se usa este método, cada nodo de bajo nivel en el sistema jerárquico es probado en forma individual. Entonces el próximo módulo a ser evaluado llama a los módulos previamente evaluados. Esto se hace iterativamente hasta que todos los módulos sean incluidos en la prueba. Este método es utilizado cuando muchos de los módulos de bajo nivel son rutinas de propósito general que son utilizadas para obtener otras o cuando el diseño es orientado a objetos.

Esta técnica primero evalúa los módulos de nivel más bajo (que son los más detallados y específicos) y deja al final los módulos de más alto nivel (que son generalmente los más importantes). Por ello los errores mayores se descubren hasta el final de la prueba.

Top-Down: El nivel más alto, que generalmente es un módulo de control, es probado desde el principio. Entonces, todos los módulos llamados son probados hasta terminar la unidad. Esto se repite hasta que todos los módulos son incorporados.

Un módulo que ha sido probado puede llamar a otro que no ha sido probado. Cuando esto ocurre, se escribe un programa de propósito especial para simular la actividad del módulo faltante.

El programa que simula las actividades de un módulo faltante se llama **stub**(fragmento).

Como es obvio, en este caso, se pueden localizar rápidamente las fallas en los módulos más importantes, que son las fallas más críticas.

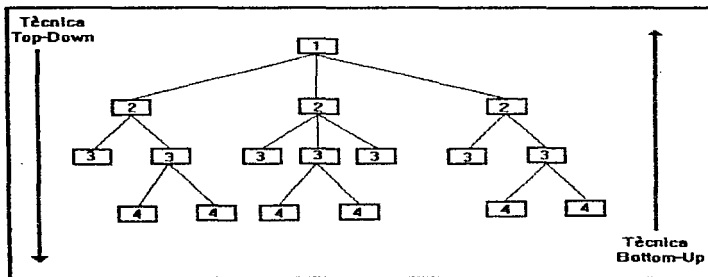


Figura L3.2.4.2 Técnicas Bottom-Up y Top-down.

TECNICA	VENTAJAS	DESVENTAJAS
GANE-SARSON	<ul style="list-style-type: none"> - Utiliza un lenguaje gráfico - Incorpora conceptos de bases de datos - Representación lógica entendible para el usuario y diseñador - Emplea modelo relacional 	<ul style="list-style-type: none"> - No proporciona una gran variedad de mecanismos estructurales - Los programadores podrían interesarse en un nivel más detallado antes de completar el diccionario y el flujo de datos.
TOP-DOWN	<ul style="list-style-type: none"> - Se puede llamar a módulos no probados aún - Se pueden localizar rápidamente las fallas en los más importantes. 	<ul style="list-style-type: none"> - Se tiene que escribir un programa que simule el módulo faltante
BOTTOM-UP	<ul style="list-style-type: none"> - Útil cuando muchos de los módulos de bajo nivel son rutinas de propósito general - Útil cuando el diseño es orientado a objetos 	<ul style="list-style-type: none"> - Los errores se descubren hasta el final de la prueba

L3.3 LA INTERFAZ DEL USUARIO.

Una interfaz mal diseñada puede hacer que el usuario cometa errores catastróficos. Si la información se presenta de una forma confusa o engañosa, el usuario puede interpretar mal el significado de un elemento de información, y a partir de ahí seguir acciones que pueden resultar peligrosas.

Las interfaces del usuario se pueden clasificar en interfaces interactivas en línea y en interfaces fuera de línea. En una interfaz de línea, el usuario interactúa en forma directa con el sistema de computación a través de su terminal. Una interfaz fuera de línea, se basa en que el usuario prepara por separado una entrada legible para la máquina y después la presenta al sistema de computación.

CAPITULO 1

El diseño de una interfaz adecuada requiere conocer tanto los procedimientos del usuario como lo que resulta práctico proporcionar con el sistema de información.

Las posibilidades humanas y el poder de la máquina son complementarios, y es tarea del diseñador de la interfaz asegurar que trabajen bien juntas.

1.3.3.1 DISEÑO DE INTERFACES DEL USUARIO.

En general los sistemas de interfaces se pueden clasificar como sigue:

- a) Interfaces iniciadas por el usuario.
- b) Interfaces iniciadas por la computadora.

El principio fundamental del diseño de interfaces del usuario es que éste se debe adecuar a las necesidades y posibilidades del usuario individual. Diseñar la interfaz a la medida del usuario significa que debe estar en términos familiares para éste.

El segundo principio es que el diseño de interfaces debe ser consistente. Esto significa que los mandatos del sistema deben ser del mismo formato, los parámetros se deben pasar a todos los mandatos de la misma manera y la puntuación de estos debe ser similar.

El tercer principio del diseño de interfaces es que la interfaz debe tener dispositivos integrados de "ayuda". Los dispositivos de ayuda se deben estructurar de modo que el usuario no se vea abrumado con la información cuando pida ayuda.

Las interfaces iniciadas por el usuario se pueden clasificar en dos clases:

- a) *Interfaces controladas por mandatos.* En este tipo de interfaz, el usuario introduce mandatos o preguntas que la computadora interpreta.
- b) *Interfaces orientadas a las formas.* Aquí, el usuario llama a una imagen de una forma en la pantalla y la llena escribiendo en las partes apropiadas de la pantalla.

En general, las interfaces orientadas a formas tiene un alcance más limitado que las interfaces regidas por mandatos y son más adecuadas para usar en situaciones en las que los usuarios tienen poca experiencia o poca comprensión de la computación.

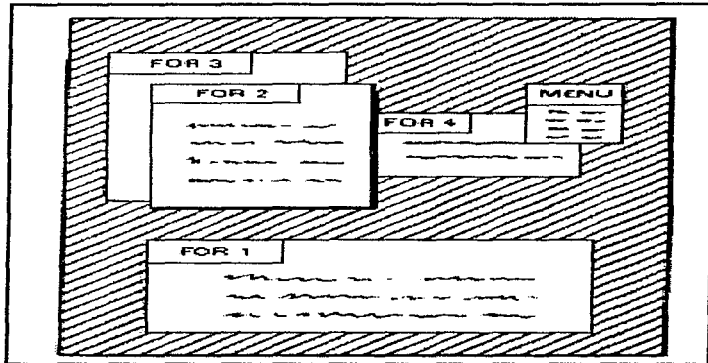


Figura I.3.3.1 Interface con formas.

I.3.3.2 INTERFACES INICIADAS POR LA COMPUTADORA.

Las interfaces iniciadas por la computadora son aquellas en las cuales el sistema de cómputo guía el progreso del diálogo hombre-máquina.

Estas interfaces se agrupan en dos grandes clases:

- a) Los sistemas de menú en donde se presenta al usuario una lista de posibilidades y éste elige una de ellas.
- b) Sistemas de pregunta y respuesta en donde la computadora hace una pregunta al usuario e inicia una acción basada en la respuesta.

1	Notas
	Precios determinados
2	Mantenimiento
	Acuerdos de Mantenimiento
3	Fallos
	Carta de fallos de hardware
4	Errores
	Errores de software advertidos
5	Ampliaciones
	Nuevo sistema propuesto
6	Mejoras
	Mejoras propuestas al sistema
7	Contabilidad
	Información de contabilidad
8	Nuevos usuarios
	Nuevos usuarios desde 1993
9	Catalogado
	Archivos catalogados

Figura 1.3.3.2.1 Sistema de menú.

Un sistema de pregunta y respuesta se usaría con preferencia sobre las interfaces tipo menú cuando se tiene que construir una imagen general de los requisitos del usuario teniendo en cuenta las respuestas a varias preguntas.

1.3.3.3 EL USO DE GRAFICAS.

Los sistemas gráficos tienen la ventaja de que la información almacenada y procesada por la computadora se puede desplegar de forma que el usuario obtenga una impresión general de las entidades descritas por esa información. Este tipo de impresión general es lo que necesitan muchos usuarios del sistema. Un administrador que utilice un sistema de información, suele estar más interesado en las tendencias y en los patrones de sus datos que en cifras exactas. Estas tendencias pueden ser difíciles de percibir si la presentación de los datos es alfanumérica, sobre todo cuando se buscan correlaciones. El despliegue gráfico de los datos, hace que las tendencias se noten de inmediato.

La capacidad de los sistemas gráficos para presentar información aproximada en una forma fácil de asimilar, también se puede usar en el diseño de interfaces del usuario de sistemas de control incorporados por computadora.

Además de presentar la información de una forma más aceptable para muchas clases de usuarios, las interfaces gráficas también pueden tener asociados dispositivos más fáciles y cómodos de usar que los teclados convencionales.

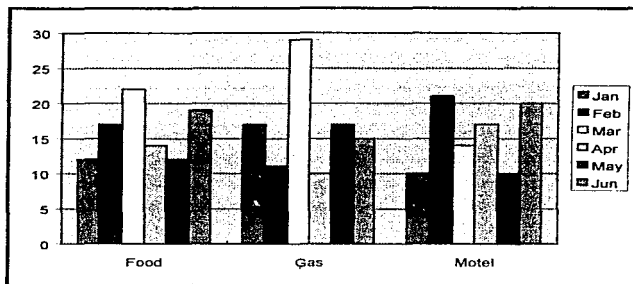


Figura 1.3.3.3.1 Histograma.

1.4. BASES DE DATOS

1.4.1 CONCEPTOS GENERALES.

¿QUE ES UN SISTEMA DE BASES DE DATOS?

En el mundo actual se están presentando cambios que impactan radicalmente nuestra sociedad. Fenómenos como la Globalización de la Economía el libre mercado y la sociedad apoyada en la información son factores que modifican indudablemente nuestra forma de vida. Por lo tanto, cualquier plantel educativo y/o organización grande o pequeña, no se pueden sustraer de esta realidad.

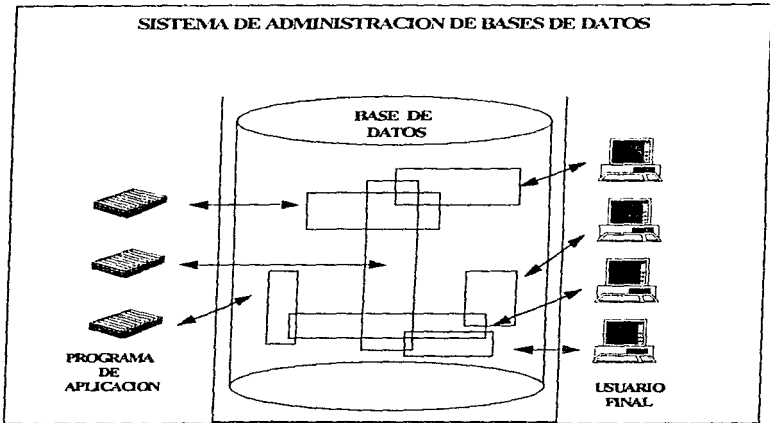
CAPITULO I

El fenómeno para hacer frente a los problemas de información, hace resaltar las facilidades que ofrece hoy en día la tecnología. Esto da como resultado que los empleados, sin importar su especialidad, sean más productivos en su trabajo, pero al mismo tiempo, ha provocado que las personas adquieran una mayor "Cultura Informática" fenómeno que demanda tareas especializadas en el desarrollo de sistemas, así como mejorar la atención hacia los usuarios.

Las Bases de Datos, se han descrito como una de las áreas de la ciencia de la computación y la información de más rápido desarrollo; los fabricantes y vendedores no empezaron a ofrecer sistemas de administración de bases de datos, sino hasta la década de 1960, ya que antes para una organización de este tipo se utilizaban los archivos y los conjuntos de datos, quienes no ofrecían las ventajas que hacen de las bases de datos una mejor opción:

La integridad de datos, independencia de información, no redundancia, seguridad de acceso, recuperación de cambios incompletos, actualización más rápida y efectiva, es dinámica ya que se pueden hacer respaldos parciales ó totales, tiene una administración interna y una de las ventajas más grande de un sistema de Bases de Datos es que nos sirve para estandarizar la información en cualquier empresa por muy grande que esta fuese; además que en muchos lugares de organización dependen de la operación continuada y eficaz de un sistema de bases de datos.

Ahora bien, un **sistema de Bases de Datos**, no es más que un *SISTEMA DE MANTENIMIENTO DE REGISTROS BASADO EN UNA COMPUTADORA*, y su propósito general es *REGISTRAR Y MANTENER INFORMACIÓN*. Esta información puede estar relacionada con cualquier cosa que sea significativa para la organización donde el sistema este trabajando.



Un sistema de Bases de Datos, incluye los siguientes componentes principales:

- Datos .
- Lenguaje de definición de datos.
- Manejador de base de datos.
- Usuarios.

- **DATOS:**

Estos son almacenados en el sistema, se dividen en una o más Bases de Datos. y hacen de una B.D. que sea *integrada* o *compartida*, por la primera entendemos que puede considerarse como una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos, por *compartida*, concurrente ó de usuarios *múltiples*, entiéndase que partes individuales de la base de datos pueden compartirse entre varios

CAPITULO I

usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la base de datos y utilizarla con propósitos diferentes.

- **LENGUAJE DE DEFINICIÓN DE DATOS.**

Un esquema de bases de datos se especifica por un conjunto de definiciones que son expresadas por un lenguaje especial llamado **DATA DEFINITION LANGUAGE (DDL)**, este se encarga, del almacenamiento y métodos de acceso al sistema de bases de datos.. El resultado de la compilación del código de DDL es un conjunto de tablas que son almacenadas en un archivo especial llamado diccionario de datos.

Un diccionario de datos es un archivo que contiene datos sobre los datos. Este archivo es consultado antes de que los datos reales sean leídos o modificados en el sistema de bases de datos.

El resultado de la compilación de estas definiciones es un conjunto de instrucciones para especificar la implantación de detalles de los esquemas de bases de datos que están ocultos a los usuarios.

- **MANEJADOR DE BASE DE DATOS:**

Este es un conjunto de programas que proporcionan la interfaz entre los datos de bajo nivel almacenados en la base de datos, los programas de aplicación y consultas del sistema; es responsable de lo siguiente:

a) **Interacción con el Manejador de archivos.** Los datos son almacenados en disco usando el sistema de archivos que es proporcionado por un sistema operativo convencional. El manejador de la base de datos traduce las instrucciones del **DATA MANAGEMENT LANGUAGE (DML)** en comandos de sistema de archivos de bajo nivel. Entonces, el manejador es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos.

b) **Garantizar la Integridad** (de la que más adelante se habla). El valor de los datos almacenados en la base de datos debe satisfacer ciertos requerimientos de consistencia. Estos deben ser especificados por el administrador de la base de datos y así el manejador de la base de datos puede checar si las actualizaciones a esta no violan sus restricciones, tomando medidas necesarias.

c) **Garantizar la seguridad.** No todos los usuarios podrán acceder todos los datos, y esto es responsabilidad del manejador. En otras palabras, éste debe conservar y aplicar la seguridad de los datos en la base de datos.

d) Respaldo y recuperación. La responsabilidad del manejador es detectar las fallas que pudieran afectar los datos y restaurar la base de datos al estado en que se encontraba antes de la ocurrencia de la falla. Esto se lleva a cabo mediante la iniciación de procedimientos de respaldo y recuperación.

e) Control de concurrencia (usuarios compartidos). Cuando varios usuarios accesan la base de datos concurrentemente, la consistencia de los datos puede ser dañada. Es necesario que se controle la interacción entre los usuarios concurrentes y ésta es una de las responsabilidades del manejador.

ESTRUCTURA GENERAL DEL SISTEMA.

Un sistema de base de datos se divide en módulos que se encargan de cada una de las funciones del sistema. Algunas de estas pueden ser proporcionadas por el sistema operativo de la computadora. En la mayoría de los casos, el sistema operativo prevé solamente los servicios básicos y el sistema de bases de datos debe ser constituido sobre éste.

Un sistema de base de datos consiste en un número de componentes funcionales, los cuales incluyen:

a) Manejador de archivos: Controla la distribución de espacio de almacenamiento en disco y las estructuras de datos usadas para representar la información almacenada en disco.

b) Manejador de la Base de Datos: Suministra la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y queries que se envían al sistema.

c) Procesador de queries. Traduce instrucciones de un lenguaje de Query a instrucciones de bajo nivel que sean entendidas por el manejador de base de datos. Además, éste intenta transformar los pedidos del usuario a una forma equivalente pero más eficiente buscando una nueva estrategia para ejecutar el Query.

d) Precompilador DML. Convierte las instrucciones de DML incluidas en un programa de aplicación a llamadas normales de procedimientos en el lenguaje de programación. El precompilador debe interactuar con el procesador de query para generar el código apropiado.

e) Compilador DDL. Transforma las instrucciones de DDL a un conjunto de tablas que contienen metadatos. Estas tablas son almacenadas en el diccionario de datos.

Adicionalmente se requieren también varias estructuras de datos como parte de la implementación del sistema físico, que incluyen:

CAPITULO I

e.1.) Archivo de datos: Los cuales almacenan la base de datos.

e.2.) Diccionario de datos: Almacena información sobre la estructura de la base de datos, éste se utiliza por los componentes funcionales por esto se debe dar gran énfasis al logro de un buen diseño y una implementación eficiente del diccionario.

e.3.) Índices: Proporcionan un acceso rápido a los datos con valores particulares.

● USUARIOS:

Existen tres tipos de USUARIOS:

- Programador de aplicaciones,
- Usuario final
- Administrador de Bases de Datos.

a) El programador de aplicaciones, se encarga de escribir programas de aplicación que utilicen bases de datos para nuestro caso utilizaremos el manejador ORACLE 7, estos programas de aplicación operan con los datos de todas las maneras usuales: recuperan información, crean información nueva, suprimen o cambian información existente, etc. Los programas en si, pueden ser aplicaciones convencionales de procesamiento por lotes o programas en línea diseñados para apoyar a un usuario final, quien interactúa con el sistema desde una terminal en línea.

b) El usuario final accesa a la base de datos desde una terminal. Un usuario final puede emplear un lenguaje de consulta proporcionado como parte integral del sistema ó recurrir al programa de aplicación escrito por un usuario de aplicación que acepte órdenes desde la terminal y a su vez formule solicitudes al DBMS en nombre del usuario final.

c) El administrador de base de datos (DBA) que tiene un control central de los datos y los programas que los accesan, éste tipo de usuario, es el que lleva el control del sistema, se encuentra entre la base de datos (almacenamiento real de los datos) y el usuario final, maneja todas las solicitudes de acceso a la base de datos formulados por los usuarios.

Sus funciones:

Definición de esquemas: Creación del esquema original de la base de datos.

Estructura de almacenamiento y definición del método de acceso: La creación de estructuras adecuadas de almacenamiento y métodos de acceso. Esto se logra escribiendo un conjunto de definiciones que son traducidas por el compilador.

Permitir acceso de datos: Diversos niveles de acceso a los diversos usuarios. Esto le permite regular a que partes de la base pueden acceder los diferentes usuarios.

Restricciones de integridad: Para realizar una actualización en la base, el manejador consulta estas restricciones, creadas con anterioridad.

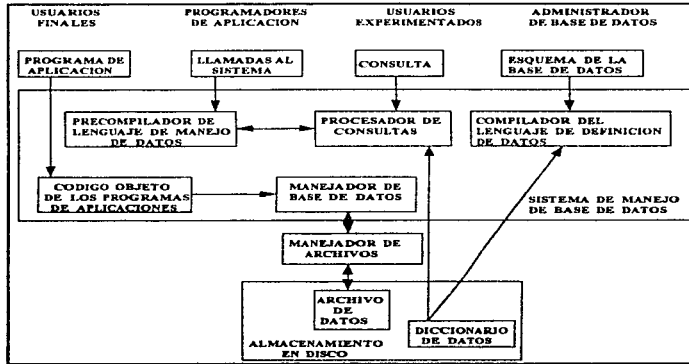


Figura I.4.1.2. Estructura general de una base de datos

I.4.1.1. VENTAJAS

Las ventajas de los sistemas de B.D. repercuten en todas las etapas del ciclo de vida del sistema, estas son:

a) **La etapa de diseño.** Está facilitada por la estructura modular y por el hecho de que todos los programas de aplicación y los subsistemas de integridad y privacidad se refieren a las bases de datos en términos lógicos. Cada uno de estos módulos puede ser diseñado independientemente y los diseñadores no necesitan preocuparse de los detalles de implantación de la estructura física de almacenamiento utilizada.

b) La presencia de esquemas conceptuales y de bases de datos facilitan el empleo del sistema. Los usuarios finales pueden referirse a estos esquemas para aclarar su comprensión del área de aplicación y para descubrir qué datos están almacenados en la base de datos.

c) El módulo de transformación lógica a física. Facilita en gran medida el mantenimiento del sistema. La introducción de nuevos equipos y/o sistemas operativos tiene impacto mínimo sobre el sistema si este módulo está presente.

En la práctica pocos sistemas existentes se adecuan a la arquitectura ideal. Esto se debe en gran parte a la dificultad para diseñar el módulo de transformación lógica a física. En general, los programas de aplicación están directamente ligados con la estructura de almacenamiento de datos y los controles de privacidad e integridad; todos estos están inmersos en programas de aplicación.

1.4.1.2. ¿POR QUE UTILIZAR BASES DE DATOS?

Un sistema de Bases de Datos, proporciona a la empresa un control centralizado de sus datos de operación que constituyen uno de sus activos más valiosos. Según se mencionó antes, las ventajas de tener un control centralizado de los datos se especifica enseguida:

A.- REDUCCIÓN DE REDUNDANCIAS.

En sistemas que no usan bases de datos, cada aplicación tiene sus propios archivos privados. Esto origina enorme redundancia en los datos almacenados, así como desperdicio que se da del espacio de almacenamiento. Cabe aclarar que en una base de datos no toda la redundancia se eliminará. A veces hay sólidas razones técnicas para mantener múltiples copias de los mismos datos. Pero en nuestros sistemas la redundancia se controla ya que debe estar al tanto y asumir la responsabilidad de propagar las actualizaciones.

B.- PUEDE EVITARSE LA INCONSISTENCIA.

Una base de datos que posea inconsistencia puede suministrar información incorrecta o contradictoria. Pero si la redundancia es eliminada y controlada este estado no se dará en el sistema.

Para el usuario el asegurar que cualquier cambio hecho desde cualquier entrada se efectúe de manera automática en todo el sistema de base de datos es algo muy importante. Este proceso se denomina propagación de actualizaciones.

C.- LOS DATOS PUEDEN COMPARTIRSE.

No solo implica que las aplicaciones existentes puedan compartirse, sino también que es factible desarrollar nuevas aplicaciones que operen con los mismos datos almacenados. En otras palabras, las necesidades de datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos almacenados.

D.- PUEDEN APLICARSE RESTRICCIONES DE SEGURIDAD

Al tener jurisdicción completa sobre los datos de operación, el DBA puede:

- 1.- Asegurar que el único medio de acceder la base de datos sea a través de los canales establecidos y en consecuencia:
- 2.- Definir controles de autorización para que se apliquen cada vez que se intente el acceso a los datos .

E.- PUEDE CONSERVARSE LA INTEGRIDAD.

Si los datos de la base de datos, no son exactos, entonces caemos en un problema de integridad. Un ejemplo sería el que el sistema reportara cantidades exageradas en alguna rutina de trabajo o bien que el número de empleados sea mucho mayor al total del que se tiene en la nómina.

El control centralizado de la base de datos ayuda a evitar estas situaciones en la medida de lo posible, pues permite al DBA realizar algún proceso o proceso de validación que habrán de efectuarse cada vez que se intente una operación de actualización y así la aplicación no genera errores de datos que afecten a otros programas que utilicen esa información.

F.- INDEPENDENCIA DE LOS DATOS.

Si los datos se organizan en almacenamiento secundario y la manera como se accesan dependen de los requerimientos de la aplicación, además del conocimiento de la organización de los datos y de la técnica de acceso formando parte de la lógica del sistema, estamos frente a una dependencia de datos, ya que es imposible cambiar la estructura de almacenamiento (la manera como están físicamente los datos), la estrategia de acceso ó cualquier definición del esquema sin afectar la aplicación. Esta dependencia de datos no es recomendable para un sistema de bases de datos por dos razones.

CAPITULO I

I.- Aplicaciones diferentes requerirán vistas diferentes de los mismos datos.

II.- El DBA debe tener la libertad de modificar la estructura de almacenamiento o la estrategia de acceso o ambas, en respuesta al cambio de necesidades sin tener que alterar las aplicaciones existentes.

La provisión de independencia de los datos es un objetivo esencial de los sistemas de bases de datos, y la podríamos definir como **la inmunidad de las aplicaciones a los cambios de la estructura de almacenamiento y de la estrategia de acceso.**

L4.1.3. DESVENTAJAS

A.- Experiencia insuficientes en bases de datos. Quizá la desventaja más seria es que la tecnología de las bases de datos es compleja y muy pocas organizaciones tienen personal con la suficiente experiencia para implantar y manejar las bases de datos correctamente. Esto aumenta la probabilidad de implantación sin éxito o con éxito parcial de las bases de datos.

B.- Costos más altos de procesamiento de datos. Las bases de datos son responsables en general del aumento significativo en el procesamiento de datos en una organización. En gran medida se debe a los programas del DBMS, que son grandes, deben procesarse para acceder, obtener y actualizar los datos, y en parte se debe a la cantidad de trabajo asociado con el manejo y la reorganización de los sistemas de archivos complejos.

C.- Aumento en las necesidades de hardware y software. La capacidad de memoria de acceso directo, una mayor capacidad de comunicaciones (incluyendo paquetes de software de comunicaciones), y una capacidad adicional en el procesador se requieren en la mayoría de los sistemas de bases de datos. Por supuesto, esto aumenta en forma sustancial los costos de hardware.

D.- Seguridad e integridad de los datos. Son los principales problemas de una base. La mayoría se relacionan con el hecho de que muchos usuarios tienen derecho de acceso a la base de datos y que las capacidades asociativas de esta última permiten que todos los datos estén disponibles en una forma u otra para cada usuario. Deben implantarse sistemas de seguridad para evitar al personal no autorizado su acceso a secciones de la base de datos.

Otro problema relativo del anterior, es el que los usuarios de microcomputadoras buscan tener acceso regular a las bases de datos para obtener datos con el fin de manipularlos con sus micros.

Existe el peligro de que estos usuarios contaminen la base de datos alterando los archivos sin darse cuenta, existe también el inconveniente, de que en un disco flexible de la microcomputadora, pueden ser copiados los datos con rapidez y sacarse del lugar sin ser detectado.

En muchas situaciones un archivo de bases de datos perdido o destruido que no puede reconstruirse llevaría a la organización a la bancarrota, por esta razón, deben implementarse controles para recuperar datos a través de otras fuentes de datos establecidas. Estos sistemas de recuperación de bases de datos son complejos y tienen altos costos por su continuo mantenimiento.

1.4.1.4. ENTIDADES Y ATRIBUTOS:

Los objetos básicos acerca de los cuales se registran datos en la base de datos, es lo que se conoce como **entidad**, y esta puede ser un objeto determinado como un paciente, o bien algo imperceptible como un suceso. Las entidades tienen características propias que conviene registrar a los que se le conoce como **atributos**. (ver figura 1.4.1.3.)

Así como las entidades describen el mundo real, existen términos que se utilizan para describir la información acerca de ellas. Por lo general mantenemos un registro para cada entidad y estos se agrupan en conjuntos de registros pertenecientes a entidades similares. Los registros se refieren a atributos de las entidades y contienen los valores de estos atributos.

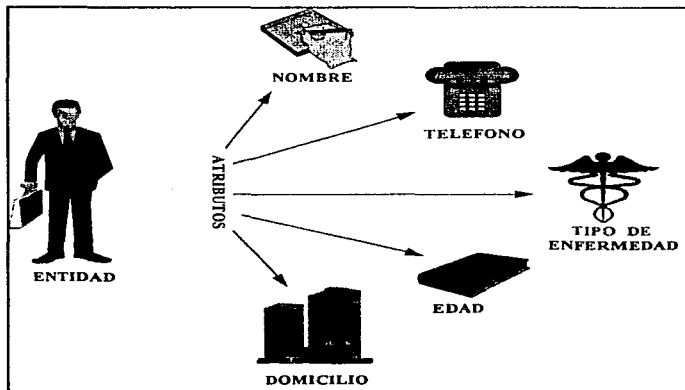


Figura I.4.1.3. Entidades y atributos.

La manera de asociar un valor con un dato y de asociar datos con atributos de entidad, consiste en almacenar juntos los datos en una secuencia fija, por ejemplo un registro de enfermedades en un determinado consultorio médico. En este podrían usarse valores diversos que identifiquen al atributo.

Un identificador de entidad es un atributo que es único a esa entidad y que sirve para reconocerlo y en ocasiones se requiere más de un atributo para identificar un registro.

Una llave es el atributo o conjunto de estos que se utilizan para reconocer un registro. Una llave primaria se utiliza para definir unívocamente un registro, es decir el identificador de entidad formado por uno o más atributos. También se usan llaves que no identifican registros únicos, sino todos aquellos que tienen cierta propiedad, y a estas se les conoce como secundarias. (figura I.4.1.4).

LLAVE PRIMARIA		
Nº. DE PACIENTE	NOMBRE DEL PACIENTE	ENFERMEDAD
3333	JUAN GOMEZ BILBAO	ARTRITIS
3334	JUAN GALINDO BRANZ	CELULITIS
3459	JUANA GASTELU BRITO	REUMAS CRONICAS

LLAVE SECUNDARIA		
ENFERMEDAD	PACIENTE	Nº. DE PACIENTE
OSTEOPOROSIS	RAMIRO ZAPATA LOPEZ	4535
OSTEOPOROSIS	GENARO CURTIS ORTA	5563
OSTOPOROSIS	OLIVIA NEWTON RUIZ	3482

Figura I.4.1.4. Llaves primarias y secundarias.

I.4.1.5 ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS.

Aquí hablaremos de la arquitectura de un sistema de base de datos, para poder proporcionar un marco de referencia y describir los conceptos generales sobre las bases de datos.

No se pretende que cada sistema corresponda con fidelidad a este marco de referencia específico, ni se da a entender que la arquitectura presentada aquí sea la única.

El manejo y organización de la información en una Base de Datos puede ser vista en 4 niveles de Arquitectura que van desde el nivel de Organización Lógico hasta su correspondiente en el nivel de Organización Físico, desde el más independiente del equipo hasta el más dependiente.

CAPITULO I

En el nivel 4, el más alto, es donde se hacen sólo las consideraciones lógicas de Datos, donde se hace el análisis de necesidades del usuario, de los datos con que cuenta y los que necesita, se especifica el flujo de la información y sus transformaciones. Posteriormente se definen entidades u objetos y sus relaciones funcionales.

A los niveles 3 y 4 se les da el nombre de interfaces físicas - lógicas, El nivel 2 refleja alternativas de organización Física y formatos de almacenamiento de datos en dispositivos de almacenamiento lineal para cada una de las organizaciones definidas en el nivel 3. Algunos autores sólo reconocen 3 niveles y manejan como uno solo al 3 y al 2.

El nivel 1, es más dependiente que los anteriores del equipo donde se implementa la Base de Datos, se refiere a las características y aspectos particulares de los dispositivos físicos de almacenamiento. En este nivel se ven aspectos como la organización de los dispositivos en subdivisiones tales como bloques y sectores; el mapeo de datos dentro de estas subdivisiones; el almacenamiento de áreas de sobreflujo y características dependientes de cada máquina, así como la localización de datos de control, verificación de integridad etc.

ARQUITECTURA ANSI/SPARC

EN 1972 fue fundado el grupo de estudio de ANSI/X3/SPARC sobre sistemas de Administración de Bases de Datos por el Standard Planning and Requirements Committee (SPARC) de ANSI/X3 (American National Standards Committee on Computers and Information Processing), para determinar las áreas, si las había de la tecnología de bases de datos para los cuales la estandarización era adecuada y producir un conjunto de recomendaciones de acción en cada una de las mismas. Este grupo definió una arquitectura generalizada que presentó en un reporte provisional en 1975.

La arquitectura se divide en tres niveles generales: Interno, conceptual y externo figura 1.4.1.5 (ANSI / SPARC)

A) El nivel **INTERNO** es el más cercano al almacenamiento físico, es decir, el que corresponde a la manera como los datos se almacenan en realidad. Aquí se define dónde están colocados los valores de atributo de la base de datos y cómo se obtiene el acceso a ellos; se ve el desempeño y las afinaciones para equilibrar los tiempos de respuesta de algunas transacciones, y se manejan los dispositivos de almacenamiento.

B) El nivel **EXTERNO** es el más cercano a los usuarios, es decir, es una representación que el usuario directo del sistema entiende perfectamente; En este nivel se cuenta con diversos

lenguajes orientados a aplicaciones, por ejemplo: COBOL, CLIPER, C (que es el que se utilizará para esta aplicación), etc.

Estos lenguajes cuentan con sublenguajes para la definición de datos (DDL) que sirven para la descripción de los objetos de la base tal como los ve el usuario y sublenguajes de manipulación de datos (DML) que apoyan al manejo o procesamiento de esos objetos.

C) El nivel **CONCEPTUAL**, este es el nivel interprete, es decir un mediador entre los otros dos, si el nivel externo se relaciona con las vistas de los usuarios individuales, el nivel conceptual puede considerarse como el que define la vista de la comunidad de usuarios. En este nivel se implanta la Base de Datos en su forma más general, se define el esquema de la Base de Datos, sus vínculos, relaciones, etc. La función principal en este nivel es el diseño y la generación de esquemas. (figura I.4.1.5.)

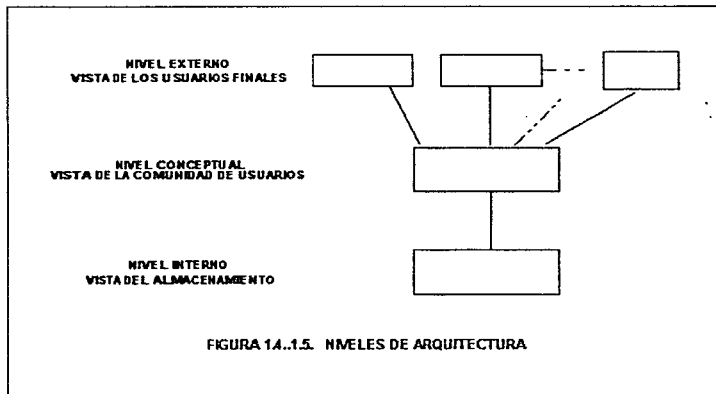


Figura L4.1.5. Niveles de arquitectura.

CAPITULO I

Examinando los componentes de la arquitectura con un poco más de detalle los usuarios son programadores de aplicaciones o usuarios de terminales de línea cuyo nivel de destreza varía, cada usuario tiene un lenguaje a su disposición; para el usuario de una terminal se trata de un lenguaje de consulta o un lenguaje de propósito especial hecho a la medida de sus necesidades y apoyado por un programa de aplicación en línea, lo importante de esto es que incluye un sublenguaje de datos (DSL), es decir un subconjunto del lenguaje total que concierne a los objetos y a las operaciones de la base de datos.

Se dice de éste que está inmerso en un lenguaje anfitrión un sistema específico puede incluir múltiples lenguajes anfitrión y numerosos lenguajes de datos.

En principio cualquier sublenguaje de datos en realidad es una combinación de dos lenguajes: Un lenguaje de definición de datos (DDL) que permite la definición o descripción de los objetos de la base de datos (como el usuario lo entiende), y un lenguaje de manipulación de datos (DML) que apoya el procesamiento de esos objetos.

Una vista **externa** es el contenido de la base de datos tal como lo ve un usuario específico (Es decir es la base de datos en sí).

La **vista conceptual** es una representación del contenido total de información de la base de datos, una vez más en forma relativamente abstracta en comparación con la forma en la cual los datos se almacenan físicamente. (Puede ser también diferente de la manera en que cualquier usuario específico vea los datos). En términos generales, se pretende que sea una vista de los datos como son en realidad y no como los usuarios están obligados a verlos por las restricciones del lenguaje en particular o del hardware que utiliza.

El tercer nivel de la arquitectura, es el **interno**. La vista interna es una representación de nivel muy bajo de la base de datos en su totalidad; En este nivel se implantan las definiciones de almacenamiento a nivel registro almacenado; la vista interna aún se mantiene a un **paso** del nivel físico ya que no se involucra con registros físicos o bloques, ni con ninguna restricción específica de dispositivos tales como capacidades de cilindros o pistas.

L.4.1.5.1. ¿QUE ES UN MODELO?

Una de las responsabilidades más grande de una base de datos administrativa, es el desarrollo de un **MODELO CONCEPTUAL**, también llamada modelo emprendedor de la organización. El modelo da cuerpo a las entidades y a sus relaciones, y es la herramienta

utilizada, para representar la organización conceptual de datos. El modelo conceptual, es una herramienta de comunicación entre varios usuarios de datos y el desarrollo de la base, sin tomar en cuenta la representación física.

El modelo conceptual, es usado para organizar, visualizar, planear y comunicar ideas. Es o debe ser independiente del sistema manejador de base de datos (DBMS).

Las entidades en la organización y las relaciones entre estas, pueden ser representadas por el modelo de datos. El Sistema Manejador de Bases de Datos comercial, se puede basar en un MODELO DE DATOS JERÁRQUICO, en un MODELO DE RED ó en un MODELO RELACIONAL o bien alguna combinación de los tres.

El modelo conceptual, tiene que ser mapeado del MODELO LÓGICO, utilizado para delinear estructuras para el Sistema Manejador de Base de Datos (DBMS), y el MODELO LÓGICO, tiene que ser mapeado del MODELO FÍSICO, (también llamado modelo interno o estructura física.) El MODELO LÓGICO, puede ser también relacional, jerárquico o bien de red.

El Sistema Manejador de Base de Datos, no es un factor imprescindible para realizar el MODELO CONCEPTUAL, sin embargo, diseñar el MODELO LÓGICO, si depende de éste.

El elegir, el DBMS debe hacerse después de que el modelo conceptual es designado.

La principal diferencia entre los tres tipos de modelo de datos recae, en la representación de las relaciones entre entidades, de esta se hablará a continuación..

I.4.1.5.2. RELACIÓN SIN MODELO

Una relación es un mapeo o una ligadura entre dos grupos de datos. Estas pueden ser UNO A UNO, UNO A MUCHOS o MUCHOS A MUCHOS.

A) RELACIÓN UNO A UNO (Entre dos entidades) .

Dando un ejemplo en un hospital, un paciente, es asignado a una cama. La relación existente entre el paciente y la cama es uno a uno (1:1) y se representa por una flecha en la señalación. (figura I.4.1.6.A).

B) RELACIÓN UNO A MUCHOS (Entre dos entidades)

Siguiendo con el ejemplo, 0, 1 o muchos pacientes son asignados a un cuarto de hospital, es decir un cuarto de hospital, puede tener ninguno o muchos pacientes, pero un paciente es asignado únicamente a un solo cuarto de hospital. Esta relación gráficamente se representa por

CAPITULO I

una flecha simple yendo en dirección del "uno" y una flecha doble yendo en dirección del "muchos". (figura 1.4.1.6.A.).

C) RELACIÓN MUCHOS A MUCHOS (Entre dos entidades).

En el ejemplo que estamos usando, un cirujano puede operar a muchos pacientes, por otro lado un paciente puede ser intervenido por varios cirujanos en diversas visitas al hospital, se representa esta relación por dos flechas yendo a ambos lados de la relación. (figura 1.4.1.6.A.).

También existen relaciones entre los atributos de las entidades, y se clasifican de igual forma:

D) RELACIÓN UNO A UNO (Entre dos atributos).

Asumimos, que el número de paciente es un identificador único para un paciente, es decir que el número de paciente es usado siempre que el paciente visita el hospital. Si junto con el número de paciente, otro único identificador es almacenado en la base de datos la relación entre estos es de 1:1. (figura 1.4.1.6.D.).

E) RELACIÓN UNO A MUCHOS (Entre dos atributos).

El nombre del paciente y el número, existen juntos. Puede haber muchos pacientes con el mismo nombre, pero su número tiene que ser diferente y único, esto es a un número de paciente dado solo corresponde un nombre. (figura 1.4.1.6.E.).

F) RELACIÓN MUCHOS A MUCHOS (Entre dos atributos).

Un número de pacientes con el mismo nombre podrían ser operados por muchos cirujanos. Un número de cirujanos con el mismo nombre puede operar a muchos pacientes, esta relación es de M:M. (figura 1.4.1.6.F.).

Los modelos jerárquico y de red, se han venido usando como estructuras básicas para el Sistema Manejador de Base de Datos, desde principios de los años 60's, mientras que el modelo de datos relacional fue propuesto como estructura básica hasta la década de los 70's. La principal diferencia entre los tres modelos de datos, es la forma de representar sus relaciones entre entidades.

En el modelo de datos RELACIONAL, las entidades y sus relaciones son representadas con tablas de dos dimensiones. Su relación es considerada como entidad, cada tabla representa una entidad y es construida por renglones y columnas.

El modelo de datos JERÁRQUICO, es compuesto por jerarquías de los tipos de entidad quedando arriba, el tipo de entidad más dominante y una o más subordinadas en los demás niveles.

En el modelo de RED, el concepto de dominante y subordinado, se expande, cualquier entidad puede ser dominante o subordinada, ya sea uno a la vez o ambos si se requiere. Esto significa que cualquier entidad puede participar en un ilimitado número de relaciones.

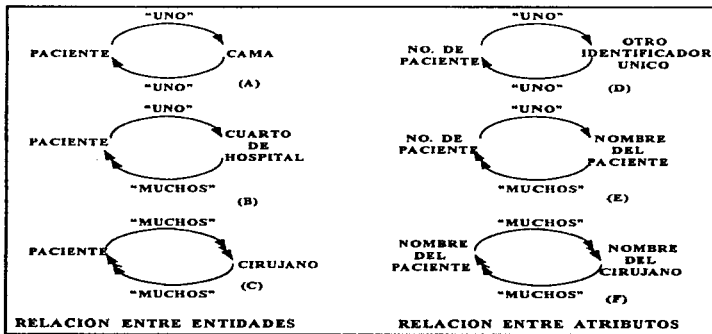


Figura L4.1.6. Relación entre entidades y atributos.

I.4.1.5.3. MODELO JERÁRQUICO.

Una base de datos jerárquica consiste en un conjunto de registros que se conectan entre sí por medio de ligas. Un registro es un conjunto de campos, cada uno de los cuales contiene un solo valor. Una liga es una asociación entre dos registros, exclusivamente.

Un diagrama de estructura de árbol es el esquema de una base de datos JERÁRQUICA, y consta de dos componentes básicos cuadros que corresponden a tipos de registros, y líneas que corresponden a ligas. Un diagrama de estructura de árbol tiene el objetivo de especificar la estructura lógica general de la base de datos. Los diagramas de estructura de árbol son similares a los de estructura de datos del modelo de red.

Haciendo una analogía con la estructura de árbol, en las cuales se tienen nodos, y una relación de jerarquía que se da de un nodo padre a hijo se define un conjunto como una jerarquía de dos niveles de registros. El registro padre es llamado propietario, cada propietario puede tener una ocurrencia del propietario y cualquier número de ocurrencias de los registros hijos llamados miembros. Las relaciones de padre a hijo pueden ser 1:1, 1:M, pero nunca de M:M. En el caso de las relaciones M:M, es necesario repetir registros si se desea conservar la organización de estructura de árbol de la base de datos. La figura 1.4.1.7. ejemplifica el uso de este modelo.

I.4.1.5.3.A. VENTAJAS DEL MODELO JERÁRQUICO.

- A)** La mejor ventaja de este tipo de modelo, es la existencia de probar que el sistema manejador de base de datos usa el modelo de datos jerárquico como estructura básica.
- B)** La relativa simplicidad y facilidad del uso del modelo de datos jerárquico y la familiaridad del procesador de datos con la jerarquía.
- C)** Existe una reducción de la dependencia de los datos.
- D)** El pronostico de su desarrollo es simplificado, a través de sus relaciones predefinidas.

I.4.1.5.3.B. DESVENTAJAS DEL MODELO JERÁRQUICO.

- A)** La relación de M:M, puede ser implementada únicamente por un camino muy largo. Esto puede caer en redundancias, característica que lejos de ayudar al modelo le resta simplicidad.
- B)** Como un resultado de la estricta ordenación jerárquica, la operación de inserción y borrado resulta realmente compleja.
- C)** El deshacer a un padre, trae como consecuencia desaparecer a un hijo. Por tal motivo, debe uno tener mucho cuidado en tomar este tipo de decisiones.

D) Cada nodo hijo solo puede ser accesado a través de su nodo padre, lo que hace más difícil la ruta de acceso.

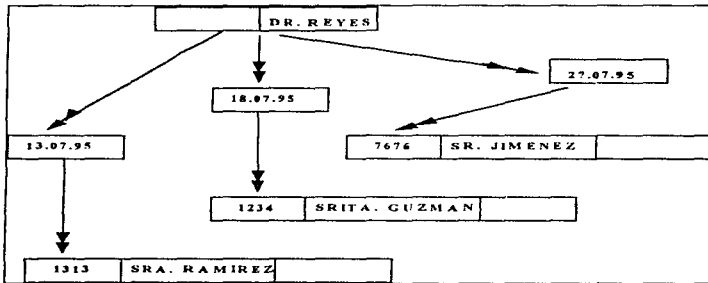


Figura I.4.1.7. Modelo jerárquico.

I.4.2. BASES DE DATOS DISTRIBUIDAS.

La tecnología de bases de datos distribuidas constituye un avance relativamente reciente dentro del campo general de las bases de datos.

En un sistema de bases de este tipo, los datos se almacenan en varias computadoras, las computadoras de un sistema distribuido se comunican entre sí a través de diversos medios de comunicación, como pueden ser cables paralelos de alta velocidad o líneas telefónicas.

Pueden incluir microcomputadoras pequeñas, estaciones de trabajo, minicomputadoras y sistemas de computo grandes de aplicación general. Estos procesadores reciben diferentes nombres: localidades, nodos, etc. dependiendo del contexto en el que se mencionen. Aquí se usará normalmente el término localidad, para hacer hincapié en la distribución física de estos sistemas.

Un sistema distribuido de base de datos consiste en un conjunto de localidades, cada una de las cuales puede participar en la ejecución de transacciones que acceden datos de una o varias localidades. La diferencia principal entre un sistema de bases de datos centralizado y uno

CAPITULO I

distribuido, es que en los primeros, los datos residen en una sola localidad, mientras que en los últimos se encuentran en varias localidades.

Una base de datos distribuida es, por lo común una base de datos no almacenada en su totalidad en un solo lugar físico, sino que se distribuye a lo largo de una red de computadoras geográficamente separados que se conectan por medio de enlaces de comunicación.

La figura 1.4.2.1.1. se refiere a la arquitectura de la base de datos distribuida. Esta arquitectura, no es explícitamente implementada en todas las bases de datos distribuidas, además sus niveles son conceptuales, con el fin de entender la organización de cada una de estas bases. En seguida se analizará y entenderá todos los componentes de esta arquitectura referencial

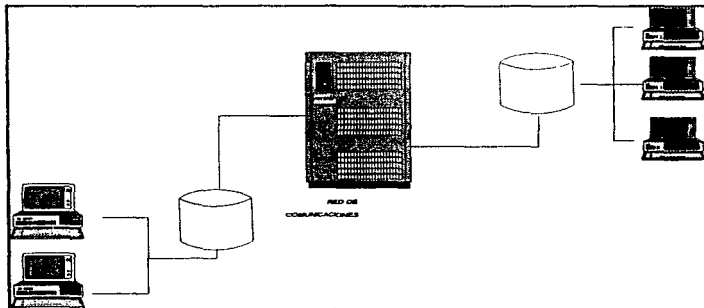


Figura 1.4.2.1 Red de comunicaciones

Se observa que en la parte superior de la figura, se encuentra el **ESQUEMA GLOBAL**, quien define todos los datos que son contenidos en la base de datos distribuida como si la base fuera no distribuida del todo. El modelo de datos que se utiliza para definir el esquema global es conveniente para la definición del mapeo de otros niveles de las bases distribuidas. El esquema global consiste en la definición de la relación global.

Cada relación global, puede ser dividida dentro de pequeñas porciones llamadas FRAGMENTOS. Existen muchos caminos para realizar esta operación. el mapeo entre la relación global y la fragmentación se define en el ESQUEMA DE FRAGMENTACIÓN. Los fragmentos son porciones lógicas de la relación global localizada físicamente en algún lugar de la red. El ESQUEMA DE LOCALIDAD define estos sitios como fragmentos localizados. El tipo de mapeo definido en el esquema de localización determina si es redundante la información o no.

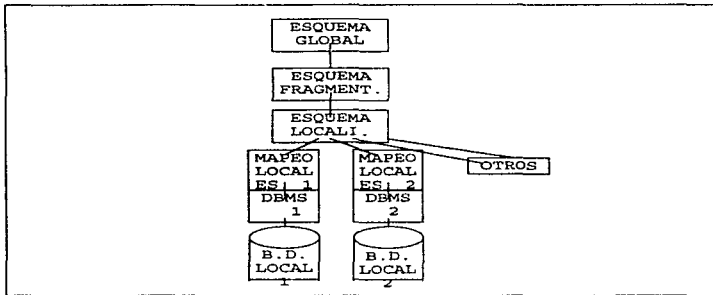


Figura I.4.2.1. Arquitectura de referencia de b.d. distribuidas

Para completar la terminología nos referiremos a una copia de fragmentos, esta usa el nombre de la relación global y dos indexaciones (una indexación de fragmento y una de lugar).

Los tres objetivos más importantes de esta arquitectura, son la separación de datos en la fragmentación y en la localidad así como el control de redundancia y la independencia de la DBMS local.

1. Separando el concepto de la fragmentación de datos de el concepto de localidad de datos. Esta separación nos permite distinguir dos niveles diferentes de distribución, transparencia, llámese FRAGMENTACIÓN TRANSPARENTE y LOCALIDAD TRANSPARENTE. La fragmentación transparente, es el más alto nivel de transparencia y consistencia que el usuario o el programador de aplicación puede obtener en una relación global. Localidad transparente, es el nivel más bajo de transparencia y requerimiento de usuario o programador de aplicación para

CAPITULO I

trabajar en un fragmento en lugar de con la relación completa. Esta separación es muy conveniente, en el diseño de las bases de datos distribuidas pues la determinación de una porción relevante del dato, disminuye el problema.

2. **Explicito control de redundancia.** La arquitectura de referencia, provee de un control explícito de redundancia en un nivel de fragmentación. Por ejemplo, Si dos imágenes físicas son sobrepuestas, estas contendrán datos comunes.

3. **Independencia del DBMSs local.** A esta característica, se le llama Transparencia local de mapeo, disminuye muchos problemas del manejo de B.D. sin tomar en cuenta el modelo específico del DBMSs. En un sistema homogéneo, es posible que el esquema independiente, sea definido usando el mismo modelo de datos para el DBMSs local, Así se reduce la complejidad del mapeo.

La descomposición de una relación global en fragmentos puede desarrollarse en dos diferentes tipos de fragmentación: La **FRAGMENTACIÓN HORIZONTAL** y la **FRAGMENTACIÓN VERTICAL**. En cualquiera de los dos tipos de fragmentación, un fragmento puede ser definido por una expresión en un lenguaje relacional (álgebra relacional) que toma relaciones globales como operandos y como resultado obtenemos los fragmentos. Por ejemplo, si una relación global contiene datos sobre pacientes, un fragmento puede contener solo datos sobre pacientes con un tipo de enfermedad específica.

1.4.2.1. ESTRUCTURA DE LAS BASES DE DATOS DISTRIBUIDAS

Un sistema de base de datos distribuido consiste de un conjunto de localidades, cada una de las cuales mantiene un sistema de base de datos local. Cada localidad puede procesar transacciones locales, es decir, aquellas que sólo procesan información que reside en esa localidad. Además una localidad puede participar en la ejecución de transacciones globales, es decir aquellas que accesan información de varias localidades. La ejecución de transacciones globales requiere comunicación entre las localidades.

Las localidades del sistema pueden conectarse físicamente de diversas formas. Las distintas configuraciones se representan por medio de gráficas cuyos nodos corresponden a las localidades.

Una saliente del nodo A al nodo B corresponde a una conexión directa entre las dos localidades. En la figura 1.4.2.2. se ilustran algunas de las configuraciones más comunes. Las diferencias principales entre estas configuraciones son:

- i) **Costo de instalación.** El costo de conectar físicamente las localidades del sistema.
 - ii) **Costo de comunicaciones.** El costo en tiempo y dinero que implica enviar un mensaje de la localidad A a B.
 - iii) **Confiabilidad.** La frecuencia con que falla una línea de comunicación a una localidad.
 - iv) **Disponibilidad.** La posibilidad de acceder la información a pesar de fallas en algunas localidades o líneas de comunicación.
- Como se verá, estas diferencias juegan un papel importante en la elección del mecanismo apropiado para manejar la distribución de los datos.

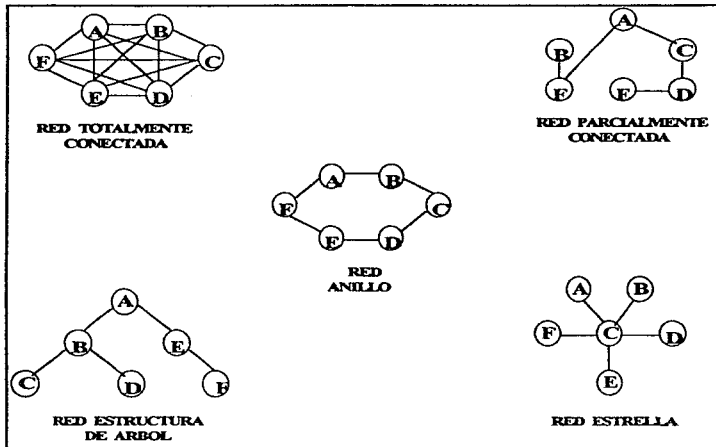


Figura L4.2.2. Topología de las redes

Las localidades de un sistema distribuido de base de datos pueden estar dispersas de manera física ya sea en una área geográficamente extensa o en una área reducida. Una red del

CAPÍTULO I

primer tipo se denomina red de larga distancia, mientras que las últimas se conocen como redes de área local.

Puesto que las localidades de las redes de larga distancia (área amplia) están distribuidas en forma física en una área geográficamente extensa, es probable que las líneas de comunicación sean relativamente lentas y menos confiables en comparación con las redes de área local.

Las líneas de comunicación de larga distancia normales son las líneas telefónicas, conexiones de microondas y canales de satélite. Por otra parte, como todas las localidades de las redes de área local están próximas entre sí, las líneas de comunicación son de más alta velocidad y menor tasa de errores que sus contrapartes en las redes de área amplia. Las conexiones más comunes son a través de cables trenzados, coaxiales de banda base, coaxiales de banda ancha y de fibras ópticas.

Estos ejemplos se ilustran por medio de un sistema bancario que cuenta con sucursales situadas en ciudades diferentes.

Cada sucursal tiene su propia computadora con una base de datos que incluye todas las cuentas que tiene esa sucursal. Así pues, cada una de estas instalaciones es una localidad. También existe una localidad única que mantiene información acerca de todas las sucursales del banco.

Una transacción local es una transacción que accesa cuentas sólo en la localidad en que se inició. En cambio, una transacción global accesa cuentas de una localidad que no es en la que se inició, o cuentas de varias localidades distintas.

Por tanto, las condiciones para que se pueda tener una base de datos distribuida son:

- a) Cada localidad debe estar consciente de la existencia de las demás.
- b) Cada localidad permite ejecutar transacciones tanto locales como globales.

CARACTERÍSTICAS.

- Los datos están dispersos en varios lugares.
- La localización geográfica de los datos es transparente para los programas que los utilizan.
- Cada nodo puede trabajar en forma autónoma
- La integridad de los datos es controlada por el DBMS.
- Debe ser independiente del hardware en que corra.

OBJETIVOS.

- Autonomía local.
- No debe depender de un nodo centralizado.
- Operación continua
- Localización independiente.
- Generación de réplicas independientes.
- Manejo de transacciones distribuido.
- Independencia entre el sistema operativo y la base de datos.
- Independencia entre la red y la base.
- Independencia entre el Administrador de la base de datos y la base de datos en si.

1.4.2.2 VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

La principal ventaja de los sistemas distribuidos de base de datos es la capacidad para compartir y acceder la información de manera confiable y eficiente. Entre las principales ventajas tenemos:

a) UTILIZACIÓN COMPARTIDA DE LOS DATOS Y DISTRIBUCIÓN DEL CONTROL

Si varias localidades diferentes están conectadas entre si, entonces un usuario de una localidad puede acceder datos disponibles en otra localidad. Si no se contara con esa facilidad, un usuario que quisiera transferir fondos de una sucursal a otra, tendría que recurrir a un mecanismo externo para realizar la transacción. Este mecanismo externo sería, de hecho, una base de datos única centralizada.

La ventaja principal de compartir los datos por medio de la distribución de la información es que cada localidad puede controlar hasta cierto punto los datos almacenados localmente. En un sistema centralizado, el administrador de la base de datos de la localidad central controla ésta. En un sistema distribuido, existe un administrador global de la base de datos que se encarga de todo el sistema. Parte de estas responsabilidades se delegan al administrador de base de datos de cada localidad. Dependiendo del diseño del sistema distribuido cada administrador local podrá tener un grado de autonomía diferente que se conoce como **autonomía local**. La posibilidad de contar con autonomía local es en muchos casos una ventaja importante de las bases de datos distribuidas.

b) CONFIABILIDAD Y DISPONIBILIDAD.

Si se presenta una falla en una localidad distribuida, es posible que las demás localidades puedan seguir trabajando. En particular, si los datos se repiten en varias localidades, una transacción que requiera de un dato específico puede encontrarse en más de una localidad. Así, la falla de una localidad no implica necesariamente la desactivación del sistema.

El sistema debe detectar cuando falla una localidad y tomar las providencias necesarias para recuperarse de la falla. El sistema no debe seguir utilizando la localidad que falló. Por último, cuando se recupere o repare ésta localidad, debe contarse con mecanismos para reintegrarla al sistema con un mínimo de complicaciones.

Aunque la recuperación de fallas es más compleja en los sistemas distribuidos que en los centralizados, la capacidad que tiene el sistema para seguir trabajando a pesar de la falla de una localidad resulta en una mayor disponibilidad. La disponibilidad es fundamental para los sistemas de base de datos que se utilizan en aplicaciones de tiempo real. Por ejemplo, si una línea aérea no puede tener acceso a la información, es posible que pierda clientes a favor de la competencia.

c) AGILIZACIÓN DEL PROCESAMIENTO DE CONSULTAS.

Si una consulta comprende datos de varias localidades, puede ser posible dividir la consulta en varias subconsultas que se ejecutan en paralelo en distintas localidades. Este cálculo en paralelo permite procesar con rapidez la consulta del usuario. En los casos en que hay repetición de los datos, el sistema puede pasar la consulta a las localidades con carga más ligera.

d) INDEPENDENCIA DE USUARIOS.

Es una demanda, que muchos usuarios ahora manejen sistemas distribuidos, que acepten más rápido la automatización que a diferencia de antes, era centralizada en un sistema en línea totalmente bajo el control y supervisión del departamento donde se encuentre el manejador. El resultado de la independencia de usuarios, es la facilidad de distribuir bajo el control de un manejador de línea local, con considerables incrementos de confidencialidad en la línea de staff y del manejador, cualquier tipo de tarea.

e) INCREMENTA LA EFICIENCIA.

Las computadoras pequeñas, se encuentran dentro de la distribución de sistemas que normalmente se dedican por lo menos en un periodo de tiempo a una sola tarea. Estos así tienen una eficiencia mayor en términos de costo por unidad o trabajo realizado si dejamos de realizar un multiproceso o multiprogramas un sistema centralizado. Ya que un pequeño grupo de

computadoras desperdician en un sistema centralizado tiempo máquina y por lo tanto tiempo hombre.

1.4.2.3 DESVENTAJAS DE LA DISTRIBUCIÓN DE LOS DATOS.

La desventaja principal de los sistemas distribuidos de bases de datos es la mayor complejidad que se requiere para garantizar una condición adecuada entre localidades. El aumento en la complejidad se refleja en:

a) **COSTO DE DESARROLLO DE SOFTWARE.** Es más difícil estructurar un sistema de base de datos distribuida y por lo tanto su costo es mayor.

b) **MAYOR POSIBILIDAD DE ERRORES.** Puesto que las localidades del sistema distribuido operan en paralelo, es más difícil garantizar que los algoritmos sean correctos. Existe la posibilidad de errores extremadamente sutiles. El arte para construir algoritmos para sistemas distribuidos, sigue siendo un campo de investigación activo e importante.

c) **MAYOR TIEMPO EXTRA DE PROCESAMIENTO.** El intercambio de mensajes y los cálculos adicionales que se requieren para coordinar las localidades son una forma de tiempo extra, que no existe en los sistemas centralizados.

Al escoger el diseño de una base de datos el diseñador, debe equilibrar las ventajas y las desventajas de la distribución de los datos.

1.4.2.4 DISEÑO DE LAS BASES DE DATOS DISTRIBUIDAS.

Esta sección se dedica a los factores de diseño específicos de las B.D. distribuidas.

Considérese una relación que se va almacenar en la base de datos, hay muchos factores que se deben de tomar en cuenta al almacenar esta relación en la B.D. distribuida, tres de ellas se mencionan a continuación:

a) **REPETICIÓN.** El sistema mantiene varias formas idénticas de la relación, cada copia se almacena en una localidad diferente, lo que resulta en repetición de información. La alternativa a la repetición es almacenar una sola copia de la relación.

Si la relación r esta repetida, se almacena una copia en dos o más localidades. En el caso extremo se tiene repetición total, en la que se almacena una copia de la relación en cada una de las localidades del sistema.

CAPITULO I

La repetición tiene ventajas y desventajas:

1.- DISPONIBILIDAD. Si falla una de las localidades que contiene a la relación, puede disponerse de esta en otra localidad. Así el sistema puede continuar procesando consultas que impliquen a r a pesar de haber fallado una localidad.

2.- MAYOR PARALELISMO. En el caso en la que la mayor parte de los accesos a la relación r resulten solo en la lectura de la relación, varias localidades podrán procesar consultas que involucren a r en paralelo. Mientras más copias de r existan, mayor será la probabilidad de que los datos requeridos se encuentren en la localidad donde se está ejecutando la transacción. Por tanto, la repetición de los datos reduce al mínimo el movimiento de información entre las localidades.

3.- MAYOR TIEMPO EXTRA PARA LAS ACTUALIZACIONES. El sistema debe asegurarse de que todas las copias de la relación r sean consistentes, pues de otra manera pueden hacerse cálculos erróneos. Esto implica que cada vez que se actualice r , la actualización debe propagarse a todas las localidades que contengan copias lo que resulta en un mayor tiempo.

a) FRAGMENTACIÓN. La relación se divide en varios fragmentos. Cada fragmento se almacena en una localidad diferente, si la relación r está fragmentada, r se dividirá en varios fragmentos, r_1, r_2, \dots, r_n . Estos fragmentos contienen información suficiente para reconstruir la relación r original, como se verá esta reconstrucción puede llevarse a cabo ya sea aplicando la operación de unión o un tipo especial de operación sobre los diversos fragmentos. Existen dos esquemas diferentes para fragmentar una relación: Fragmentación horizontal y Fragmentación vertical. La fragmentación horizontal divide a la relación asignando cada tupla de r a uno o más fragmentos, la fragmentación vertical divide a la relación descomponiendo el esquema R de la relación r de una manera especial. Estos dos esquemas pueden aplicarse en forma sucesiva a la misma relación, lo que resulta en varios fragmentos diferentes.

b) REPETICIÓN Y FRAGMENTACIÓN. Esta es una combinación de los dos incisos antes mencionados.

La relación se divide en varios fragmentos. El sistema mantiene varias copias idénticas de cada uno de los fragmentos.

La técnica que se escribió para repetir y fragmentar la información, puede aplicarse de manera sucesiva a la misma relación. Es decir un fragmento puede repetirse, las copias pueden fragmentarse, etc.

1.4.3. BASES DE DATOS RELACIONALES.

Clasificar a los sistemas de bases de datos de acuerdo con el enfoque que adoptan, elegir ¿Que estructura de datos y operadores asociados debe soportar el sistema? es algo que se debe de tomar en cuenta para la creación de los sistemas de las bases de datos, los tres enfoques mayormente conocidos son:

- El enfoque relacional,
- El enfoque jerárquico; y
- El enfoque de red

Estos tres difieren en la manera en que permiten al usuario ver y manipular las asociaciones o entidades. Cualquier sistema de Base de Datos tiene que tener la capacidad de representar y manipular entidades, en forma de registros o segmentos y las relaciones que tienen entre sí.

Del enfoque jerárquico y de red se han dado detalles más a fondo, con anterioridad, para poder entender el modelo **RELACIONAL**, volveremos con los hospitales, ya que un hospital cuenta con datos médicos, de seguros, de revisión y resultados a los estudios de reacciones químicas, cuenta con pacientes, médicos, auxiliares, farmacéuticos y mucho más, por tanto podemos explicar todos los tipos de relaciones que existen en una B.D.

Considerando el ejemplo de la figura 1.4.3.1. y recordando un poco los conceptos básicos dados con anterioridad; en el que se representa en una tabla bidimensional llamada comúnmente modelo de datos relacional o bien **relación** (para no confundir este término con la **relación entre entidades**, se le conoce como **TABLA**)

Cada columna en la tabla es un **ATRIBUTO**, los valores que las columnas contienen son conocidos como **DOMINIO**, conjunto de valores que toma cada atributo, por ejemplo, el dominio para el número de pacientes esta formado por cuatro dígitos integrados, (0000,0001,...., 0009) pero el valor actual que **PACIENTE** toma en la tabla es 1111, 1234, 2345, 4876, 5123 y 6845. Los renglones de la tabla, son llamados **TUPLAS**.

Podemos decir, que la columna de la tabla, representan elementos de datos y los renglones representan el dato grabado.

La tabla de cirujanos figura 1.4.3.1.A. representa a la entidad cirujano. la Tabla de **PACIENTE Y CIRUJANO** 1.4.3.1.B representa la relación entre el paciente y el cirujano así

TABLA DE CIRUJANO

NO. DE LICENCIA DEL CIRUJANO	NOMBRE DEL CIRUJANO
145	DAVID JIMENEZ
189	CARLOS DOMINGUEZ
243	JAVIER LLAMOSAS
311	MIGUEL GUZMAN
467	PATRICIA PERALTA

A) REPRESENTACION DE DATOS USANDO MODELO DE DATOS RELACIONAL DONDE LA LLAVE PRIMARIA ES EL NO. DE LICENCIA DEL CIRUJANO

TABLA DE PACIENTE Y CIRUJANO

NO. PACIENTE	NO. DE LICENCIA DEL CIRUJANO	FECHA DE LA CIRUGIA	CIRUGIA	MEDICAMENTO PRESCRITO
1111	145	01.01.95	APENDICITIS	PENICILINA
1111	311	12.01.95	AMIGDALAS	
1234	243	05.04.95	CATARATAS	TETRACICLINE
1234	467	10.05.95	CORAZON	CEFALFORIN
4876	145	10.05.95	LIPOSUBCION	AGUALINA
6845	243	05.04.95	CORNEAS	TETRACICLINE

B) SE OBSERVA AQUI QUE LA LLAVE PRIMARIA ES UNA COMBINACION DEL NO. DE PACIENTE EL NO. DE LICENCIA DEL CIRUJANO Y LA FECHA DE CIRUGIA, YA QUE ESTOS TRES DATOS SON UNICOS A UNA CIRUGIA.

Figura 1.4.3.1. Tipos de llave

como con la fecha de cirugia y de igual forma, en la figura 1.4.3.1.C. se representa el medicamento aplicado y el efecto que este tiene.

Una columna o grupo de columnas puede ser llamada "LLAVE" cuando es un valor único identificado por una flecha arriba de la tabla, es posible tener más de una llave, pero solo una se designara como primaria.

En la tabla de pacientes se da el número de paciente 1234, al que llamamos llave, con este número de identificación obtenemos el nombre del paciente, y su dirección, datos que son únicos a este número de paciente.

La noción de usar el modelo de datos lineal como una estructura de un sistema manejador de base de datos fue propuesto por el Dr. E. F. Cood en 1970. El proceso de depurar las entidades y sus relaciones en formatos de tabla usando el concepto relacional es llamado proceso de normalización (de la que se hablará más adelante). La teoría de normalización tiene por objeto evitar que los datos se repitan innecesariamente.

Una de las mayores ventajas del modelo relacional, es la simplicidad de este, ya que el usuario final lo comprende y maneja fácilmente con poco o ningún entrenamiento en programación, y no implica consideración alguna sobre aspectos posicionables, de apuntadores o de rutas de acceso. El usuario final no tiene que intervenir en el diseño del sistema, sin embargo entiende lo que ve, no necesita preocuparse por los detalles de la implementación física; esta habilidad se conoce como INDEPENDENCIA DE DATOS de la que ya se habló con anterioridad.

Este enfoque relacional, constituye un camino muy diferente para la descripción y manipulación lógica de datos. Se esfuerza por evitar muchas de las desventajas de los otros modelos (jerárquico y de red), que fueron utilizados para los primeros sistemas de base de datos.

Como hasta ahora se ha mencionado, el modelo de datos relacional se representa como una colección de tablas. Aunque las tablas son simples y de noción intuitiva, están en correspondencia directa entre el concepto de tabla y el concepto matemático de relación.

En años posteriores a la introducción del modelo relacional, se desarrolló una teoría para bases de datos relacionales. Esta teoría asiste en el diseño de bases de datos relacional y en el procesamiento eficiente de los requerimientos de información de los usuarios de la base de datos.

De todo lo anterior y en términos menos complicados, tenemos:

Una **relación** es una tabla BIDIMENSIONAL de n columnas constituidas por un conjunto de tuplas de n elementos. Cada una de las **columnas** en una relación es un conjunto de valores de elementos de datos (tipo de atributo o campo) al que se le llama **dominio**.

Una **relación o tabla** es un arreglo BIDIMENSIONAL con las siguientes características:

CAPITULO I

a) Cada entrada en la tabla es un elemento de datos o dato elemental; no hay grupos repetitivos. Es decir, cada dominio debe representar a una sola relación. Se dice que una relación está normalizada si no tiene grupos repetitivos.

b) A cada columna, es decir, al dominio, se le asigna un nombre diferente y está constituido por valores del mismo dato elemental.

c) Todos los renglones o tuplas, son distintas; no se permite la duplicación de datos.

d) Las hileras y columnas pueden ordenarse en cualquier secuencia en cualquier momento, sin que esto afecte el contenido de la información o la semántica aplicada.

Cada tupla debe poseer una llave que lo identifica unívocamente y lo diferencia de otros. La llave es un dominio simple o una combinación de dominios. Una llave constituida por una combinación de dominios es no redundante si ninguna entidad de la llave puede eliminarse o borrarse sin destruir la habilidad de identificar unívocamente a cada tupla. Puede existir más de un conjunto de dominios que pueden constituir una llave. A estos conjuntos se les denomina llaves candidatas. La llave primaria es el conjunto de dominios que se selecciona para identificar a las tuplas.

I.4.3.1 VENTAJAS DEL MODELO DE DATOS RELACIONAL.

1.- SIMPLICIDAD.

Al usuario final, se le presenta un modelo de datos con sus requisitos formulados sin complejidad ni aspectos orientados a sistemas. Un modelo relacional es lo que el usuario ve, pero no es necesariamente la implementación física.

2.- REQUISITOS DE NO PROCEDIMIENTO.

Dado que no hay dependencia posicional entre las relaciones, no requiere reflejar alguna estructura preferida y por lo tanto puede ser de no procedimientos.

3.- INDEPENDENCIA DE DATOS.

El modelo de datos relacional elimina los detalles de las estructuras de almacenamiento y estrategias de acceso desde la interfaz de usuario. El modelo proporciona un grado relativamente grande de independencia de datos.

4.- FUNDAMENTACIÓN TEÓRICA.

El modelo de datos relacional, esta basado en su buen desarrollo de la teoría matemática de las relaciones. El riguroso diseño de una base de datos, da a este tipo de modelo, una fundación muy sólida. Este tipo de fundamentación no existe para los otros modelos.

1.4.3.2 DESVENTAJAS DEL MODELO DE DATOS RELACIONAL

Aunque algunos sistemas manejadores de base de datos basados en el modelo de datos relacional están disponibles actualmente. Un DBMS relacional no ha alcanzado el nivel de desempeño de un modelo de datos jerárquico o de red. Ahora bien, existe una expectativa, ¿Puede un modelo de datos relacional, usado por un Sistema Manejador de Base de Datos, proporcionar el conjunto completo de capacidades operativas con la eficiencia requerida a gran escala?. Actualmente el adelanto de la tecnología está mejorando la rentabilidad y rapidez del hardware como respuesta a esta interrogante.

1.4.4. NORMALIZACIÓN

La normalización es el proceso mediante el cual un diseñador de bancos de datos puede transformar cualquier estructura de datos no plana, a un conjunto de relaciones normalizadas, es decir a un conjunto de tablas planas que no contengan grupos repetitivos.

Así como pueden transformarse las Bases de Datos de tipo red en jerárquicas mediante la introducción de cierta redundancia, se puede transformar cualquier tipo de base de datos a una de tipo relacional via la introducción de redundancia adicional.

Se dice que una relación no plana o no normalizada contendrá al menos un dominio que será en realidad otra relación. Una relación normalizada tiene únicamente dominios simples, es decir, dominios que no son a su vez otra relación.

Un archivo que sea plano excepto por un grupo repetitivo, se normalizará al quitarlo y formar con el una relación separada. Esta deberá tener un nombre propio así como un campo llave.

La figura 1.4.3.2. muestra esquemáticamente como se realiza la normalización para la base de datos del hospital con que hemos trabajado. Note que el campo llave NOMBRE DEL PACIENTE, debe repetirse en la nueva relación con respecto al tipo de medicamento que recibirá. Cabe mencionar aquí que el hecho de tener redundancia lógica de datos no necesariamente

CAPITULO I

significa que haya redundancia física, pero significa necesariamente al menos cierto costo extra para evitar la redundancia física.

La normalización, es utilizada generalmente para poder obtener el enfoque relacional que posee bases matemáticas rigurosas, mismas que respaldan la teoría relacional, proporcionando simplicidad en las estructuras de datos utilizadas, facilitando su uso y modificaciones.

Tomando el ejemplo de la figura 1.4.3.2 se explica lo siguiente:

La relación mostrada en la primer tabla, posee múltiples valores para el atributo No. de paciente que es la llave ó identificador, por lo tanto no es una relación plana.

Se dice que una relación esta en Primera Forma Normal si para cada valor específico de un identificador, existe uno y solo un valor de cada atributo. Es decir no hay "grupos repetitivos". Pero se mantiene un alto grado de redundancia.

Considerando el ejemplo anterior su conversión a una forma plana se observa en la tabla b) de la figura en cuestión.

TABLA DE PACIENTE Y CIRUJANO

<i>NO. PACIENTE</i>	<i>NO. DE LICENCIA DEL CIRUJANO</i>	<i>FECHA DE LA CIRUGIA</i>	<i>CIRUGIA</i>	<i>MEDICAMENTO PRESCRITO</i>
<i>1111</i>	<i>145</i>	<i>01.01.95</i>	<i>APENDICITIS</i>	<i>PENICILINA</i>
	<i>311</i>	<i>12.01.95</i>	<i>AMIGDALAS</i>	
<i>1234</i>	<i>243</i>	<i>05.04.95</i>	<i>CATARATAS</i>	<i>TETRACICLINE</i>
	<i>467</i>	<i>10.05.95</i>	<i>CORAZON</i>	<i>CEFALSFORIN</i>
<i>4876</i>	<i>145</i>	<i>10.05.95</i>	<i>LIPOSUBCION</i>	<i>AGUALINA</i>

a) Originalmente, un paciente se registra tantas veces como lo necesite, Ira normalización, se enlistan los datos.

<i>NO. PACIENTE</i>	<i>NO. DE LICENCIA DEL CIRUJANO</i>	<i>FECHA DE LA CIRUGIA</i>	<i>CIRUGIA</i>	<i>MEDICAMENTO PRESCRITO</i>
<i>1111</i>	<i>145</i>	<i>01.01.95</i>	<i>APENDICITIS</i>	<i>PENICILINA</i>
<i>1111</i>	<i>311</i>	<i>12.01.95</i>	<i>AMIGDALAS</i>	
<i>1234</i>	<i>243</i>	<i>05.04.95</i>	<i>CATARATAS</i>	<i>TETRACICLINE</i>
<i>1234</i>	<i>467</i>	<i>10.05.95</i>	<i>CORAZON</i>	<i>CEFALSFORIN</i>
<i>4876</i>	<i>145</i>	<i>10.05.95</i>	<i>LIPOSUBCION</i>	<i>AGUALINA</i>

b) Ira. Normatividad, cada tratamiento o enfermedad se vuelve independiente aunque pertenezca al mismo paciente.

<i>NO. PACIENTE</i>	<i>NO. DE LICENCIA DEL CIRUJANO</i>
<i>1111</i>	<i>145</i>
<i>1111</i>	<i>311</i>
<i>1234</i>	<i>243</i>
<i>1234</i>	<i>467</i>
<i>4876</i>	<i>145</i>

<i>NO. DE LICENCIA DEL CIRUJANO</i>	<i>NOMBRE</i>	<i>DIRCCIÓN ...</i>
<i>145</i>	<i>DR. REYES</i>	<i>ALZATE 3</i>
<i>311</i>	<i>DR. GATICA</i>	<i>LERDO 34</i>
<i>243</i>	<i>DR. FERRER</i>	<i>SABINO 18</i>
<i>467</i>	<i>DR. HUERTA</i>	<i>NEPTUNO 6</i>
<i>145</i>	<i>DR. NAVARRETE</i>	<i>JUAREZ 67</i>

c) 2da. y/o 3ra Normatividad. Aquí se pueden hacer independiente las tablas sin afectar la relación, es decir si se consulta el No. de paciente en una tabla, se podrá conocer los datos del cirujano, al relacionarlo con el NO. DE LICENCIA DE CIRUJANO que es un campo llave en otra de nuestras tablas que contiene los datos específicos para cada cirujano..

Para las normativas subsiguientes se pretende llevar a cabo este recorte de campos, para poder manejar las tablas con facilidad.

Figura I.4.3.2. Normatividad

PRIMERA FORMA NORMAL.

Esta es la forma más común de llegar a la Primera forma normal. La redundancia puede reducirse separándola en otro grupo aparte de la entidad debiendo contener el identificador de la entidad original, para mantenerse relacionadas.

SEGUNDA FORMA NORMAL.

Para determinar que una entidad este en segunda forma normal se requiere lo siguiente:

Que el valor de cualquier atributo que no es llave dependa de todos los atributos que forma la llave. Siguiendo el ejemplo NO. DE CIRUGÍA no depende de los atributos que forman al identificador o llave, sino que se convierte en un campo llave en otra tabla sin afectar su movimiento en la tabla en cuestión.

CAPITULO I

Para normalizar una entidad en la Segunda Forma, se crea una nueva entidad de los atributos que dependen parcialmente de una llave (arriba) siendo parte del identificador de esta nueva entidad el atributo del cual depende para mantenerla relacionada con la original.

TERCERA FORMA NORMAL

Se dice que una entidad se encuentra en Tercera Forma Normal, si el valor de cada atributo depende de toda la llave y no de cualquier otro que no lo sea.

En esta forma normal, se buscan los atributos que están dependiendo de otro que no es una llave.

Para poner una entidad en Tercera Forma Normal, se crea una entidad con los atributos que no dependen de ningún atributo que forma la llave, siendo el identificador de la nueva entidad el atributo del cual era dependiente.

1.4.4.1. OBJETIVOS DE LA NORMALIZACIÓN.

- Eliminar en lo posible todos los datos que mantengan anomalías.
 - Conservar toda la información
 - Maximizar la flexibilidad
- 1.- La estructura debe ser tal que haya lugar para todos los datos requeridos.
 - 2.- La redundancia que pueda existir deberá ser causada por los elementos que son identificadores o llaves. Por lo que se debe elegir aquellos que no estén sujetos a actualizaciones.
 - 3.- Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.
 - Anomalia de Inserción: No cabe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.
 - Anomalia de eliminación: La eliminación de un solo registro puede ocasionar la eliminación de toda la ocurrencia de una entidad.
 - Anomalia de actualización: Si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentra definido.
- 3.- La capacidad de adaptabilidad ante los cambios, maximizan la independencia de uso particular de los datos.

La normalización requiere tres o más acciones sobre un atributo de una entidad, reciben el nombre de :

Primera forma normal, segunda forma normal y tercera forma normal.

Y así sucesivamente hasta que el modelo alcance su última etapa de no redundancia, por lo regular la normalización está comprendida hasta la tercer etapa, donde el modelo ya alcanza un estado manejable y entendible para el usuario y el programador.

Esto se establece, en las dependencias funcionales que existan entre los atributos y los dominios particulares a esa relación. Las dependencias funcionales las determina directamente el significado o la semántica del contenido de la base de datos según la interpretación del diseñador de ésta.

1.4.5. DISEÑO E IMPLEMENTACIÓN DE UNA BASE DE DATOS.

El diseño de una base de datos es el proceso de arreglar en una estructura organizada los campos de datos necesarios para una o más aplicaciones.

Esta estructura debe contemplar las relaciones necesarias entre los campos y al mismo tiempo debe adaptarse a las restricciones del sistema de manejo de datos que se este utilizando. Existen dos partes es este proceso: El de DISEÑO LÓGICO y el DISEÑO FÍSICO.

El DISEÑO LÓGICO es un ejercicio independiente de la implementación que se lleva a cabo en los campos y las relaciones necesarias para una o más aplicaciones.

El DISEÑO FÍSICO depende de la implementación y toma los resultados del diseño físico y lo define de acuerdo a las características del sistema y del manejo de datos que se este utilizando.

Existen muchas razones que hacen necesario el diseño de la base de datos. Estos incluyen la no redundancia de datos, el funcionamiento de la aplicación, independencia de datos, seguridad y facilidad de programación. Todos estos factores son importantes en el ambiente del procesamiento de datos, y todos pueden ser afectados cuando el diseño no es el apropiado.

La mayoría de los involucrados en el diseño de la base de datos concuerdan en que existen dos fases separadas en el proceso del diseño de la base: el diseño de una estructura lógica que pueda ser procesada por el manejador de la base (DBMS) y que describa el punto de vista del usuario; y la selección de la estructura física, tal como los métodos de acceso.

Según Novak existen cuatro componentes básicos necesarios para la metodología de diseño de bases de datos:

- Un proceso de diseño estructurado que consiste en una serie de pasos donde se escoge una alternativa de varias presentadas.

CAPITULO I

- Técnicas de diseño para realizar la enumeración requerida y criterios de evaluación para seleccionar alternativas en cada paso.
- Requerimiento de información de entrada para el proceso de diseño como un todo y para cada paso del diseño.
- Un mecanismo descriptivo para representar las entradas de información y los resultados en cada paso del diseño.

La tecnología actual de base de datos, muestra muchos efectos residuales de su crecimiento desmedido a partir de los métodos de diseño de archivos de un solo registro.

El diseño de archivos depende del programa de aplicación ya que los datos han sido definidos y estructurados en términos de aplicaciones individuales que los usan. La llegada del DBMS da énfasis al enfoque de diseño de datos y programas. El concepto de la base de datos integrada con atención a muchos usuarios fue el resultado directo de las capacidades de estructuración de datos complejos que podría manejar el DBMS. Los datos pueden ser vistos ahora como un recurso, en lugar de ser cercano a los programas, y por lo mismo debe tener la orientación y los requerimientos integrados en lugar de estar orientados a los programas.

Es una tarea difícil, para todos los usuarios, lograr un nivel aceptable de funcionamiento de la base de datos.

El diseñador de la base de datos debe estar siempre consciente de los efectos de costo/beneficio asociados con múltiples usuarios en una sola base de datos integrada. El ahorro de espacio de almacenamiento y la aplicación expandida de la base de datos en la toma de decisiones deben ir acompañadas por un análisis crítico de la degradación del servicio para algunos usuarios, ésta debe ser evitada en lo posible ya que la meta debe ser un desempeño aceptable para todos los usuarios.

Otro aspecto del diseño de la base de datos es la flexibilidad. Las bases de datos que están dedicadas completamente a las aplicaciones actuales pueden ser muy limitadas en ciertas empresas.

Los requerimientos de cambio constantes y los nuevos elementos de datos pueden provocar que el mantenimiento de los programas sea costoso, la proliferación de archivos temporales y un desempeño cada vez peor. El diseño de la base de datos debe tomar en cuenta la integración y la flexibilidad.

I.4.5.1. ELEMENTOS DEL PROCESO DE DISEÑO.

Las principales clases de entradas y resultados del proceso de diseño son los siguientes:

Entradas:

- Requerimientos de información general.
- Requerimientos de procesamiento.
- Especificaciones del DBMS.
- Configuración de hardware y sistema operativo.
- Especificaciones del programa de aplicación.

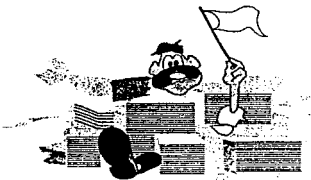
Resultados:

- Estructura lógica de la base de datos (presentación al usuario)
- Estructura de Almacenamiento (diseño físico)

Los requerimientos de información general representan las descripciones de varios usuarios de la organización para la cual los datos son reunidos, los objetivos de la base de datos, y las vistas de los usuarios de los cuales los datos deben ser colectados y almacenados en la base de datos. Estos requerimientos son considerados como independientes de los procesos porque no están atados a ningún sistema de manejo de datos específico o aplicación. El diseño de bases de datos basado en estos requerimientos es considerado ventajoso a largo plazo para las bases de datos que deben ser adaptables a los requerimientos cambiantes del procesamiento .



CAPITULO II



HERRAMIENTAS DE DISEÑO

CAPITULO II. HERRAMIENTAS DE DISEÑO

II.1. SERVIDORES DE RED.

Un servidor es una computadora compartida por múltiples usuarios de una red. A medida que evolucionan las redes y en virtud del crecimiento en los requerimientos de recursos, cada vez es más necesario que los procesos se deleguen a servidores de propósito específico. Su papel es muy importante, tanto a nivel de comunicaciones, como organizador y coordinador del trabajo al integrar redes de área local, compartir recursos, atender procesos, transferir archivos y mensajes, y permitir en general enlaces locales y remotos.

Anteriormente todas las actividades de computación se cargaban a un gran procesador que tenía tareas múltiples o podía manejar varias operaciones a la vez. Este mismo manejaba la lectura y escritura en el disco, en una cinta de respaldo, corría todas las bases de datos, proporcionaba el servicio de procesamiento de palabras y coordinaba el flujo de información hacia todas las terminales de trabajo enlazadas. Este procesador dividía sus recursos entre cada una de tales tareas.

Actualmente con la ayuda de servidores para redes todas estas actividades no se ejecutan en un solo procesador grande, sino que se efectúan en conjunto en una red de computadoras con un procesador más pequeño. Los procesos que en el pasado se combinaban en una mainframe, se dividen y se distribuyen en computadoras individuales; algunos ejemplos son: la impresión, las bases de datos y el manejo de archivos.

Los servidores se pueden clasificar en los de propósito general y en los de propósito específico. A continuación se describen las funciones más importantes de éstos:

Servidores de archivos. Su principio de operación es permitir el acceso al disco desde las estaciones de trabajo en la red. La tarea principal de la máquina dedicada es ofrecer entrada-salida (lectura-escritura) de archivos para los usuarios coordinando todo acceso a los controladores de disco de la red. Un proceso de servidor que requiera de un equipo dedicado se denomina usualmente como esquema cliente-servidor.

Servidor de impresión. Su función básica es coordinar, supervisar y ejecutar la impresión, ordenando los trabajos en colas de impresión, así como asignarlos a las diferentes impresoras con que cuenta la red.

Servidor de bases de datos. Se ha advertido una evolución en este tipo de servidores, ya que tradicionalmente han sido máquinas dedicadas (con mucho espacio en disco) que contienen toda la base de datos. El motor de la base de datos, las tablas y las herramientas de investigación residen en el servidor, mientras que toda o parte de la interfaz del usuario se encuentra en la estación de trabajo del cliente.

Servidores de comunicaciones. Estos ofrecen una puerta hacia y desde la red local para la entrada y salida de comunicaciones sincronas y asincronas, módem, computadoras Host o centrales y otras redes, siendo éstas sólo algunas de sus posibilidades de conexión.

Servidores de administración. Se refiere a aquellos que recopilan, analizan y procesan información para segmentos individuales de red. La información basada en el servidor, puede **accesarse** a través de consolas de administración de ubicación central, donde el administrador de la red puede analizarla. Se utilizan para la continua supervisión del tráfico de la red.

Servidores de correo electrónico. Estos servidores, algunos los han denominado "oficinas electrónicas de correo" controlan todo el correo entre los usuarios de la red.

Los servidores por lo regular funcionan dentro de la arquitectura llamada **cliente-servidor**; la cual es una aplicación al método de diseño, que resulta en la descomposición de un sistema de información dentro de un número pequeño de servidores con diferentes funciones ejecutando las tareas en una o más plataformas de hardware, los cuales dan servicio a un gran número de clientes conectados a través de una o varias plataformas de hardware.

Plataformas cliente-servidor. La plataforma es la combinación de software y hardware sobre la cual el sistema C-S DBMS (Sistema de Administración de base de datos, Cliente-Servidor) trabaja. Los servidores con el sistema SQL (Standard Query Lenguaje) (Lenguaje Normalizado de Consulta) pueden funcionar bajo cualquiera de las siguientes tres plataformas: Pcs, Unix workstations y super-servidores.

Los diferentes sistemas de hardware varían grandemente de acuerdo a sus características y capacidades. Sin embargo tienen ciertas características comunes en cuanto a requerimientos del sistema operativo. El sistema operativo es el software primitivo, el cual actúa como una interface entre el hardware y las aplicaciones que funcionarán en él; las aplicaciones son usualmente escritas para funcionar en un sistema operativo particular.

Computadoras Personales (Pcs) Pcs son los tipos de computadoras más usadas en la actualidad, pero apenas en los recientes años han tenido una aceptación como plataforma de los sistemas C-S. Los sistemas basados en los microprocesadores 80386 y 80486, con discos duros del rango de gigabytes de tamaño y sistemas operativos multitareas hacen de las Pcs competidores de las workstation basadas en la tecnología RISC y minicomputadoras las cuales han sido la plataforma tradicional para los sistemas C-S DBMS.

Workstation RISC. Las workstation (estaciones de trabajo) basadas en la tecnología RISC (Reduce Instruction Set Computing) (Computadora con Conjunto Reducido de Instrucciones) fueron inicialmente utilizados para aplicaciones científicas y de ingeniería, este tipo de tecnología utiliza microprocesadores usualmente más rápidos que los de 80x86 de Intel.

La mayoría de workstation RISC utiliza una variante del sistema operativo Unix; en cuanto a su apariencia y operación son semejantes a las Pcs, actualmente es una plataforma muy importante dentro de los sistemas C-S; ya que el incremento de uso de los sistemas C-S ha llevado a diversos fabricantes de workstation a diseñar explícitamente servidores de propósito específico como son los servidores de bases de datos, ejemplos de este tipo de servidores son la Sun 470 y la IBM RS-6000.

Mainframes. Las mainframes son las computadoras de propósito general más poderosas disponibles en la actualidad, soporta múltiples procesadores de alta velocidad, enorme cantidad de espacio en disco duro y cientos o miles de usuarios simultáneos. Las mainframes ofrecen el máximo de seguridad en cuanto a seguridad de datos y hardware se refiere.

A diferencia de los sistemas pequeños de computación las mainframes requieren de ambientes controlados, incluyendo control constante de temperatura, pisos especiales y estar en secciones completamente aisladas.

Este tipo de computadoras es accesada a través de terminales o a través de Pcs con terminales emuladoras, los cuales son conectados a los controladores de las terminales (los cuales son subsistemas que proveen conectores para comunicaciones de red), y así son conectadas a la mainframe.

El poder de procesamiento y rapidez de este tipo de máquinas se debe principalmente a la capacidad de trabajar con múltiples CPU, discos duros de alta rapidez y una rápida comunicación

entre los diferentes elementos del sistema. Las mainframes por lo regular contienen desde 256 MB de memoria RAM hasta gigabytes.

A continuación se muestra una conexión común de un servidor en una red de tipo Ethernet.

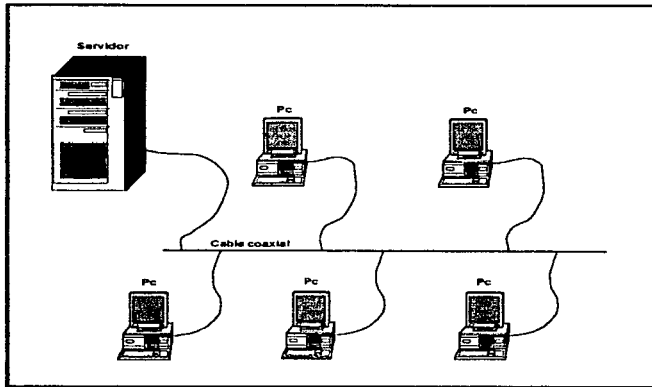


Figura II.1.1 Ejemplo de un servidor de red.

II.1.1. SERVIDOR HP9000 SERIE 827.

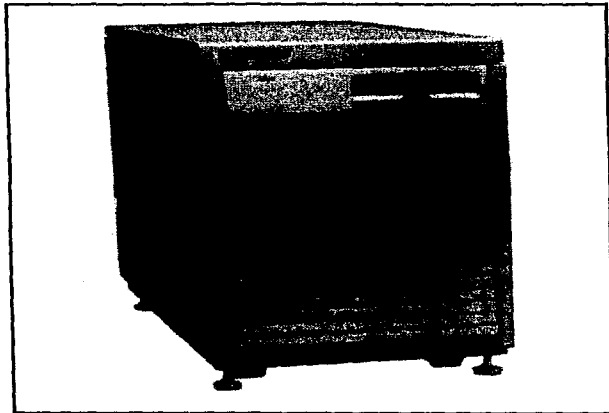


Figura II.1.1.1 Servidor HP9000 827s

DATOS TÉCNICOS.

El sistema de computador HP 9000 serie 827 (HP9000 827s) está construido con la tecnología llamada PA-RISC (Precision Architecture-Reduced Instruction Set Computing) (Arquitectura de Precisión-Conjunto de Instrucciones Reducido de Computadora), lo cual garantiza una mayor simplicidad en su construcción, ya que es implementada con un menor número de componentes y da una mayor capacidad de ejecución en comparación con los sistemas más antiguos de computadoras basadas en la tecnología CISC (Complex Instruction Set Computer).

CAPITULO II

Las características más sobresalientes de la tecnología PA-RISC son las siguientes:

- Conjunto de instrucciones reducido (140 instrucciones).
- Instrucciones tipo **pipelining** (Proceso de encauzamiento).
- Tamaño fijo de instrucción.
- Pocos accesos a memoria.
- Ejecución de la instrucción en un solo ciclo de reloj.
- Formato fijo de instrucciones de 32-bit.
- Direccionamiento a memoria virtual de 48 bits ó 64 bits.

Todos los sistemas HP 9000 series 800 tienen un código objeto compatible ya que todos los procesadores están basados en la misma arquitectura, es decir, la PA-RISC y usan el mismo conjunto de instrucciones.

Hay dos tipos de máquinas HP9000 con la arquitectura HP-PA, las que poseen la arquitectura HP-PB (HP-Precision Bus) y los que poseen la arquitectura CIO (Channel I/O), la siguiente figura muestra un diagrama de bloques de cada una de estas arquitecturas.

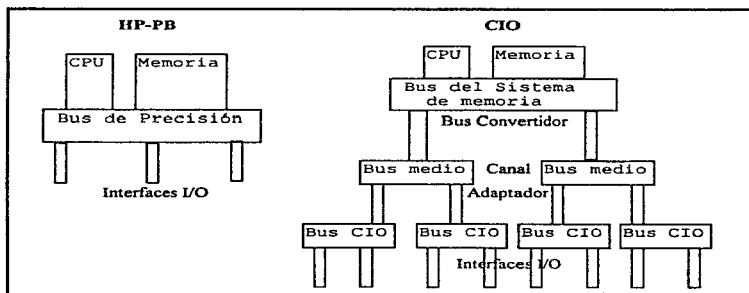


Figura II.1.1.2 Diagrama de bloques de arquitecturas HPB y CIO

Los sistemas con arquitectura HP-PB son los modelos de la familia 8x7S, la familia 8x2S y los modelos 808S y 815S. Los sistemas con la arquitectura CIO incluyen a las familias 8x5S y 6x5SV.

Las máquinas CIO tienen una estructura jerárquica de bus. Dependiendo del punto jerárquico donde se encuentren los buses serán más rápido. El bus medio es más rápido que el bus CIO.

Las máquinas HP-PB tienen solamente un tipo de bus, el bus de precisión, es decir, las máquinas HP-PB no requieren de buses convertidores.

Los componentes de las arquitecturas CIO y HP-PB son descritas a continuación.

SPU Es la Unidad Procesadora del Sistema (System Processing Unit) consiste de todos los componentes del hardware que se encuentran dentro del gabinete de la computadora excluyendo periféricos tales como disco y el drive (manejador) de cinta.

CPU Unidad de Procesamiento Central (Central Processing Unit), en este caso el término se refiere al módulo de procesamiento de instrucciones de la computadora, no a la computadora en sí.

CIO Canal de Entrada/Salida (Channel Input/Output), se refiere al hardware de la arquitectura del sistema I/O (entrada/salida), de los modelos 8x5S, 6x5SV.

HP-PB Bus de Precisión HP (HP-Precision Bus), se refiere al hardware de I/O de los modelos 8x7S y 8x2S así como los modelos 808 y 815 bus.

BUS Un bus es una pieza de hardware la cual acarrea datos y señales entre módulos de hardware dentro del sistema.

Bus Medio El bus medio (también conocido como bus central) existen solamente en las máquinas que tienen la arquitectura CIO. El bus medio es un bus de alta velocidad sobre el cual se comunican la memoria y el CPU.

Bus CIO El bus CIO existe solamente en las máquinas que tienen arquitectura CIO. El bus CIO es un bus I/O de propósito general (de baja velocidad en comparación con el bus medio), sobre el cual los dispositivos I/O se comunican unos con otros y con el canal adaptador.

Bus de precisión El bus de precisión es encontrado en las máquinas que poseen la tecnología HP-PB. El de precisión es el único bus en los sistemas HP-PB (es decir, no se necesita canal

CAPITULO II

adaptador ni bus convertidor). El CPU, memoria y dispositivos I/O son comunicados con cada uno de los otros dispositivos por medio del bus de precisión.

Canal adaptador El canal adaptador funciona como un convertidor entre el bus medio y el bus CIO, de tal manera que la memoria y CPU puedan comunicarse con los dispositivos I/O.

Dispositivo adaptador Es la tarjeta en la cual un periférico se conecta. Algunas veces se refiere a la tarjeta CIO (máquinas CIO); una tarjeta HP-PB ó un adaptador I/O (En máquinas IIP-PB).

Los modelos HP 9000 807S/817S/827S son sistemas de nivel de entrada; todos estos sistemas usan el sistema HP-PB y están integrados con un procesador, un manejador de disco y un dispositivo de cinta para la recepción de datos digitales en acuerdo con el estándar europeo de tarjetas, el HP-PB está diseñado para acomodar dos tarjetas de entrada y salida. Una ranura de expansión HP-PB es típicamente ocupada por una tarjeta LAN, dejando las demás ranuras listas para usarse para otras tarjetas I/O. El sistema 827S tiene un total de 6 HP-PB ranuras de expansión que pueden soportar el mismo número de tarjeta.

A continuación se muestran algunas diferencias en cuanto a memoria de la familia HP9000 8x7S:

Modelo	Disco Duro	Disco Interno	Máximo Disco Externo	Número de Ranuras	Mem. Base
807S	234 MB	2 ó 1	10.9 GB SCSI	2	8 MB
817S	328 MB	2 ó 1	28.5 GB SCSI	2	16 MB
837S	328 MB	2 ó 1	28.5 GB SCSI	2	16 MB
827S	328 MB	4 ó 2	47.7 GB SCSI	6	16 MB
847S	677 MB	4 ó 2	47.7 GB SCSI	6	32 MB
857S	677 MB	4 ó 2	47.6 GB SCSI	12	64 MB

Tabla II.1.1.1 Capacidad de memoria de la familia 8x7

A continuación se muestra una tabla con las capacidades máximas de memoria de la familia HP9000:

HP 900 series 800	807	817	827	837	847	857	822	832
Memoria Máxima (MB)	192	192	192	192	192	128	256	256
Disco Duro (GB)	11	29	48	29	48	48	24	24

Tabla IL1.1.2 Capacidad máxima de memoria de la familia HP9000

En la actualidad el sistema HP9000 827s existente en la UAM-Xochimilco tiene una capacidad de 128 Mb en RAM.

TARJETAS I/O (Input-Output) (Entrada-Salida) PARA LOS MODELOS HP9000 8x7S.

Estas tarjetas pueden ser puestas en las ranuras de expansión HP-PB y son las siguientes:

- Tarjeta MUX de 8 ó 16 canales. Soporta terminales, impresoras, plotters, modems y puertos de acceso.
- Tarjeta SCSI/Centronics. Soporta discos SCI, impresoras y drives de CD-Rom.
- Tarjeta HP-IB. Soporta discos, drives de cinta, CD-Rom, impresoras y plotters.
- Tarjeta LAN. Para IEEE 802.3 y red Ethernet.
- Tarjeta X.25
- Tarjeta SNA
- Tarjeta PBA-FL
- Tarjeta MAP 3.0.

A continuación se muestra una lista de dispositivos I/O comúnmente utilizadas en las Series 800:

I/O Interface	Máximos Dispositivos en una Interface	Máximo Largo de Cable de Cable (Metros)	Mezclado de Dispositivos en Interface
SCSI	7 Dispositivos SCSI	6	Si
HP-IB	4 Discos HP-IB 4 Plotters HP-IB 4 Impresoras HP-IB 8 Dispositivos (No discos)	1.25 K	No con Disco
HP-FL ó PBA-FL	8 Discos FL	500	Discos solamente
MUX 16 Canales	16 Dispositivos Seriales	5 ó 15	Si
MUX 8 Canales	8 Dispositivos Seriales	5 ó 15	Si
MUX 6 Canales	6 Dispositivos seriales	5 ó 15	Si
Centronics	1 Impresora Centronics	2	No

Tabla II.1.1.3 Dispositivos I/O utilizados en la familia HP9000 serie 800

ARCHIVOS DE DISPOSITIVOS (Device Files).

El sistema HP 9000 827S con sistema operativo Unix (HP-UX) se comunica con los periféricos (impresoras, terminales, drives, modems, etc) a través de archivos llamados archivos de dispositivos (Device files). Antes de que el sistema HP-UX pueda comunicarse con un periférico,

este debe de tener su propio archivo de dispositivo. Por ejemplo cada terminal tiene su propio archivo de dispositivo a través del cual el sistema HP-UX escribe datos (los cuales aparecen en el monitor de la terminal) y lee datos (podría ser a través del teclado del usuario).

Un archivo de dispositivo no contiene datos, es decir, un archivo de dispositivo específica de que forma el sistema HP-UX se va a comunicar con el periférico. Para tener una idea más concisa de lo anteriormente explicado se muestra el siguiente diagrama de bloques:

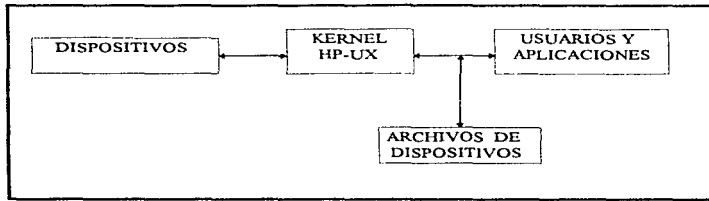


Figura II.1.1.2 Diagrama de flujo de un sistema HP-UX

Todos los dispositivos de entrada/salida pueden ser clasificados como dispositivos de bloque o dispositivos de carácter.

Los dispositivos de bloque son las unidades de entrada/salida en bloque, discos ó cintas que hallan sido montados posteriormente en el sistema.

Los dispositivos de carácter son los discos ó cintas que no hallan sido montados con anterioridad en el sistema, terminales, impresoras, módem.

Lo anterior es importante ya que dependiendo de la clasificación del periférico se le asignará un archivo de dispositivo los cuales también se dividen en los de bloque y carácter.

Los archivos de dispositivos de **bloque** transfieren datos en bloques a través del sistema de buffers de tipo cache para agilizar las transferencias de entrada/salida.

Los archivos de dispositivos de **carácter** usualmente transfieren bloques de datos de diferentes tamaños, difieren de los archivos de bloque en que no utilizan ningún tipo de sistema de

buffers de cache. Esto se refiere a que los periféricos utilizados por este tipo de archivos, no han sido montados posteriormente al sistema.

A continuación se muestra un diagrama de bloques de lo anteriormente explicado:

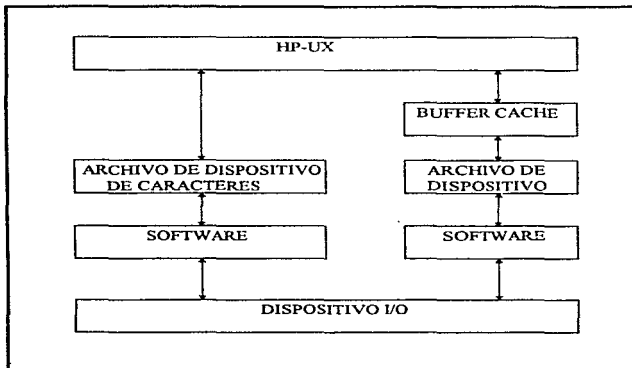


Figura IL1.1.3 Diagrama de flujo de la transferencia de archivos

A continuación se muestran las características principales de del servidor HP9000 827s:

Instrucciones (MIPS)	13
Instrucciones en punto flotante (MFLOPS)	1.1
Memoria RAM	16-192 Mbytes
Ancho de banda (I/O)	25 Mbytes/s
Slots disponibles	6
Almacenamiento interno en disco	328 Mbytes - 2.68 Gbytes
Máximo almacenamiento (con HP-IB)	8 0/48 Gbytes
Tecnología de circuitos integrados	CMOS
Frecuencia de instrucción	13 Mhz
Rapidez de reloj en instrucción	70 ns
Tamaño de cache	64 Kbytes
Conexiones tipo MUX	48

Tabla II.1.1.4 Características generales del Servidor HP9000 827s

A continuación se muestra las características técnicas del servidor HP9000 827s:

Especificaciones eléctricas.	Nominal	Rango	Corriente
	Entrada voltaje AC-Frecuencia	100-120V, 50/60 Hz 200-240V, 50/60 Hz	90-132VAC/47-63Hz 180-264VAC/47-63Hz
Especificaciones de Ambiente			
Temperatura	Operando: 5° C a 40° C No operando: -40° C a 65° C		
Máximo cambio de temperatura	20° C / hora		
Humedad relativa	Operando: 20% a 80% no condensado No operando: 5% a 80% no condensado		
Máximo cambio de humedad	30% / hora		
Altitud	Operando: 3.0 Km No operando: 15.0 Km		
Características Físicas			
Dimensiones	Altura:	750 mm	
	Ancho:	375 mm	
	Profundo:	600 mm	
Peso		90 Kg	

Tabla II.1.1.5 Características técnicas del Servidor HP9000 827s

II.2 SISTEMA OPERATIVO UNIX 9.04

Para explotar las posibilidades del hardware en una computadora en comunicaciones, almacenamiento de datos y tratamiento de la información, el software de la aplicación requiere algún tipo de supervisor que pueda manejar los detalles de la gestión de los recursos del software, del acceso a los archivos y de la interacción con los usuarios. Estas funciones de supervisión son el trabajo del **Sistema Operativo** que, en este caso es el sistema operativo UNIX 9.04 y que en adelante llamaremos **HP-UX** (Versión para la plataforma HP-9000).

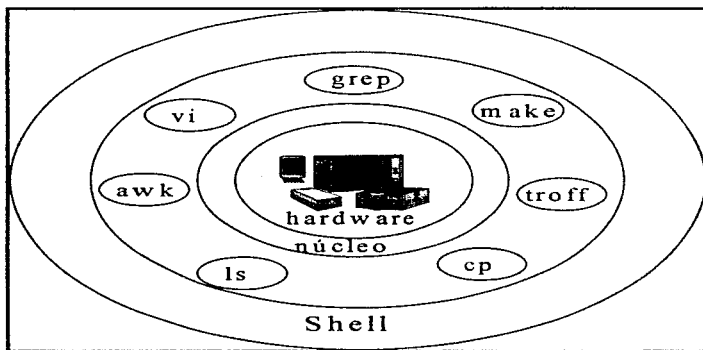


Figura II.2.1 Estructura del HP-UX.

El HP-UX es un versátil y poderoso sistema que puede reconocer diversos grupos de usuarios. Se puede utilizar para correr aplicaciones simples, o desarrollar aplicaciones propias en este rico ambiente de desarrollo de software. Además, HP-UX ofrece poderosos ambientes para subsistemas tales como correo electrónico, ventanas, trabajo en red y gráficas. Este sistema se compone de tres partes principales. De éstas, el Sistema de Archivos y el Shell (armazón) son visibles al usuario.

En tanto que el Kernel (núcleo) no es realmente visible, pero es igualmente importante a nivel general. A continuación se muestra un resumen de estas tres partes del sistema: El Kernel es el centro del software de HP-UX y es parte del sistema operativo que actúa directamente con el hardware. Es la parte del sistema que gestiona los recursos de la computadora, ocupándose de los discos, cintas, impresoras, terminales, líneas de comunicación y cualquier otro dispositivo. El kernel también maneja funciones de bajo nivel para el sistema operativo y aísla el sistema operativo de los detalles del hardware. La figura II.2.2 muestra el kernel y su relación con el ambiente del HP-UX:

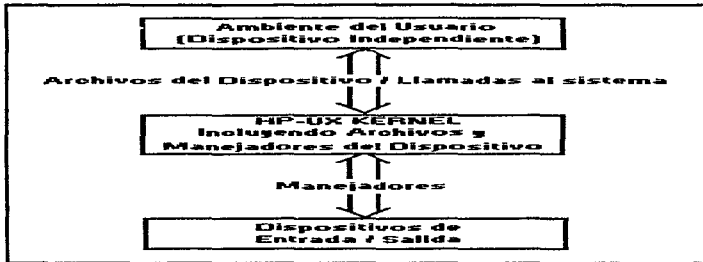


Figura II.2.2 El Kernel en relación con el ambiente del usuario y los dispositivos de entrada salida.

El kernel reconoce cualquier hardware por dos elementos que son el driver (manejador) de dispositivos y la información contenida en el dispositivo de archivos. Para que el kernel pueda buscar y encontrar cualquier dispositivo de hardware, se debe incluir el driver de dispositivos apropiado para el kernel y se debe instalar el dispositivo de archivos apropiado para el sistema de archivos. En cualquier momento que se modifique el archivo de configuración del kernel, se está creando un nuevo kernel y es necesario reinicializar el sistema para que este reconozca los cambios. La figura II.2.3 muestra que se tiene cuando se genera un nuevo kernel en la serie 800.

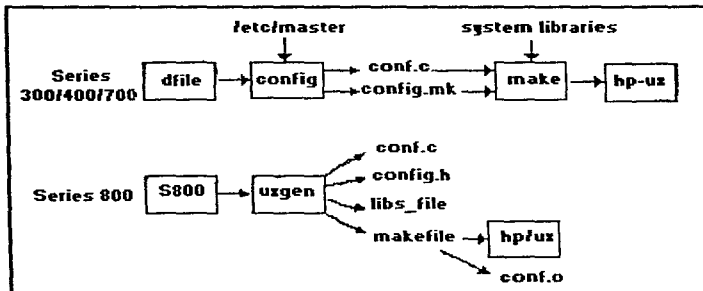


Figura II.2.3 Archivos de Configuración del Kernel.

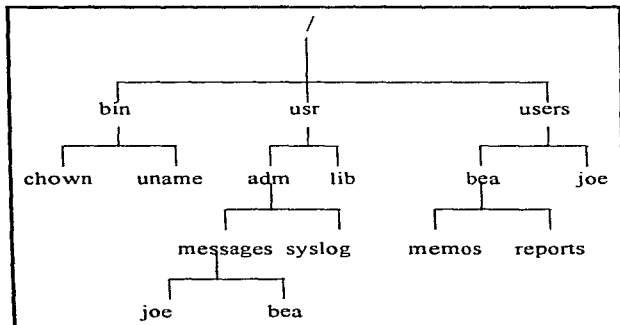


Figura II.2.4. Estructura típica de un Sistema de Archivos.

- **El Sistema de Archivos** se refiere integralmente al sistema de archivos (todos los archivos del sistema) de HP-UX, los cuales están organizados en una estructura de árbol. Este sistema de archivos es jerárquico, comienza solamente con el directorio raíz ("/") y después se va ramificando hacia abajo. En la figura 11.2.4 se muestra una estructura típica de sistema de archivos.

Los tipos principales de sistemas de archivos que utiliza el HP-UX son el HFS o High Performance System (Sistema de Alto Rendimiento), el NFS o Network File System (Sistema de Archivos de Red), el CDFS, o CD-ROM File System (Sistema de Archivos de CD-ROM).

HFS: Reside físicamente en los dispositivos de almacenamiento masivo, generalmente en los drives de disco duro.

NFS: Es un sistema de archivos HFS remoto al cual se puede acceder mediante una red y se puede acceder en un sistema de archivos local.

CDFS: La información en el CD es virtualmente permanente; se pueden leer datos del CD, pero no se puede escribir sobre este. Un CDFS puede alojar fácilmente grandes cantidades de información que no requieren modificaciones.

Los LVM o Logical Volume Manager (Manejador de Volúmenes Lógicos), nos abre la posibilidad de crear grupos de discos (volúmenes físicos) llamados grupos de volúmenes. Se puede subdividir el espacio de un grupo de volúmenes en volúmenes lógicos.

- Un volumen lógico es como una sección de disco en que se puede tener un sistema de archivos o espacio de **swap** (intercambio). Pero un volumen lógico es una partición "blanda" de espacio de disco; se puede definir el tamaño del volumen lógico (las secciones de disco tienen tamaño fijo) y se puede incrementar el tamaño de un volumen lógico existente (no podemos cambiar el tamaño de una sección de disco).

Podemos expandir el tamaño de un volumen lógico, si es necesario, hasta tener el tamaño de un disco.

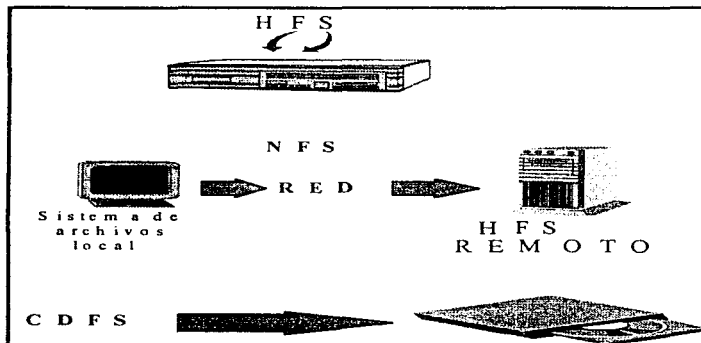


Figura IL2.5 HFS, NFS y CDFS.

- El Shell es el intérprete de órdenes. Aunque el shell es sólo un programa de utilidad, y no forma parte propiamente del sistema, es la parte que ve el usuario. El shell atiende la terminal del usuario y traduce sus peticiones en acciones a realizar por el kernel y otros muchos programas de utilidad. El HP-UX ofrece diversos tipos de shell que pueden ejecutarse. Cada uno de esos shells tienen diferentes características, y pueden aumentar la velocidad y eficiencia con que se interactúa con el HP-UX y se pueden construir las características de shell que uno desee.

Usando comandos simples se puede determinar el shell que queremos ejecutar, cambiar de shell temporal o permanentemente. Cuando inicializamos el sistema, el shell define automáticamente un ambiente de trabajo. Este ambiente define características tales como quien somos, donde estamos trabajando y que procesos estamos corriendo. Nuestro shell mantiene este ambiente hasta que nos salimos. Las características de nuestro ambiente están definidas por los valores asignados a las variables del ambiente. Estas variables consisten en un nombre y un valor.

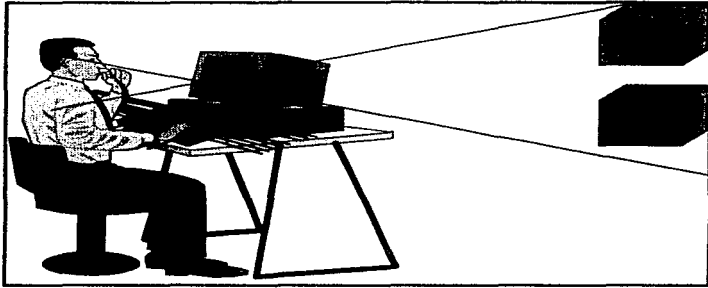


Figura IL2.6 Función del Shell.

Nuestro shell utiliza tanto variables del ambiente como del shell mismo para definir nuestro ambiente. Nuestro shell usa variables de ambiente y los pasa a todos los procesos y subshells que hemos creado. Las variables de shell son conocidas sólo por nuestro shell actual y no pasan hacia los subshells.

Para un uso más efectivo de nuestra terminal, HP-UX debe conocer el tipo de terminal o despliegue gráfico que utilizemos. Si no se proporciona un tipo de terminal, el valor por default es `TERM=hp`. El comando `tset` nos proporciona las características de las terminales.

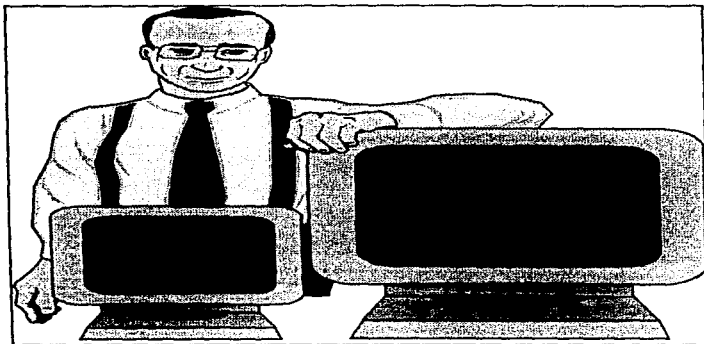


Figura II.2.7 El comando `tset` nos ayuda a conocer las características de las terminales.

El HP-UX es un sistema interactivo. Esto significa que el usuario escribe las órdenes, el sistema las interpreta, devuelve la respuesta apropiada y espera una nueva orden.

Además, es HP-UX es un sistema multitarea, es decir, el sistema puede realizar varias tareas -llamadas procesos- al mismo tiempo. La característica multitarea significa que se le pueden dar al sistema una o más tareas para hacer en modo de prioridad subordinada y mientras se puede seguir con otra cosa distinta, sin tener que esperar a que las otras tareas se terminen.

El sistema HP-UX también es un sistema operativo **multiusuario**, lo cual significa que puede trabajar más de una persona al mismo tiempo.

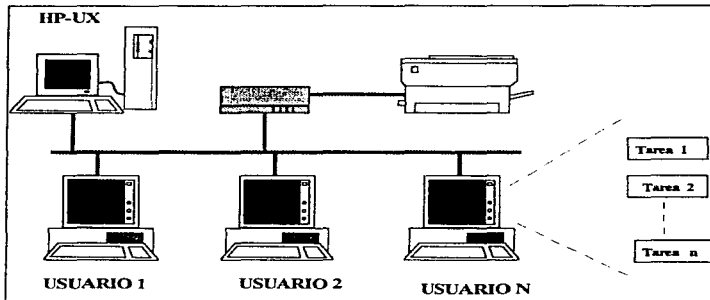


Figura IL2.8 HP-UX es un ambiente multiusuario.

EL MANEJADOR DEL SISTEMA DE ADMINISTRACION.

El manejador del sistema de administración o SAM por sus siglas en inglés (System Administration Manager), es una interfaz de usuario para desarrollar rutinas de administración de tareas sin usar los comandos fundamentales de HP-UX.

SAM es un auxiliar en tareas para las siguientes áreas:

- Contabilidad de usuarios.
- Grupos de usuarios.
- Mantenimiento de la seguridad del sistema.
- Sistema de archivos.
- Configuración de espacio de swap.
- Aumentar o disminuir periféricos.
- Cola de impresión.
- Respalda y recuperar archivos.
- Configurar conexiones de red.

- Administrar sistemas remotos.
- Configurar el Kernel del HP-UX.

SAM tiene dos interfaces de usuario, un sistema de interfaz X Window y una interfaz de terminal de texto. Ambas interfaces difieren en la apariencia de la pantalla y las interacciones de teclado/mouse.

SAM nos proporciona ayuda en línea para ayudarnos cuando necesitemos información adicional.

SAM permite la administración de puntos simples de sistemas remotos para ejecutar SAM en sistemas remotos y desplegarlos en la máquina local. La administración de un punto simple se refiere a la administración de múltiples sistemas remotos de una localidad. No hay restricciones de hardware. Se puede usar en cualquier HP-9000 Series 300/400/700/800 para administrar otra HP-9000 Series 300/400/700/800.

PROCESOS.

Un proceso es la ejecución de un programa, manejado por componentes del sistema tales como el horario y el manejo de memoria de los subsistemas. Un proceso puede estar formado de texto, datos y **stack** (pila) (un lugar en el kernel donde se depositan las partes de un programa como un proceso). Existen dos **stack** asociados con un proceso, el **stack** del kernel y el **stack** del usuario. Después de que se introduce un comando de línea, ocurre lo siguiente:

1. - El shell interpreta el comando y busca en el disco hasta encontrar el programa requerido.
2. - El control se transfiere del shell al HP-UX.

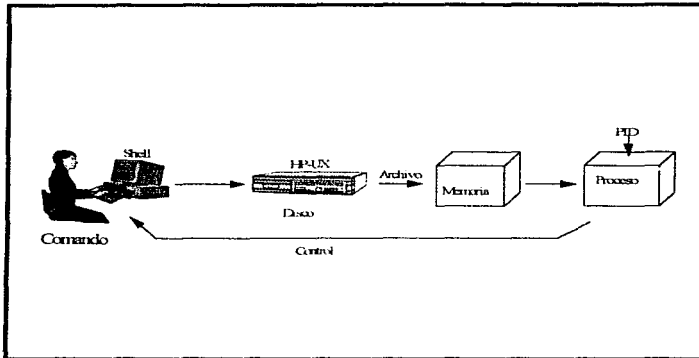


Figura II.2.9 Ciclo de vida de un proceso.

- 3.- HP-UX copia el programa especificado de un archivo de disco a la memoria. Cuando el programa reside en memoria, comienza la ejecución - se ha creado un proceso.
- 4.- A cada proceso se le asigna un **identificador de proceso o PID**.
- 5.- Cuando finaliza la ejecución del programa, el control regresa al shell y el proceso desaparece.

Los procesos mantienen una relación jerárquica padre-hijo. Cualquier proceso tiene sólo un padre, pero puede tener muchos procesos hijos. Los procesos pueden crear procesos, que a su vez, crean más procesos. Las relaciones entre esos procesos pueden ser representadas por una gráfica de árbol. Un proceso hijo es inherente en el ambiente del proceso padre. En la figura II.2.10 se presenta un ejemplo de las relaciones entre los procesos P1, P2, P3, P4, P5 y P6.

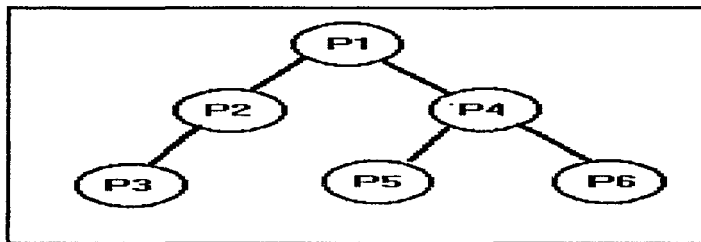


Figura II.2.10: Gráfica de árbol de una relación de procesos.

El sistema de control de procesos interactúa con el sistema de archivos cuando se leen archivos de la memoria antes de ejecutarlos. En ocasiones varios procesos pueden pertenecer al mismo programa.

Un proceso se comunica con otro proceso a través de la memoria compartida o llamadas del sistema.

El principio de distribución del tiempo del CPU se llama *time-slice* (porción de tiempo). Se le asigna el *time-slice* a los procesos de acuerdo a su prioridad. El rango de prioridades va de 0 (mayor prioridad) a 256 (menor prioridad), y se clasifican de acuerdo a las necesidades.

Cada proceso abre tres "archivos" estándar: entrada estándar (*stdin*: el lugar donde el programa espera leer la entrada. Por default es el teclado), salida estándar (*stdout*: el lugar donde el programa manda la salida. Por default es la pantalla), y error estándar (*stderr*: el lugar donde el programa escribe los mensajes de error. Por default es la pantalla.).

La figura II.2.11 ilustra las relaciones entre estos archivos y el proceso.

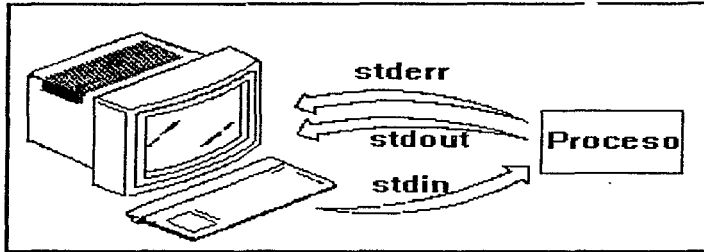


Figura II.2.11 Entrada, salida y error estándar.

NIVELES DE EJECUCION.

Un nivel de ejecución es un estado del sistema en que un set específico de procesos está disponible para ejecutarse. Este set de procesos está definido en el archivo `/etc/inittab` para cada nivel de ejecución. Se pueden definir hasta seis niveles de ejecución.

Los niveles `s` y `2` están predefinidos: `s` es el nivel de administración y `2` es el nivel de operación normal. Por lo regular no se necesitan definir los otros niveles.

Nivel de ejecución 2: Es el nivel que se llama automáticamente por default.

Nivel de ejecución 0: Es un nivel de ejecución especial reservado para la instalación de sistema. No se debe trabajar en este nivel.

Nivel de ejecución s: Es el nivel de ejecución especial reservado para el sistema administrador de tareas.

MANEJO DE MEMORIA.

La memoria es un almacenamiento de datos de alta velocidad, implementada por varios dispositivos de hardware en el sistema HP-UX.

Los datos e instrucciones de cualquier proceso deben estar disponibles para el CPU residiendo en una **memoria física** en tiempo de ejecución. La memoria RAM (Random Access Memory) o memoria principal, es compartida por todos los procesos.

El término "manejo de memoria" se refiere a todas las reglas que gobiernan la memoria física y virtual y que logran que se tenga un eficiente manejo de memoria compartida entre los recursos y los procesos del sistema.

La memoria virtual nos permite ejecutar un proceso llevando a la memoria principal sólo las partes del proceso que sean necesarias, es decir, sobre la demanda. Desalojando las partes del proceso que no han sido usadas recientemente. Complejos algoritmos de conmutación y paginación son los que determinan cuando los datos y el código del actual programa en ejecución deben regresar de RAM a disco. Los algoritmos de memoria intermedia tratan de minimizar el acceso a disco tanto como sea posible.

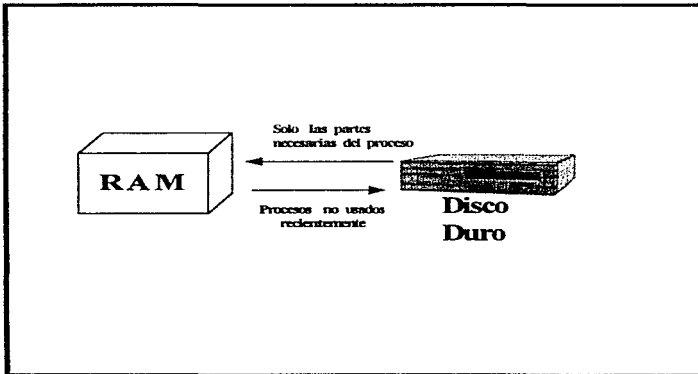


Figura IL2.12 Memoria virtual.

Existen otras dos características de la memoria física involucradas en la inicialización del sistema: memoria disponible y memoria lockable (reservada).

Memoria disponible.

No toda la memoria física está disponible a los procesos de usuarios. El kernel siempre reside en memoria principal ocupando 1.6 MB aproximadamente. La cantidad de memoria principal no reservada para el kernel es llamada memoria disponible y es usada para ejecutar procesos.

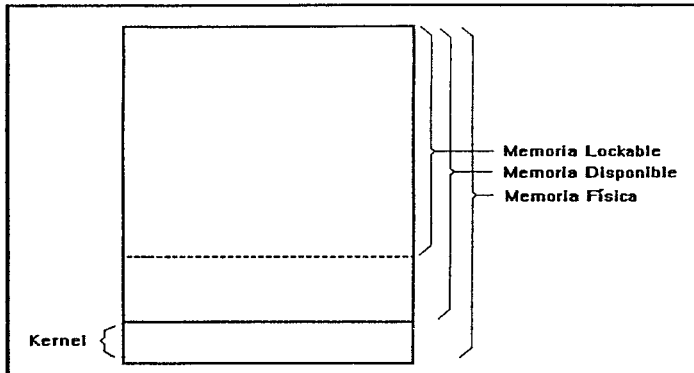


Figura II.2.13: Memoria física disponible para los procesos.

Memoria lockable.

Como se muestra en la figura II.2.13, la memoria física puede ser "reservada" (es decir, se guardan páginas en memoria para la duración del proceso) por el kernel y es llamada memoria lockable.

CAPITULO II

La memoria lockable no puede ser paginada o conmutada (swap). Generalmente, guarda con frecuencia el acceso a los programas o a estructuras de datos, tales como las secciones críticas del código aplicado.

Por default, la memoria lockable no puede ser de más de tres cuartos de la memoria disponible.

Este tipo de memoria es usada ampliamente en los ambientes de "tiempo real", como hospitales donde algunos procesos requieren respuesta inmediata y debe estar completamente disponible. Si se reduce el tamaño de esta memoria, el sistema está más expuesto a un **deadlock** (bloqueo).

Almacenamiento Secundario.

La memoria principal almacena los datos requeridos por el programa en ejecución. Los archivos de programas son guardados en memoria o almacenamiento secundario, generalmente discos que pueden ser accedidos por los buses del sistema o por red.

Una forma de almacenamiento secundario transitoria es llamada **swap** (conmutación), los datos para cada implementación de UNIX son manejados por los recursos de memoria física para mover procesos enteros entre memoria principal y almacenamiento secundario.

Librerías Compartidas.

Una librería compartida es una colección de subrutinas comunes ubicadas en una localidad de memoria compartida y que puede ser invocada de manera dinámica en tiempo de ejecución por los programas que las necesiten.

Una librería compartida reduce la cantidad de memoria ocupada por el código durante la ejecución, dado que sólo existe una copia de estas rutinas en memoria.

Cuando se incluye una llamada a una librería archivada, por otro lado, el código de la librería es copiado al archivo ejecutable del proceso en tiempo de liga.

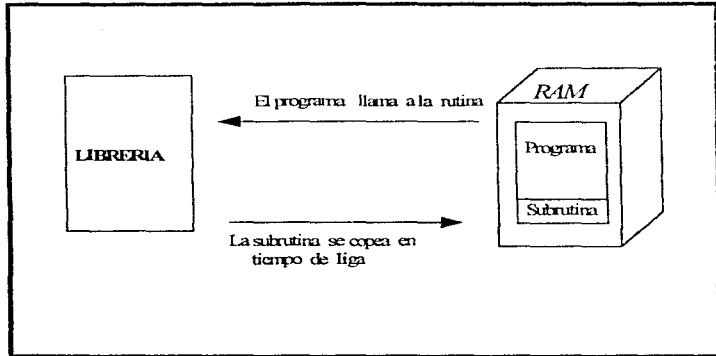


Figura II.2.14 Uso de librerías compartidas.

Manejo de Espacio de Swap.

El espacio de swap es un área de alta velocidad de almacenamiento, reservada para el uso del sistema de memoria virtual para conmutar y paginar procesos.

Hay dos tipos de espacio de swap:

- **Espacio de swap para dispositivos** está localizado en una partición. El swap de dispositivos es más rápido que el swap del sistema de archivos que se verá más adelante. En un sistema con una partición de disco tradicional, el espacio de swap de dispositivos es de tamaño fijo, en un sistema implementado con un Manejador Lógico de Volúmenes, el volumen lógico de swap puede incrementarse de acuerdo a las necesidades que se tengan.
- **Espacio de swap para el sistema de archivos** está localizado en un sistema de archivos. Este es pequeño, pero varía en tamaño de acuerdo con la actividad del sistema de swapping.

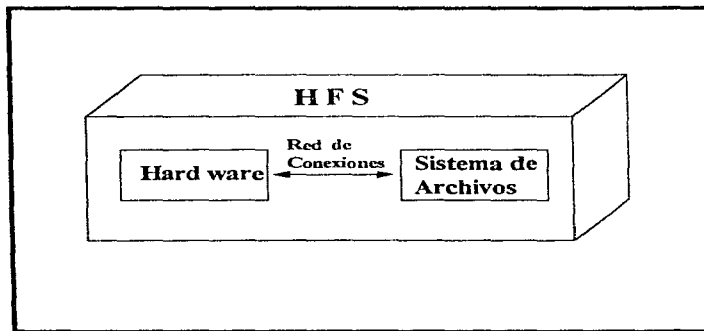
EL SISTEMA DE ARCHIVOS DE ALTO RENDIMIENTO.

HP-UX utiliza un sistema de archivos llamado Sistema y Archivos de Alto Rendimiento o HFS por sus siglas en inglés (High Performance File System), que también es conocido como el Sistema de Archivos McKusick (o BSD).

Para trabajar efectivamente con el sistema de archivos, es necesario conocer las interrelaciones con los discos físicos. Los manejadores de dispositivos en el sistema operativo permiten la comunicación con el disco.

En forma conceptual, el crear un sistema de archivos implica:

- Tomar el ambiente físico disponible para el sistema de archivos (el dispositivo de discos).
- Crear la propia entidad de software (el sistema de archivos).
- Establecer la "red de conexiones" entre los elementos físicos y de software.



Transferencia de Archivos Entre HP-UX y Otros Sistemas.

No todas las computadoras utilizan el sistema de archivos HP-UX. Por ejemplo, entre los productos HP, existen las series 200 HP-UX 2.x que usa el sistema de archivos Bell System III (BFS), y las series 500 HP-UX utilizan el Formato de Directorios Estructurado. Para acoplar las variantes entre ellos, HP-UX contiene diversas utilidades y servicios para transferencia de archivos, incluyendo otras marcas de sistemas operativos.

TRABAJO EN RED

Con el **Networking** (Red) podemos transferir datos a través de diferentes sistemas. Las redes pueden ser simples (como por ejemplo con máquinas del mismo tipo) o poderosas (con máquinas de diferentes fabricantes y con distintos sistemas operativos cada una de ellas).

II.3. RED ETHERNET.

INTRODUCCIÓN

El centro de investigación Palo Alto, de la corporación Xerox empezó a desarrollar un sistema LAN que es una tecnología de Red de Area Local de paquetes conmutados (paquet switched). Conocido como Ethernet experimental. Mas tarde la corporación DIGITAL, INTEL y XEROX en conjunto, crearon una norma Ethernet que es compatible con el estándar IEEE 802.3. Esta popular red fue inventada a principios de los años setenta

Ethernet es un ambiente de comunicación entre microcomputadoras. Este tipo de red es el que probablemente mas industrias de iniciativa privada, usan; además de; fabricas, sector educacional, sector gobierno y científico. Ethernet se puede utilizar con diversas opciones de cableado.

Este tipo de redes utiliza una topología de bus lineal con un protocolo de acceso CSMA/CD. En este tipo de red cada estación se encuentra conectada bajo un mismo bus de datos, es decir las computadoras se conectan a la misma línea de comunicación y por esta transmisión los paquetes de información viajan hacia el servidor y/o los otros nodos.

Cada estación se encuentra monitoreando constantemente la línea de comunicación con el objeto de transmitir o recibir sus mensajes. Si la línea presenta tráfico en el momento que una estación quiere transmitir, la estación espera un periodo muy corto (milisegundos) para continuar monitoreando la red.

Si la línea esta libre. la estación transmisora envía su mensaje en ambas direcciones por la red. Cada mensaje incluye una identificación del nodo transmisor hacia el nodo receptor y solamente el nodo receptor puede leer el mensaje completo. Cuando dos estaciones transmiten sus mensajes simultáneamente una colisión ocurre y es necesaria una retransmisión. Ya que el nodo aún está monitoreando, éste es capaz de detectar la colisión e intentará nuevamente la transmisión del mensaje. El protocolo incluye las reglas que determinan cuanto tiempo tendrán que esperar los nodos o estaciones para realizar sus envíos nuevamente.

La velocidad de transmisión de Ethernet es de 10 MBPS.

DEFINICIÓN DE ETHERNET.

Ethernet es una arquitectura que da un servicio con determinación de errores pero no corrección de errores. Es una red de multiacceso que utiliza un medio pasivo. Sin un control central. Las unidades de datos que se transmiten en la red alcanzan cada estación. y cada estación es responsable de reconocer la dirección contenida en una unidad de datos y aceptar estas unidades con una dirección del medio de transmisión.

II.3.1. ARQUITECTURA.

Ethernet esta constituido por cable coaxial de 1/2 pulgada de diámetro y hasta 500 metros de largo. Para aumentar su resistencia se le agrega una capa de polietileno relleno entre el hilo central y el protector de metal en cada terminación para evitar que se reflejen las señales eléctricas. Este cable es conocido también bajo el nombre de cable pasivo por si mismo, esto significa que todos los componentes electrónicos que hacen que la red funcione están asociados con computadoras que están conectadas a la red.

La red Ethernet puede extenderse con unos dispositivos de hardware llamados repetidores que entregan las señales eléctricas de un cable a otro.

Cada repetidor puede conectar dos segmentos de cable estándar Ethernet hasta 500 metros. instalando 99 transceivers Ethernet. Las funciones del repetidor son las de reciclar, amplificar y repetir todas las señales que recibe de un segmento de cable y pasar la señal al siguiente segmento.

Existen dos formas de conexión de repetidores, la local y la de fibra óptica. La local conecta dos segmentos de cable de bandabase de Ethernet por medio de cables transceivers y de transceivers H4000. La versión de fibra óptica interconecta un cable con un enlace a fibra óptica que debe ser conectada con otro repetidor de fibra óptica o un puente LAN de fibra óptica 100.

Las conexiones hechas al Ethernet son por medio de **taps (derivaciones)**. En cada tap, un pequeño orificio en la capa externa del cable permite que los pequeños pines toquen el hilo central a través de un conector tipo vampiro y el escudo trenzado (algunos conectores manufacturados requieren que el cable sea cortado y se inserte una "T"). Cada conexión a un Ethernet tiene dos componentes electrónicos principales: un **transceiver** que se conecta al hilo central y al escudo trenzado en el Ethernet, percibiendo y mandando las señales a través del cable. Un **host interfaz** se conecta al transceiver y se comunica con la computadora generalmente a través del bus de la computadora.

Un **transceiver** es una pequeña pieza de hardware que físicamente se encuentra por lo general, adyacente al Ethernet. Este hardware analógico percibe y controla el Ethernet, un transceiver contiene circuitería digital que le permite comunicarse con una computadora digital. El transceiver percibe cuando el Ethernet está en uso, y puede traducir señales eléctricas analógicas a forma digital. El cable transceiver que va del transceiver al host interface, está diseñado de una forma tal que pueda realizar dos funciones simultáneas: operar las señales y controlar esta operación.

El **host interface** controla la operación de un transceiver de acuerdo a las instrucciones que recibe de la computadora. Al sistema operativo de la interface es un dispositivo de entrada salida que acepta instrucciones transferidas de datos básicos de la computadora, irrumpe cuando la tarea ha sido completada y reporta la información del estatus. Mientras que el transceiver es un dispositivo de hardware relativamente simple, el host interface es un dispositivo mucho mas complejo, que en algunas ocasiones utiliza microprocesadores para controlar transferencias.

TIPOS DE CABLES BANDA BASE.

Existen tres tipos de cable en Bandabase : el estándar, el Thin Wire (hilo delgado) y el Twisted-Pair o par trenzado.

CAPITULO II

El tipo de cable estandar de Ethernet es recomendado para equipos donde se reciben señales via satélite y pisos de un edificio. Este cable es llamado también "backbone"(columna vertebral). Este cable es conectado a un transceiver utilizando un cable especial de hilo conductor. El transceiver esta sujeto sobre el cable Estándar de Bandabase sin cortar el cable o interrumpir la transmisión de datos.

Un máximo de 100 transceivers pueden ser usados en un segmento de cable Estándar. Un segmento de cable solo puede tener hasta 500 metros de largo (1640 pies).

Segmentos de cables múltiples pueden unirse mediante conectores barrel (barril). También múltiples segmentos de un cable transceiver pueden unirse mediante conectores slide-latch en las partes finales del mismo.

Este tipo de cable Bandabase de Ethernet que es el tradicional cable coaxial puede tener hasta 500 metros de largo con 100 nodos conectados a el. Esto implica que la distancia máxima entre conexiones es de 5 metros.

El tipo de cable Thin Wire de Bandabase de Ethernet que es otro tipo de cable coaxial permite tener hasta 30 nodos conectados a un cable de hasta 185 metros de largo. En este caso la distancia máxima en la que puede haber conexiones es de aproximadamente 6.1 metros.

Este tipo de cable. Thin Wire es relativamente barato, utiliza conectores BNC, también trabaja ha una velocidad de 10Mb/s. Podemos argumentar que esta diseñado especialmente para cableado horizontal debido a su facilidad en la instalación.

El tipo de cable Twisted-Pair de Bandabase de Ethernet, también llamado unshield Twisted-Pair por Digital Co., tiene mayor compatibilidad, debido a que posee un tipo especial de adaptadores.

DETECCIÓN DE COLISIONES

DETECCIÓN DE COLISIÓN EN EL SISTEMA DE TRANSMISIÓN

Generalmente es posible para el sistema de transmisión reconocer una colisión. Ya que compara la señal inyectada con la señal recibida.

Para asegurar la detección propia de las colisiones cada transmisor adopta un procedimiento que hace seguro que todas las partes reconozcan que una colisión a ocupado su lugar.

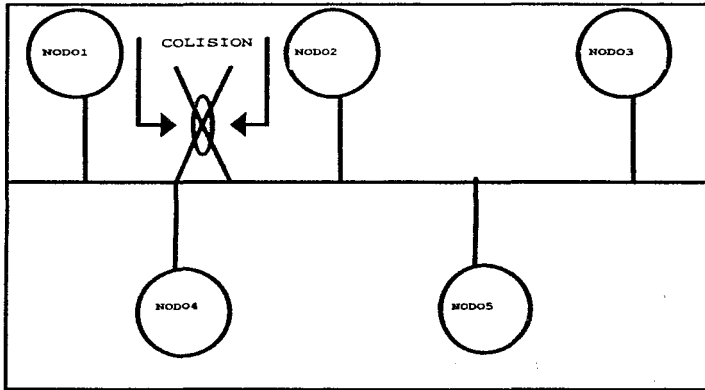


Figura 11.3.1.1 Se genera una colisión.

DETECCIÓN DE COLISIÓN EN EL CONTROL.

Alternativamente el control puede reconocer una colisión comparando la señal recibida o la sección del receptor se puede probar para reconocer la colisión desde que aparecen frecuentemente como fases de violación.

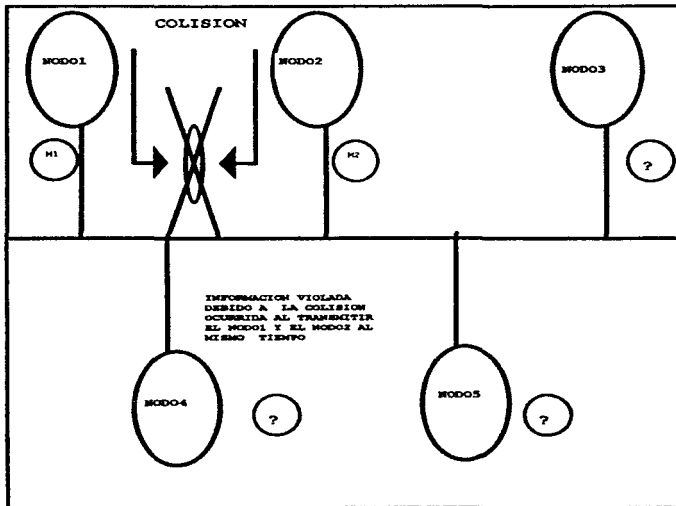


Figura II.3.1.2 Detección de información corrupta.

GENERACIÓN Y CHEQUEO.

El transmisor genera un ciclo de redundancia, o CRC de cada paquete transmitido y lo añade a otro paquete antes de la transmisión. El receptor checa el CRC en los paquetes que recibe, y los desaloja antes de dar el paquete a la estación. Si el CRC es incorrecto hay dos opciones:

--Descartar el paquete

--Entregarlo dañado con un estatus apropiado indicando un error CRC

El CRC solo protege un paquete del punto en el cual el CRC es generado para el punto en el cual es checado. El CRC no puede proteger un paquete del daño que ocurre en partes del control. La red Ethernet experimental utiliza un CRC de 16 bits . La red de especificación Ethernet detecta mejor errores con un CRC de 32 bits.

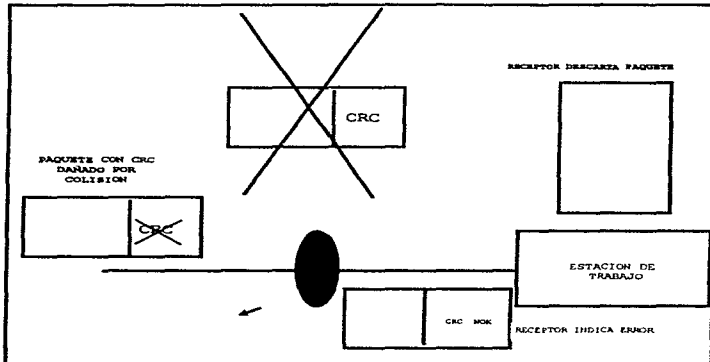


Figura II.3.1.3 Chequeo de colisión.

DIRECCIONAMIENTO.

El formato de paquete incluye una dirección de origen y una de destino. Un diseño de red local puede adoptar cualquiera de las dos estructuras básicas de dirección:

- La red especifica la estación de direcciones: En la cual las estaciones son asignadas a la red de direcciones que debe de ser única.

- Estación única de direcciones: Consistiendo en que cada estación le es asignada una dirección que es única sobre todo en espacio y tiempo.

Tales direcciones son conocidas como absolutas o universales.

La red Ethernet experimental utiliza un campo de 8 bits para las direcciones de origen y destino. Mientras que la red de especificación Ethernet utiliza un campo de 48 bits para las direcciones.

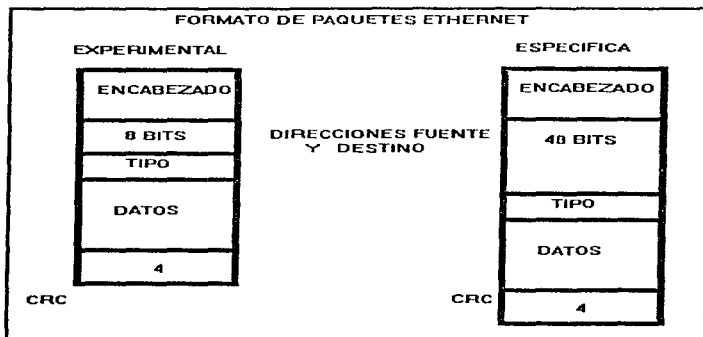


Figura II.3.1.4. Campo de direcciones en paquetes Ethernet.

CSMA/CD (Carrier Sense multiple Access Colisión Detection) CANAL DE DIRECCIÓN.

Una mayor parte del control pertenece a la dirección del canal de Ethernet. Estas convenciones especifican los procedimientos por los cuales son transmitidos los paquetes y después son recibidos en el canal de multiacceso.

TRANSMISIÓN.

El transmisor es invocado cuando el transmisor tiene un paquete para enviar. Si ocurre una colisión, el control ejecuta la colisión con un giro apropiado, cierra el transmisor y programa una retransmisión

Las políticas de retransmisión tienen dos objetivos:

- Programar una retransmisión rápidamente para mantener el paquete y mantener el uso del canal.
- Hacerlo regresar voluntariamente para reducir la carga de la estación en un canal ocupado.

Ambas generaciones de Ethernet usan el algoritmo binario Back Off (Regresar) . El Back Off se usa para calcular el retraso antes de la retransmisión. Después de que una colisión toma lugar el objetivo es obtener los periodos de retraso que reprogramarán cada estación en tiempos cuantizados en pasos por lo menos tan largos como una colisión de intervalo.

Estas cuantizaciones de tiempo se les llama la apertura de tiempo de retransmisión. Para garantizar el rápido uso del canal, esta apertura debe ser corta para evitar colisiones.

Para controlar el canal y mantenerlo estable con una alta carga, el intervalo es doblado con cada colisión sucesiva, y de esta manera se extiende la clasificación de posibles retrasos de retransmisión.

Para evitar retrasos indebidos y respuestas lentas, y mejorar las características del canal, el doblamiento puede ser detenido en algún punto. Esto es mostrado como un binario truncado exponencial del Back-Off:

El binario truncado exponencial Back-Off, aproxima el algoritmo ideal, donde la probabilidad de transmisión de un paquete es de $1/q$; donde q es el número de estaciones probadas para transmitir.

La transmisión de intervalo es truncada cuando " q " llega a ser igual al número máximo de estaciones.

El algoritmo Back-Off reinicia con un intervalo de retransmisión CERO, para la transmisión de cada paquete nuevo. Se seleccionó este algoritmo en particular porque tiene el comportamiento básico apropiado y porque permite una implementación simple.

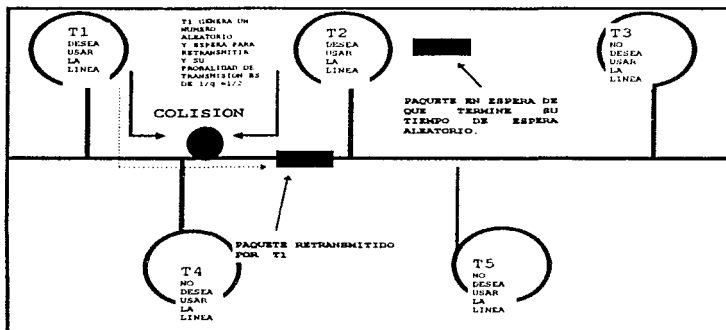


Figura IL3.1.5. Retransmisión de Paquetes.

RECEPTOR.

La sección del receptor es activada cuando el cargador aparece en el canal. El receptor procesa el flujo de bits que llega de la siguiente manera:

Primero el preámbulo existido es removido. Si el flujo de bits termina antes de que se complete el preámbulo, se asume que es el resultado de una colisión breve, y el receptor es reiniciado. En seguida el receptor determina si el paquete esta direccionado a el. El controlador aceptara un paquete en cualquiera de las siguientes circunstancias:

- La dirección destino concuerda con la dirección específica de la estación.
- La dirección destino debe tener una distinción si el destino es conocido o publico.
- Si la dirección destino es de un grupo "Multicast" (multi canal) el cual la estación es miembro.
- La estación tiene el control de un nodo promiscuo y recibe todos los paquetes.

Algún diseño del controlador podría elegir para recibir el paquete entero antes de invocar el procedimiento de reconocimiento de dirección. Esto es factible pero consume memoria y recursos de procesamiento en el controlador.

Más típicamente el reconocimiento de dirección se encuentra en un bajo nivel en el controlador, y si el paquete no es aceptado el controlador puede ignorar el resto del paquete. Asumiendo que la dirección es reconocida, el receptor ahora acepta el paquete completo.

Antes de que el paquete sea liberado hacia la estación, se verifica el CRC y son llevados a cabo otros chequeos de consistencia. Por ejemplo el paquete podría terminar en un byte o palabra apropiada, o ser de la longitud mínima apropiada.

Sin embargo las colisiones sobre el canal pueden producir paquetes cortos dañados llamados fragmentos de colisión. Generalmente es innecesario reportar estos errores a la estación, puesto que pueden ser eliminados (con un filtro de fragmentos en el controlador).

No obstante es importante para el receptor ser reiniciado rápidamente después de haber recibido un fragmento de colisión, puesto que el emisor del paquete puede estar cerca de retransmitir.

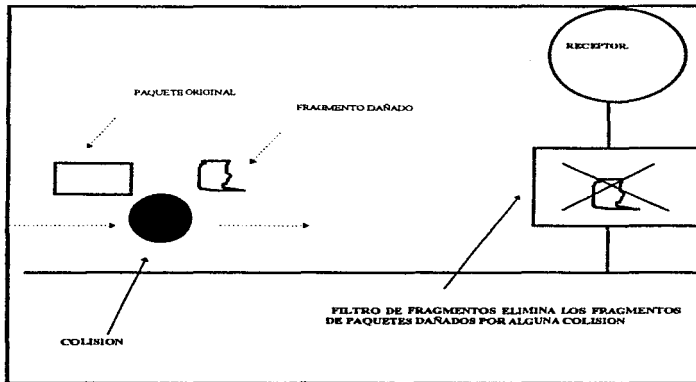


FIG. IL3.1.6. ELIMINACION DE FRAGMENTOS DE PAQUETES DAÑADOS.

LONGITUD DEL PAQUETE.

Un objetivo importante de la red Ethernet es la transferencia de datos. Esto significa que el campo de datos de un paquete puede contener cualquier patrón de bits y ser de cualquier longitud, desde "0" hasta un número arbitrariamente largo. En la practica se permiten cualquier patrón de bits para que aparezcan en el campo de datos, pero existen algunas consideraciones que sugieren una imposición de límites superior e inferior en su longitud.

En un extremo un paquete vacío (campo de datos de longitud cero), podría consistir justamente de un preámbulo, direcciones fuente y destino, un campo de tipo y un CRC.

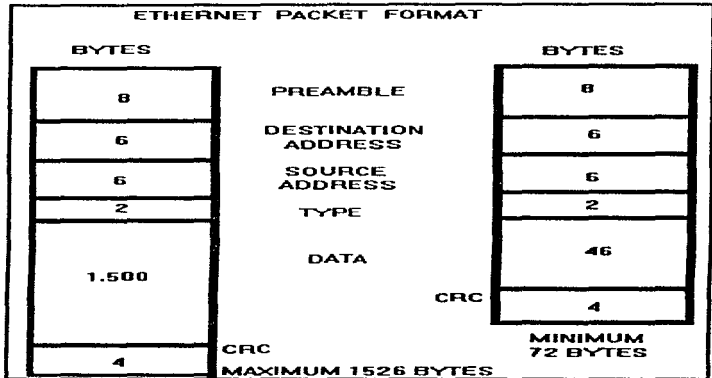


Figura II.1.7. Paquetes Ethernet.

La Ethernet experimental permite paquetes vacíos. Sin embargo en algunas situaciones es deseable forzar un mínimo de tamaño de paquete completo mandando un campo de datos de longitud mínima, como la especificación Ethernet. Los protocolos de mas alto nivel que desean transmitir paquetes mas cortos deben entonces rellenar el campo de datos para alcanzar el mínimo.

En lo que se refiere al tamaño máximo de paquete existen varios factores que tienden a limitarlo incluyendo:

- El deseo de aumentar el tamaño de los buffers en la estación para enviar y recibir paquetes.
- Consideraciones similares que conciernen a los paquetes buffers que algunas veces son interconstruidos en el mismo controlador Ethernet.
- La necesidad de evitar la obstrucción del canal e incrementar el promedio de la latencia del canal por otras estaciones. El manejo de los buffers tiende a ser la consideración importante.
- El requerimiento máximo para buffers en la estación es usualmente un parámetro de software de nivel mas alto determinado por la arquitectura de la red total, es típicamente sobre el orden

CAPITULO II

500 a 2000 bytes. El tamaño de cualquier paquete o buffer en un controlador es un parámetro de diseño de hardware y por lo tanto representa una limitación más rígida.

Para asegurar la compatibilidad de los controladores con buffer, la especificación Ethernet obliga una longitud de paquete máxima de 1526 bytes.

La eficiencia de un sistema Ethernet depende en su mayor parte del tamaño de los paquetes que son enviados y puede ser muy alta cuando se utilizan paquetes largos. Si se reduce el tamaño promedio del paquete, se disminuye la detección de colisiones y la escucha de portadora. Cuando el tamaño del paquete se aproxima al intervalo de colisión el performance del sistema se degrada y ya no detecta colisiones.

En una red cuyo intervalo de colisión es mayor que un paquete vacío se presentan dos alternativas:

- **Autorizar paquetes diminutos:** En este caso el transmisor fallara algunas veces para detectar colisiones requiriendo retransmisión en un nivel mas alto. El receptor puede utilizar un filtro de fragmentos parcial para detectar fragmentos de colisión menores que un paquete vacío, pero fragmentos de colisión mas largos deben de ser rechazados por otros verificadores de error tales como CRC.
- **Prohibir paquetes diminutos:** En este caso el transmisor siempre detecta colisión y ejecuta una pronta retransmisión. El receptor utiliza el filtro de fragmentos para descartar automáticamente todos los paquetes más cortos en el intervalo de colisión. La especificación Ethernet prohíbe paquetes diminutos requiriendo un paquete mínimo del campo de datos de 46 bytes.

DISEÑO DE INTERFACE CONTROLADOR-ESTACIÓN.

La confiabilidad y eficiencia de la red pueden verse afectadas por las características que presente la interface Controlador-Estación.

ENCENDIDO Y APAGADO DEL CONTROLADOR.

Las características que definen a un buen controlador son las siguientes:

- Mantener el receptor encendido de manera que pueda recibir paquetes pegados es decir aquellos que se hallan separados por un mínimo espacio entre paquetes.
- Recibir paquetes que una estación se transmite a si misma.

- Para que estas características se satisfagan, se requiere de ciertos requerimientos técnicos que se detallan a continuación:

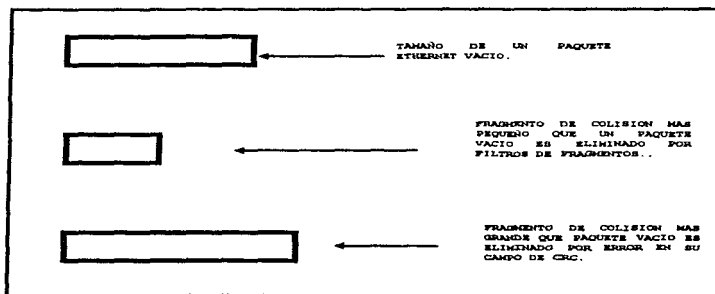


Figura II.3.1.8. Manejo de paquetes diminutos.

MANTENIENDO EL RECEPTOR ENCENDIDO.

La causa más común de la pérdida de un paquete no se debe ni a las colisiones, a CRC's erróneos sino a que simplemente el receptor no estaba "escuchando". Ethernet es un dispositivo asíncrono en el cual se puede presentar un paquete en cualquier momento, por lo cual es importante que el software de mas alto nivel mantenga habilitado al receptor.

El problema puede volverse mas delicado porque incluso cuando se esta operando normalmente, puede haber periodos durante los cuales el receptor no está escuchando. Puede haber ciclos entre ciertas operaciones en los que el receptor se queda apagado, por ejemplo, un turno de receptor a receptor, se lleva a cabo después de que un paquete es recibido y antes de que el receptor sea habilitado de nuevo. Si el diseño de la interface, controlador, o software de la estación mantiene apagado el receptor por mucho tiempo, los paquetes que van llegando pueden perderse en ese periodo del ciclo. Esto pasa con mayor frecuencia en los servidores de una red, los cuales pueden estar recibiendo paquetes de fuentes distintas en una sucesión muy rápida.

Si bajan a la línea paquetes muy juntos un segundo paquete se perderá en el tiempo de ciclo de receptor a receptor. El mismo problema puede ocurrir con una estación de trabajo normal, por ejemplo si un cierto paquete sigue inmediatamente a otro destinado a toda la red, entonces, la estación de trabajo recibiría el paquete de difusión general, pero no alcanzaría a recibir el que va específicamente direccionado hacia ella.

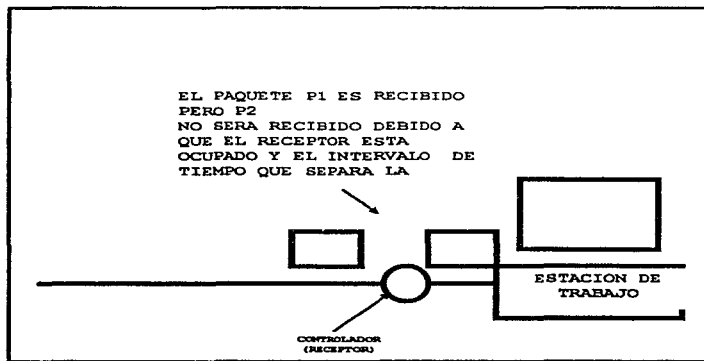


Figura IL3.1.9. Manejo de paquetes por el controlador.

REPETIDORES.

Se utilizan para extender la longitud del sistema de transmisión mas allá de los límites físicos impuestos por el medio de transmisión. Un repetidor utiliza dos transeptores para conectar a dos segmentos Ethernet diferentes y los combina en un canal lógico amplificando y regenerando

las señales que pasan a través de este canal lógico en ambas direcciones. Son transparentes al resto del sistema, por lo tanto propagan una colisión detectada sobre un segmento a través del otro segmento.

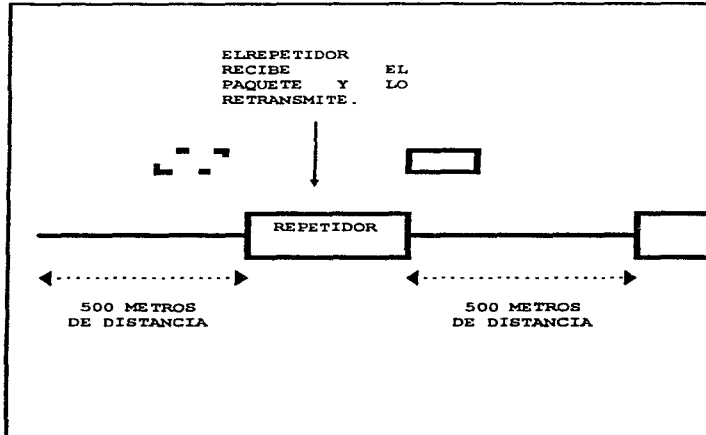


Figura IL3.1.10. Repetidores.

TRANCEPTORES.

Es un dispositivo que contiene la electrónica necesaria para transmitir y recibir señales sobre el canal y reconocer la presencia de señales cuando otra estación está transmitiendo. También reconoce una colisión cuando dos o más estaciones transmiten simultáneamente.

Para evitar la interferencia multiruta en una Ethernet debe haber unicamente una ruta entre dos estaciones a través de la red. (El nivel más alto en una arquitectura inter-red puede apoyar rutas alternas entre estaciones a través de diferentes canales de comunicación).

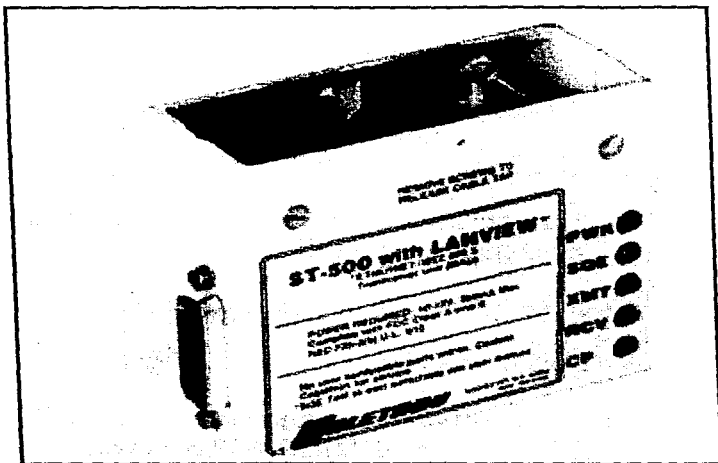


Figura II.3.1.11. Tranceptor Ethernet.

TERMINADO.

Es un componente que se instala en el extremo del cable coaxial y que tiene una impedancia característica del cable.

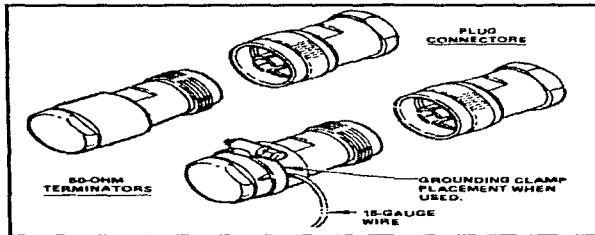


Figura IL3.1.12. Terminadores de red Ethernet.

TAPS

Dispositivo que permite la conexión con el cable coaxial sin romperlo.



Figura IL3.1.13. TAP utilizado en redes Ethernet.

II.3.2. PROTOCOLO TCP/IP

¿ QUE ES TCP / IP ?

TCP / IP es un conjunto de protocolos desarrollados para permitir a las computadoras compartir recursos a través de una red. Fue desarrollado por una comunidad de investigadores que se centraron alrededor del ARPANET. Ciertamente el ARPANET es la red más conocida de TCP / IP. Sin embargo ya en junio del 87, al menos 130 diferentes proveedores tienen productos que soportan TCP / IP, y miles de redes de todas clases lo utilizan.

Primero algunas definiciones básicas. El más exacto nombre en honor del conjunto de los protocolos que estamos describiendo es la " suit de protocolo de Internet". TCP y IP tienen dos de los protocolos en esta suit. (estos serán descritos debajo.) Debido a que TCP y IP son los más conocidos de los protocolos, se ha hecho común utilizar el término TCP/IP O IP / TCP para referir toda la familia.

El Internet es una colección de redes, incluyendo el Arpanet, NSFnet, redes regionales como NYsernet, redes locales en un número de instituciones de la Universidad e investigación, y un número de redes militares. El término " internet" aplica a este conjunto completo de redes. El subconjunto de ello que está manejado por el Departamento de Defensa es Referida como la " ddn" (Red de Información de Defensa). Recientemente, el Departamento de Defensa envió una definición de MLISPEC pero sin embargo la mayoría de la comunidad de TCP / IP continúa utilizando el Internet TCP / IP. Algunos proveen funciones de " bajo nivel " necesarias para muchas aplicaciones. Estos incluyen IP, TCP, y UDP.

Otros son protocolos para hacer tareas específicas, por ejemplo transferencia de archivos entre computadoras, correo electrónico, o para acceder información que está en otra computadora. inicialmente TCP / IP estuvo utilizado principalmente entre las minicomputadoras o computadoras grandes.

Así que los más importantes servicios de TCP / IP son:

- **Transferencia de archivos.** El protocolo de transferencia de archivo (FTP) permite a un usuario en cualquier computadora obtener archivos de otra computadora, o enviar archivos a otra computadora. La seguridad es manejada por los requerimientos del usuario, ya que puede especificar un nombre y contraseña de usuario para la computadora. FTP es una utilidad que se

ejecuta en cualquier momento que se desee para poder transferir un archivo de otro sistema o hacia otro sistema.

- **Login remoto.** El protocolo de terminal de red (TELNET) permite a un usuario conectarse a cualquier otra computadora en la red. Se comienza una sesión remota especificando una computadora para conectar . Todas las instrucciones introducidas desde la PC serán enviadas a la computadora a la que se conecta. Pero el programa TELNET hace efectivamente a la computadora local invisible mientras está siendo ejecutado. El TELNET generalmente provee de algún tipo de emulador de terminal común.

Correspondencia por computadora. Estos permiten enviar mensajes a usuarios en otras computadoras. Originalmente, la gente tendía a utilizar solamente una o dos computadoras específicas. Mantendrían " correspondencia en archivos " en aquellas máquinas. El sistema de correspondencia por computadora es simplemente un modo para añadir un mensaje a otro archivo de correspondencia del usuario. Cuando usted envía correspondencia, el software de correspondencia espera poder abrir una conexión con la computadora destino , para enviarla. Si esta fuera una microcomputadora, puede estar apagada, o puede estar corriendo una aplicación. Por esta razón, la correspondencia es manejada normalmente por un sistema mayor, que permite que funcione a toda hora el servicio de correspondencia . El software de correspondencia de microcomputadora entonces hace una interfaz de usuario que recupera correspondencia del servidor de correspondencia. Debido a que las redes de hoy en día incluyen microcomputadoras, workstations , minicomputadoras, y computadoras grandes esta labor de asignar un servidor de archivos sea facilitado. Un servidor es un sistema que provee un servicio específico para el resto de la red. Un cliente es otro sistema que utiliza este servicio.

EL NIVEL TCP

Dos protocolos separados están involucrados en manipular los paquetes TCP / IP. TCP (el protocolo de control de transmisión) es responsable de dividir el mensaje en paquetes (datagramas), reuniéndolos en el otro extremo, encontrando cualquier cosa que se pierda, y colocando los paquetes en el orden correcto en el otro extremo. IP (el " internet protocolo ") es responsable de encauzar los paquetes individualmente por la red. La interfaz entre TCP E IP es bastante simple. TCP simplemente entrega a IP un paquete con un destino. IP no conoce el orden

CAPITULO II

de los paquetes relaciona a cualquier paquete antes de que el o después el. TCP tiene que conocer el orden de los paquetes que recibe. Esta tarea es referida como "demultiplexaje ". En Realidad, hay varios niveles de demultiplexaje en TCP / IP. La información requerida para hacer este demultiplexaje está contenida en una serie de "encabezados".

Un encabezado es simplemente algunos octetos pegados al comienzo de un paquete por algún protocolo que permite seguir el orden de los mismos y poder obtener la información. - Enseguida se muestra un esbozo de los encabezamientos que se insertan en un mensaje que pasa a través de una red de TCP / IP típica.

Supongamos que el siguiente es un mensaje que deseamos enviar a alguna otra computadora:

.....

TCP lo rompe en fragmentos manejables. (Para hacer esto, TCP debe conocer cual es el tamaño del paquete que la red puede manipular)

.....

TCP pone un encabezado en el frente de cada paquete. Este encabezado realmente contienen al menos 20 octetos, pero los más importantes son una fuente y un destino; " número de puerto " y un " número de secuencia ". El número de puerto es utilizado para seguir el rastro de diferentes conversaciones. Supongamos que 3 diferente personas están transfiriendo archivos. TCP podría asignar los números de puerto 1000, 1001, y 1002 a estas transferencias. Cuando se está enviando un paquete a este se le asigna un número de puerto " fuente ", ya que usted es el emisor del paquete. Por supuesto el TCP en el otro extremo también tiene un número de puerto para la conversación. El TCP tiene que conocer el número de puerto utilizado por la otra terminal también. Cada paquete tiene un número de secuencia. Este es utilizado para que la terminal destino pueda asegurarse que obtiene los paquetes en el orden correcto, y que no haya fallado alguno.

A continuación se muestra un paquete trabajado por TCP:

Source Port				Destination Port				
Sequence Number								
Acknowledgment Number								
Data		U	A	P	R	S	F	
Offset	Reserved	R	C	S	S	Y	I	Window
		G	K	H	T	N	N	
Checksum				Urgent Pointer				
Su información siguiente es 500 octetos								
.....								

Si abreviáramos el encabezado de TCP COMO " T ", todo el archivo ahora parece este:

T.... T.... T.... T.... T.... T.... T....

Los campos que no se han descrito generalmente están para asegurarse de lograr la conexión y que el paquete a llegado a su destino El destino, tiene un campo para devolver un " Acknowledgement " . Por ejemplo, el envío de un paquete con un Acknowledgement de 1500 indica que se ha recibido toda la información hasta el número 1500 de octeto. Si el sender no obtiene un acknowledgement dentro de un tiempo razonable, envía la información de nuevo.

La ventana (window) es utilizada para el control de cuantos datos pueden estar en tránsito en cualquier momento.

El campo " Urgente " permite indicar cuando se debe saltar hacia adelante en el procesamiento de un octeto en particular. Este es frecuentemente útil para eventos manipulados asincrónicamente, por ejemplo cuando usted tecldea el carácter de control u otro mando que interrumpe la salida.

EL NIVEL IP

TCP envía cada uno de los paquetes a IP. Por supuesto tiene que saber el IP la dirección Internet de la computadora en el otro extremo. Observe que esto es lo que concierne al IP. No

checha cual es en el paquete, o el encabezado de TCP. El trabajo de IP es simplemente encontrar un itinerario para los paquetes. Para permitir la entrada a otros sistemas intermediarios y el paquete no se pierda, añade su propio encabezado. Las cosas principales en este encabezado son las direcciones Internet fuentes y destino (32 bit de direcciones, como 128.6. 4 194), el número de protocolo, y otra suma de control. La Dirección Internet fuente es simplemente la dirección de la máquina que envía el paquete. (esto es necesario debido a que la otra terminal debe saber de donde provino el mensaje.) La dirección Internet destino es la de la otra máquina. (esta es necesaria para que cualesquier nodo del medio conozca hacia donde va dirigido el paquete.) El número de protocolo dice a IP cual es el otro extremo para enviar el paquete TCP. Finalmente, la suma de control permite ha IP en la otra terminal verificar que el encabezado no se haya dañado en tránsito o evitar que el mensaje llegue a un lugar equivocado.

El mensaje quedara parecido al siguiente :

Version	IHL	Type of Service	Total Length
Identification	Flags		Fragment Offset
Time to Live	Protocol		Header Checksum
Source Address			
Destination Address			
TCP encabezado, su información.....			

Si representáramos el encabezado de IP por un " I ", su archivo ahora parece este :

IT.... IT.... IT.... IT.... IT.... IT.... IT.... IT....

La bandera y el fragmento offset es utilizado para seguir el rastro de las piezas cuando unos paquetes se tienen que dividir. Esto puede ocurrir cuando paquetes son enviados a través de una red por la que sean demasiado grandes. El tiempo en vivo(time to live) es un número que es decrementado siempre que el paquete pasa a través de un sistema. Cuando es cero, el paquete es desechado.

II.3.3. INTERCONEXIONES.

FORMA DE ORGANIZACIÓN DE LA RED INTERNA DE COMPUTO DE LA UNIDAD XOCHIMILCO.

Actualmente la Unidad Xochimilco cuenta con el siguiente equipo de computo:

- 1 Computadora HP3000 modelo 52.
- 2 Computadora HP9000 modelo 827S
- 5 Work Station HP9000 serie 700
- 1 Work Station digital modelo 3100
- 1 Work Station digital modelo 5200
- 1 Red de CD's ROM
- 500 Microcomputadoras

Este equipo se encuentra distribuido en los diferentes edificios en el campus universitario.

Para comunicar todo este equipo, se cuenta con una columna vertebral de comunicaciones con un kilometro de longitud y 9 kilómetros de ramificaciones, las cuales recorren en forma estratégica los edificios más importantes de la Unidad.

La estrategia óptima de conexión, es integrar directamente los computadores "grandes" directamente a la columna vertebral y las PC's y las Work Station en subredes, que a su vez están conectadas a la columna vertebral.

Generalmente la utilización de el Servidor de archivos la Work Station esta en los Departamentos Académicos, y son usados en los casos en que estos necesiten hacer uso de ellos.

Es importante observar que desde cualquier computador se puede hacer referencia a cualquiera de las PC de esta red, por lo cual, cualquiera de las PC's se comportan como una computadora más de la red.

CAPITULO II

Lo anterior es de vital importancia, ya que la UAM sigue este mismo modelo para su red, por lo que, podemos afirmar que todos los computadores de la UAM (que están conectados en red) están comunicados entre sí.

DESCRIPCIÓN FORMAL DE LA RED DE LA UNIDAD XOCHIMILCO.

La red de la Unidad Xochimilco técnicamente se podría describir de la siguiente manera:

- Forma topología (forma de arreglo de la conexión) de la red: tipo bus.
- El canal físico de transmisión es un cable coaxial tipo Ethernet norma IEEE 803.3.
- Protocolo de comunicación TPC/IP.
- De sistema distribuido abierto, es decir, distintos sistemas operativos funcionando al mismo tiempo en la red y con comunicación.
- Red de área local.
- Velocidad de transmisión de 10 MB/S.
- Eliminación al máximo de posibles colisiones en la comunicación, utilizando para ello puentes.
- Longitud aproximada de 10,000 metros

Enseguida, se visualizará una tabla con los diferentes equipos de cómputo de dominio de la comunidad Universitaria.

Equipo ubicado en la Unidad Azcapotzalco.

EQUIPO	NOMBRE	Sistema operativo
hp9000 modelo 827s uso académico.	hp9000a1	UNIX
hp9000 modelo 827s uso administrativo.	hp9000a2	UNIX
hp9000 modelo 827 uso S. biblioteca.	ha19000	UNIX
hp3000 modelo 52 uso administrativo.	hp3000a	MPE

Equipo ubicado en la Unidad Iztapalapa.

EQUIPO	NOMBRE	SISTEMA OPERATIVO
hp9000 modelo 827s uso administrativo.	hp9000	UNIX
TITAN uso académico.	titan	UNIX
CYBER 830-11 uso académico.	uamizt	NOS/VE
hp3000 modelo 52 uso administrativo	hp3000i	MPE

Equipo ubicado en la Unidad Xochimilco.

EQUIPO	NOMBRE	SISTEMA OPERATIVO
hp9000 modelo 827S uso académico.	hp9000x0	UNIX
hp9000 modelo 827s uso administrativo	hp9000x1	UNIX
hp9000 modelo 720 uso académico.	hp720x0	UNIX
hp9000 modelo 730 uso académico.	hp730x0	UNIX
hp3000 modelo 52 uso administrativo	hp3000x	MPE

Equipo ubicado en Rectoría General

EQUIPO	NOMBRE	SISTEMA OPERATIVO
hp9000 modelo 827S uso administrativo	hp9000r	UNIX
CYBER 830 uso administrativo	uamrec	NOS

CONEXIÓN ENTRE LOS DIFERENTES EQUIPOS:

Para poderse conectar a una red de cómputo es necesario que la computadora tenga una tarjeta especial, la cual le permitirá poder establecer dicha comunicación.

A continuación se muestra la forma de realizar la conexión entre los equipos con los que cuenta la Universidad Autónoma Metropolitana en sus cuatro Unidades (tres académicas y una administrativa):

PC --> HP-9000
PC --> HP-3000
PC --> CYBER
HP-3000 --> CYBER
PC --> HP-730
PC --> HP-720
CYBER --> HP-3000

Para darse de alta a equipos de la red UAM ubicados en cualquiera de las tres unidades o rectoría general, desde una estación de trabajo de una red de microcomputadoras, se debe dar de alta el sistema con la siguiente instrucción:

- HP9000X0** (Para la HP-9000 de Xochimilco Academia)
- HP9000X1** (Para la HP-9000 de Xochimilco Administración.)
- HP720X0** (Para la WORK STATION 720 de Xochimilco)
- HP730X0** (Para la WORK STATION 730 de Xochimilco)

HP9000A1	(Para la HP-9000 de Azcapozalco Académia)
HP9000A2	(Para la HP-9000 de Azcapozalco Administración)
HALL9000	(Para la HAL de Azcapozalco)
HP9000	(Para la HP-9000 de Iztapalapa)
TITAN	(Para la TITAN de Iztapalapa)
SUN	(Para la SUN de Iztapalapa)
HP9000R	(Para la HP-9000 de Rectoria)
UAMREC	(Para la CYBER de Rectoria)

Ejemplo de conexión de la red PC a una HP-9000:

Darse de alta en la HP-9000 de la unidad requerida, por medio de la siguiente instrucción:

hp9000x0	(Unidad Xochimilco)
hp9000x1	(Unidad Xochimilco)
hp9000	(Unidad Iztapalapa)
hp9000a1	(Unidad Azcapozalco)
hp9000a2	(Unidad Azcapotzalco)
hp9000r	(Rectoria)

Después de haber tecleado el comando que activa la conexión con el equipo solicitado se verá aparecer en la pantalla el mensaje siguiente:

login: En esta parte el equipo le solicita al usuario su clave de acceso al sistema, la cual es proporcionada por la Coordinación de Servicios de Cómputo de cada una de las unidades de la Universidad. Se muestra un ejemplo de una cuenta de la unidad Xochimilco y se verá de la siguiente manera:

login:saat0001

Al presionar la tecla de <ENTER> el equipo envía otro mensaje, en el cual le solicita al usuario su palabra clave de acceso al sistema, la cual es elegida por el usuario al solicitar su cuenta, o en su defecto el mismo usuario la asigna al trabajar dentro del sistema con su clave de acceso, el mensaje será visualizado como sigue:

password: En esta parte es donde se debe de teclear dicha palabra, conocida en términos de computación como password (palabra de acceso), se deberá de observar que aunque se este tecleando la palabra el cursor no avanza y no aparece nada en la pantalla, pero eso no debe

de tener mucha importancia ya que es la forma en la cual el equipo protege a la cuenta del usuario y así impedir que otra persona no autorizada pueda ingresar a ella. Al presionar la tecla de <ENTER> se observará que en la pantalla aparece un mensaje extenso y le pregunta al usuario el tipo de terminal que está utilizando en ese momento, no se debe de olvidar que ingresamos desde una microcomputadora (PC) como terminal remota del equipo HP-9000 solicitado.

II.3.4. ADMINISTRACIÓN.

RED:

Básicamente, el objetivo principal de una red es el de compartir recursos entre usuarios, de tal forma que al ser almacenada información en un área común, dicha información podrá ser compartida por diferentes usuarios. Esta área común es el file server, el cual administra la operación de las microcomputadoras (estaciones de trabajo) conectadas a la red en las cuales se darán de alta los usuarios.

Existen dos tipos de usuarios en una red:

1. El usuario común, el cual puede o no tener derecho de usar todos los menús de trabajo, dependiendo de los derechos que le hayan sido asignados al crear su cuenta en la red.
2. El usuario privilegiado ó supervisor, el cual tiene derecho de poder utilizar todos los menús de trabajo y es el responsable de la organización de la red.

SUPERVISOR:

Es aquel usuario responsable de organizar y coordinar el funcionamiento de la red, dando de alta o baja a usuarios comunes y otorgarles derechos (atributos) dentro de la misma. Además, este posee todos los atributos.

GRUPOS:

Por comodidad, se crean grupos de usuarios que tendrán los mismos atributos. Cada grupo tendrá sus propios atributos y los usuarios tendrán los derechos de cada grupo en que hayan sido dados de alta, además de los propios.

USUARIO:

Es aquella persona autorizada para trabajar en la red. Para esto, debe poseer una cuenta que le servirá como clave de entrada a la red y en la cual están definidos sus derechos dentro de la misma.

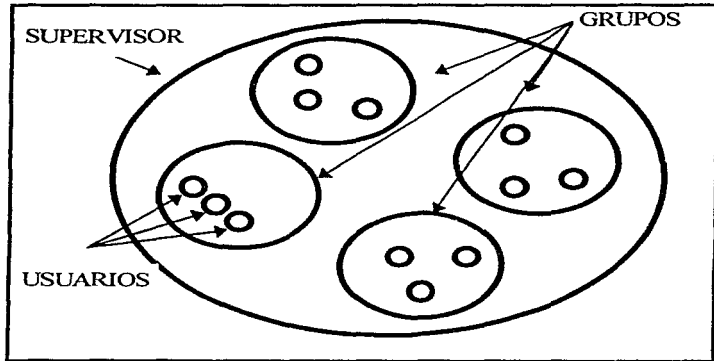


Figura II.3.4.1. Dominio de los usuarios de una red.

DIRECTORIO: Los directorios son áreas de almacenamiento que nos permiten organizar adecuadamente la información. Cada directorio tiene derechos.

ATRIBUTO: Es la característica de un archivo para permitir a un usuario accederlo o hacer uso de él. Los atributos de un archivo son los siguientes.

S - SHAREABLE	(COMPARTIDO)
N - NON SHAREABLE	(NO COMPARTIDO)
RO- READ ONLY	(SOLO LECTURA)
RW- READ WRITE	(LECTURA Y ESCRITURA)

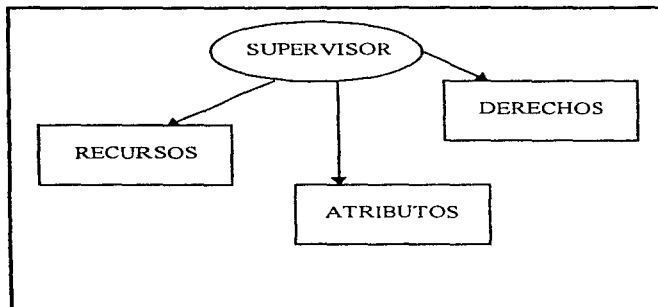


Figura II.3.4.2. El supervisor asigna atributo, recursos y derechos en la red.

SEGURIDAD: Por razones de seguridad, es estrictamente necesario controlar el acceso de los usuarios a la información, para evitar la pérdida de archivos o mal uso de los mismos.

DERECHOS: Son otra herramienta para organizar la red cuidando de la seguridad en la operación de la información. Los derechos pueden ser otorgados a usuarios, grupos y directorios en el momento de su creación.

- **DERECHOS A USUARIOS.** Al crear al usuario, el supervisor le da derechos para crear o no archivos, acceder o no subdirectorios.
- **DERECHOS A GRUPOS.** Un grupo contiene usuarios, y además puede tener definidos derechos y la posibilidad de entrar o no a un subdirectorio.
- **DERECHOS A DIRECTORIOS.** El acceso a directorios puede ser restringido por medio de derechos. Esto permite definir directorios que serán áreas de trabajo (poseen todos los atributos), directorios de solo lectura (no es permitido borrar archivos en esos directorios), etc.

En seguida se listan los derechos y su abreviación:

R.- Read. Permite la lectura de archivos

- W.- Write. Permite escribir sobre archivos
- O.- Open. Para abrir archivos ya existentes
- C.- Create. Permite crear y abrir archivos
- D.- Delete. Borrar archivos
- P.- Propiedad de derecho para:

- crear, renombrar y borrar subdirectorios
- asignar derechos a usuarios y subdirectorios
- asignar derechos a usuarios y en el propio directorio

LOGIN: Para poder darse de alta en la red, esta debe tener definidos a los usuarios. Una vez definidos, se siguen dos pasos para acceder a la red; el primero es "atarse" y el segundo "firmarse".

ATARSE: Conecta la estación de trabajo al server

FIRMARSE: Significa ser reconocido por la red. Para esto, se utiliza el comando LOGIN y si es requerido, se dará el "PASSWORD"

DEFINICIÓN Y ADMINISTRACION DE LA ESTRUCTURA DE LA RED.

El responsable de crear una estructura adecuada de la red, así como también de su administración, es el supervisor, el cual, define el perfil de los usuarios para crear grupos que utilicen las mismas aplicaciones y compartan los mismos archivos y programas. Es muy recomendable que para cada aplicación, exista un subdirectorio al que tendrán acceso determinados usuarios, los cuales deberán tener restringidos los derechos de escritura, borrado y modificación de archivos por cuestiones de seguridad. También se recomienda crear y asignar a cada usuario un subdirectorio para ser utilizado como área de trabajo, por lo que deberá tener todos los privilegios sobre él para poder grabar, modificar y borrar información en él. El supervisor debe asignar un balance a cada usuario para determinar la cantidad de servicio de que puede hacer uso. Esa cantidad se ve disminuida cuando el usuario solicita servicios de la red, pero puede ser incrementada. Se debe restringir el horario de trabajo de los usuarios indicándose horarios y días en los cuales les es permitido acceder a la red. Se debe efectuar la detección de usuarios intrusos que intenten violar la seguridad del login de algún usuario. La detección se realiza cuando el usuario intenta darse de alta en una cuenta y falla en el login 3 veces,

considerando el sistema al usuario como un intruso e impidiéndole firmarse. Además el sistema desactiva la cuenta durante cierto intervalo de tiempo, el cual también puede ser modificado por el supervisor. Es el supervisor quien define a los usuarios o grupos de usuarios que pueden ser operadores de consola. Pueden ejecutar comandos de consola desde su estación de trabajo, consultar la lista de operadores de consola, etc. El supervisor debe definir dispositivos y formas de impresión. Los usuarios pueden ver las impresoras y formas que el supervisor configuró. El supervisor debe definir las impresoras y plotters. Al mandar un archivo a imprimir, se le asigna un N° de job y se envía a cola de impresión. Inicialmente, el sistema tiene una sola cola de impresión, pero para el supervisor es posible crear más de una. Cada cola de impresión debe tener su correspondiente spool de impresión. El supervisor también define los operadores de cola, es decir, a los usuarios que podrán controlar colas de impresión. Además, se asignan impresoras a colas de impresión y obtiene información sobre cualquier cola de impresión. El supervisor de la red, debe dar de alta a los usuarios y asignarles derechos sobre directorios para que puedan o no hacer uso de las aplicaciones contenidas en ellos. Cuando un usuario es creado, se deben definir las características y restricciones que tendrá, tales como tiempo de caducidad de la cuenta, password, espacio en disco a utilizar, etc. Al crear un usuario, posee todos los derechos que le fueron asignados dentro de su área de trabajo además de los del grupo de trabajo al que pertenece por default al momento de ser creado. Cuando un usuario es asignado a un grupo, conserva todos sus derechos y además los derechos del grupo al que fue asignado. Esto facilita la asignación de derechos y privilegios a usuarios que poseen el mismo perfil. El supervisor debe asignar privilegios a cualquier usuario sobre algún directorio. También asignar atributos y niveles de seguridad a directorios (Directory Rights) o archivos ((File Atributes). Generalmente se manejan ocho tipos de restricciones para los directorios:

- R- Leer desde archivos abiertos
- W- Escribir en archivos abiertos
- O- Abrir archivos
- C- Crear archivos
- D- borrar archivos
- P- Parental, el cual permite:
 - Crear, renombrar y borrar subdirectorios

redefinir trustees y directory rights en directorios y subdirectorios

S- Comparte el uso de archivos

M- modificar los atributos de los archivos

II.3.5. PATHWAY.

INTRODUCCIÓN.

PATHWAY ACCESS combina los protocolos TCP/IP y una interfaz gráfica de usuario en un paquete sencillo y amigable. El protocolo a nivel de usuario y servicios que provee PATHWAY ACCESS son :

- TELNET (Terminal Emulation Link over Network), nos permite emular terminales conectadas a HOST remotos con sistemas UNIX y DEC-VAX, y MAINFRAMES IBM.
- FTP (File Transfer Protocol), para transferir archivos entre un HOST remoto y nuestra PC o Macintosh.

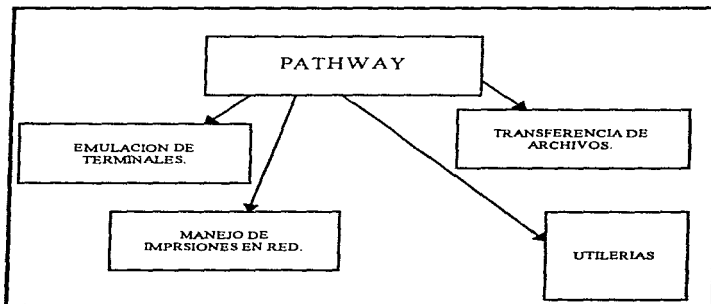


Figura II.3.5.1. Servicios que proporciona Pathway.

- Servicios de Impresión en Red (usando LPR y protocolos de impresión de imágenes , redireccionador de impresiones, y un servidor LPR), nos permite imprimir en una impresora remota y permite a usuarios remotos imprimir en nuestra impresora local.
- Una aplicación remapeadora de teclado, nos permite remapear el código de teclas generado por el teclado durante una sesión de terminal.
- Un SCRIPT COMPILER(compilador de escritura), permite habilitar los escritos y usar SCRIPT para realizar tareas automáticamente.

PATHWAY ACCESS contiene muchas características que facilitan el uso de la red desde la PC o una Macintosh mientras mantiene una interface familiar en el uso de sus herramientas y técnicas.

USANDO EL EMULADOR DE TERMINALES.

TELNET es un protocolo de acceso remoto que habilita a la PC o Macintosh para comunicarse con un HOST de la red como si fuese una terminal conectada directamente. PATHWAY ACCESS comúnmente soporta la emulación de las terminales DEC VT100, VT200, y la familia de terminales VT300; la familia de display stations IBM3270 (incluyendo la IBM 3179-G COLOR GRAPHICS DISPLAY STATION); y las terminales TEKTRONIX 4105 y 4010.

TRANSFIRIENDO ARCHIVOS.

El protocolo de transferencia de archivos (FTP) permite a la PC o Macintosh la transferencia de archivos con un HOST remoto. El FTPD (File Transfer Protocol Daemon) o FTP servidor permite utilizar la PC o Macintosh como servidor de archivos para que otro cliente FTP en la red pueda recibir archivos de la PC o Macintosh. También FTP nos permite :

- Cambiar y listar directorios.
- Renombrar y borrar archivos.
- Crear y remover directorios del HOST remoto.

IMPRIMIENDO ARCHIVOS.

Los servicios de impresión de PATHWAY ACCESS permite a la PC o Macintosh actuar como un LPR cliente o un LPR servidor. También permite redireccionar impresiones e imprimir

directamente desde Windows a impresoras remotas. Los servicios de impresión le permiten hacer las siguientes cosas:

- Usando una sesión de impresión LPR, la PC o Macintosh puede actuar como un LPR cliente e imprimir en una impresora conectada a un HOST ejecutando un LPR servidor.
- Usando una sesión de impresión con protocolo de imagen (IPR) se puede imprimir en una impresora remota de imágenes.
- En Windows, se puede usar el redireccionador de impresoras LPRINT o RFCLPRNT (basado en RFC 1179) para imprimir en un LPR servidor o impresora de imágenes. Usando el redireccionador de impresiones, se puede redireccionar un puerto paralelo local a una impresora remota, y mandar diferentes impresiones de Windows a través de este puerto.
- Podemos definir una impresora conectada a la PC o Macintosh como LPR servidor para que clientes remotos puedan hacerlo en la impresora local a la PC o Macintosh.

SESIONES DE REDES.

PATHWAY ACCESS permite tener varias sesiones de redes y tener ventanas asociadas a estas para abrirlas comúnmente. PATHWAY ACCESS permite mantener tres sesiones de red en las PCs y en la Macintosh permite hasta 12 sesiones.

EMULACIÓN DE TERMINALES Y TECLADOS.

PATHWAY ACCESS provee de archivos de mapas de teclados para permitir al teclado y pantalla trabajar como si fueran alguna de las siguientes terminales:

- VT100, VT220, VT240, VT320, y VT340 para modo texto o para sesiones modo gráfico en DEC VAX/VMS o sistemas UNIX.
- IBM 3270 e IBM 3179-G para sesiones en modo texto o gráfico.
 - IBM 3278 modelos 2,3,4 y 5.
 - IBM 3176-G modelo 3 para IBM VM o VMS sesiones en modo texto gráfico.
 - TEKTRONIX 4105 / 4010 para TEKTRONIX modo gráfico o VT100 para sesiones modo texto(solo modelo 3)

Para cada una de estas terminales provee las teclas de funciones especiales que no son mapeadas directamente por los teclados de las PCs o Macintosh, PATHWAY ACCESS mapea estas teclas especiales de funciones con varias teclas o combinación de teclas en el teclado de la PC o Macintosh. Este mapeado de teclas y combinación de teclas los provee PATHWAY ACCESS en sus archivos de mapeo. Las funciones de las teclas o combinación de ellas dependen de la aplicación remota que se este usando.

USANDO SCRIPTS PARA AUTOMATIZAR RUTINAS.

PATHWAY ACCESS proporciona un compilador de SCRIPTs que nos permite automatizar rutinas para realizar tareas. Por ejemplo, se puede escribir un SCRIPT para conectarse automáticamente a un HOST remoto, el cual pide USERNAME y PASSWORD, y checa si se tiene correo. O hacer un SCRIPT para ejecutar una aplicación en un HOST remoto.

II.4. MANEJADOR DE BASES DE DATOS ORACLE

II.4.1. DEFINICIÓN.

Un manejador de B.D. es la clave para resolver los problemas de información. En general un sistema manejador de B.D. (DBMS) debe ser un confiable manejador en un ambiente multiusuario y así muchos usuarios pueden acceder concurrentemente al mismo dato. Todo esto tiene que corresponder al nivel o desarrollo del usuario de la B.D. Un DBMS se encarga también de autorizar ó no el acceso y dar soluciones eficientes a las fallas que se pueden suscitar. A continuación se muestra un esquema de las tareas de un DBMS. (Figura II.4.1.1)

II.4.1.1. CARACTERÍSTICAS DE ORACLE

El servidor ORACLE da soluciones efectivas y eficientes a la mayoría de las fallas de la B.D.:

B.D. Grandes Y Control De Espacios	ORACLE, soporta potencialmente B.D. mayores a cientos de gigabyte en tamaño. Para hacer un uso eficiente de los costosos dispositivos de hardware, permite el control total sobre espacio.
---	--

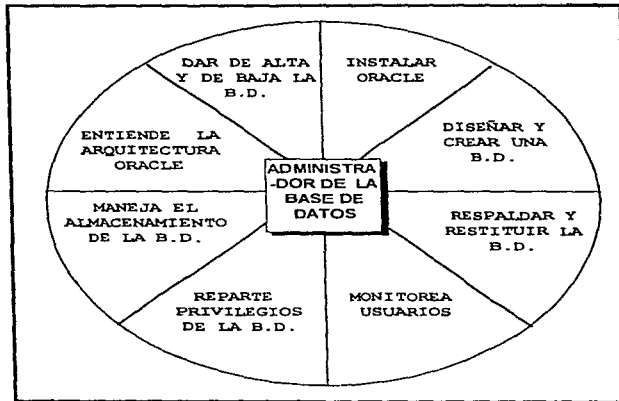


Figura II.4.1.1 Tareas de un Sistema Administrador de Base de Datos.

Usuarios concurrentes En la B.D. ORACLE, soporta un número muy grande de usuarios concurrentes, ejecutando una variedad de aplicaciones de las B.D. operando sobre el mismo dato. Esta minimiza argumentos y garantiza la concurrencia de datos.

Alto Desarrollo En El Proceso De Transacciones ORACLE, mantiene características anteriores con un alto grado de desarrollo de sistemas completos.

Alta Disponibilidad	ORACLE, trabaja 24 horas al día sin pérdidas de tiempo que limiten el rendimiento de la B.D. La operación normal del sistema como B.D. respalda , las fallas parciales del sistema no interrumpen la B.D.
Disponibilidad Controlada	ORACLE, puede controlar selectivamente la disponibilidad del dato, según el nivel y subnivel de la B.D. Por ejemplo, Un administrador puede rechazar el uso de una aplicación específica, así el dato de la aplicación puede ser recargado, sin afectar otras aplicaciones.
Estándares Aceptados Industrialmente	ORACLE, se une a los estándares aceptados industrialmente para el lenguaje de acceso de datos, sistemas operativos, interfaces de usuarios, y protocolos en la comunicación de red. Es un sistema "Abierto". El servidor de ORACLE7, ha sido certificado por el Instituto Nacional de Tecnología y Estándares de los E. U. como compatible en un 100% con el 2do. Nivel de estándares de ANSI/ISO. ORACLE, satisface completamente los requerimientos de los E. U. e incluye "banderas" para enfatizar el uso del SQL no estándar.
Seguridad Manejable	Para proteger del uso y acceso no autorizado de las B.D., ORACLE esta provisto de seguridad contra fallas, para limitar y monitorear el acceso a datos. Esta característica hace fácil el manejo hasta de el diseño más complejo para el acceso de datos.
Integridad Reforzada de las B.D.	ORACLE refuerza la integridad del dato, "reglas del negocio" que dicta el estándar para aceptar el dato . Como resultado los costos de codificar y manejar muchas aplicaciones de B.D., son eliminados.
Ambiente Cliente/Servidor (Proceso Distribuido)	ORACLE, permite procesar para diferenciar entre el servidor de la B.D. y el programa de aplicación del cliente. Toda la responsabilidad de compartir el manejo de datos, puede ser procesada por, el manejador de sistemas de B.D., mientras las estaciones de trabajo corren su aplicación, concretándose en la interpretación y los datos desplegados.

Sistema de B.D. Distribuido.	Para ambientes computacionales que están conectados via red, ORACLE, combina la localidad física de el dato en diferentes computadoras dentro de una B.D. lógica, que puede ser accesada por todos los usuarios de red. El sistema distribuido tiene el mismo grado de transparencia y consistencia en el dato, que un sistema no distribuido aún recibiendo las ventajas de un manejador de B.D. local.
Portabilidad	El software de ORACLE, esta creado para trabajar bajo diferentes sistemas operativos, ya que es lo mismo en todos los sistemas. El desarrollo de aplicaciones para ORACLE puede ser transportado a otro sistema operativo con muy pocas modificaciones o incluso con ninguna. Implementado en Lenguaje C, ORACLE, funciona sobre "mainframes" minis y micros. Claramente para el usuario y sus aplicaciones, ya que es idéntico en todos los ordenadores para los que está disponible; por lo tanto, ORACLE proporciona un entorno operativo de B.D., completo y consistente independientemente de la máquina y sistema operativo sobre los que se esté ejecutando.
Compatibilidad	El software de ORACLE, es compatible con los estándares industriales, incluyendo los de los diversos sistemas operativos, como se menciona en el punto anterior. El desarrollo de aplicaciones para ORACLE, puede ser usado en cualquier sistema virtual con pocas modificaciones ó incluso con ninguna, ya que sus posibilidades y su interfaz de usuario SQL, son idénticas a las de SQL/DS y DB2.
Conectividad	El software de ORACLE, permite trabajar con diferentes tipos de computadoras y comparte información bajo red con diversos sistemas operativos. Teniendo el mismo software ORACLE se ejecuta sobre "mainframes" minis y micros, simplificando mucho la tarea de conexión de ordenadores en red, tanto de área local como remota. SQL es la interfaz estándar de usuario, para el acceso a todos los datos de la red. Las posibilidades que ofrece ORACLE para el trabajo en el entorno de

redes de ordenadores, permiten el desarrollo de aplicaciones distribuidas.

Capacidad

ORACLE, proporciona junto con una completa implementación del lenguaje SQL, herramientas de desarrollo y de soporte para la toma de decisiones:

- Formateador de pantallas
- Generador de informes
- Generador de menús
- Gráficos en color
- Preparación de documentos
- Diccionario de Datos Integrado
- Hojas de Cálculo.

En fin un sin numero de aplicaciones que se muestran en la figura **II.4.1.2.** de las que se hablará más adelante.

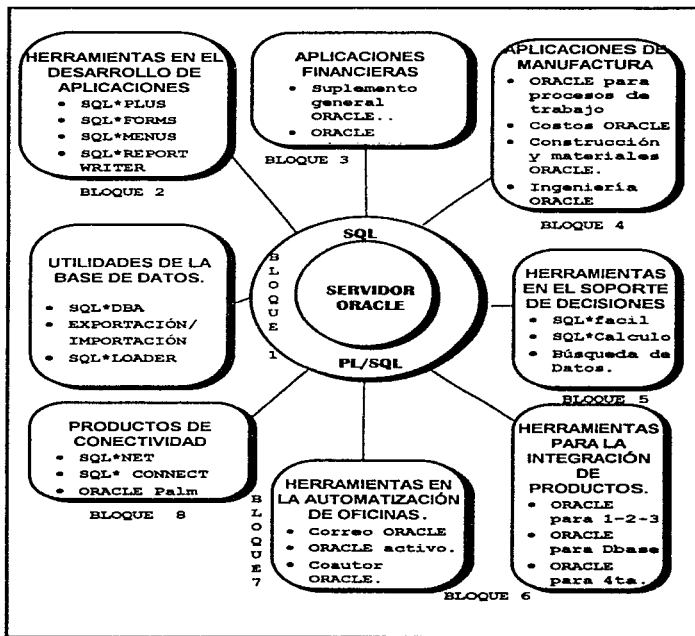


Figura II.4.1.2 ORACLE y Sus Herramientas.

II.4.1.2. COMPONENTES DE ORACLE.

Como se observa en la figura II.4.1.2., ORACLE, esta formado por una serie de herramientas muy grandes, que competen a áreas como la:

- CONTABILIDAD, ARQUITECTURA
- FINANZAS, DISEÑO
- PROCESOS, INGENIERÍA

En el presente trabajo solo se abarcaran y explicaran las herramientas con que se desarrolló el sistema que se generará como resultado a la necesidades presentadas por el Centro de Cómputo de la UAM unidad Xochimilco, para evitar salirnos de la delimitación de este tema, por consiguiente los bloques a desarrollar, tomando en cuenta la figura anterior serán:

- Bloque 1: Servidor ORACLE , SQL y PL/SQL
- Bloque 2: Herramientas en el Desarrollo de Aplicaciones.
- Bloque 8: Productos de conectibilidad.
- Bloque 9: Utilidades de las Base de datos.

II.4.1.3.1. BLOQUE 1, SERVIDOR ORACLE, SQL Y PL/SQL

¿QUE ES PL/SQL?

Las iniciales PL viene de LENGUAJE PROCEDURAL, por lo tanto PL/SQL lo consideraremos un LENGUAJE PROCEDURAL/SQL. Este es un lenguaje de Programación considerado como una extensión del Lenguaje SQL con nuevas formas de trabajo, más fáciles y entendibles para el usuario.

Es una combinación del Poder de manipulación de Datos mediante lenguaje SQL y el Poder de procesamiento de los lenguajes procedurales.

PL/SQL se localiza en el centro del RBDMS (SISTEMA MANEJADOR DE BASE DE DATOS RELACIONALES), o bien dentro de las Herramientas de ORACLE. Se tiene que hacer la observación de que PL/SQL no es un PRODUCTO como pueda serlo SQL* PLUS, SQL* FORMS, sino un nuevo LENGUAJE de programación dentro de esas herramientas.

Los Beneficios que este lenguaje ofrece se listan a continuación:

- Capacidades Procedurales.
- Mejora del rendimiento.
- Portabilidad.
- Integración con el RDBMS.

CAPACIDADES PROCEDURALES

Es un lenguaje de programación estructurado en bloque. Múltiples sentencias están agrupadas formando un bloque.

Contiene las siguientes estructuras:

- Control condicional: (IF...)
- Control iterativo: BUCLES (FOR, WHILE, LOOP).
- Anidamiento de bloques.
- Control robusto de errores: EXCEPCIONES.
- Variables y constantes.
- Conjunto de funciones.

PL/SQL soporta conceptos que no se encuentran en lenguajes tradicionales de programación.

MEJORA DE RENDIMIENTO

- Todas las sentencias de un bloque se envían directamente al núcleo ORACLE, produciendo una reducción considerable del número de llamadas al sistema.
- Si el monitor PL/SQL se encuentra en una herramienta ORACLE como SQL*FORMS, el bloque PL/SQL es procesado en el lado de la aplicación. Solo las sentencias SQL provocan llamadas al núcleo.

PORTABILIDAD

PL/SQL es totalmente portable. Esto indica que cualquier programa PL/SQL que se haya diseñado en cualquiera de las plataformas donde se ejecuta ORACLE, puede ser transportado de una plataforma a otra sin ningún problema de ejecución.

Esta ventaja incrementa la portabilidad del software.

INTEGRACIÓN CON EL RDBMS

- PL/SQL incorpora las ventajas DML (INSERT, UPDATE, DELETE) y las sentencias SELECT de SQL, el lenguaje de comunicación entre la B.D. y las herramientas ORACLE.

- PL/SQL permite obtener variables y constantes basadas en el tipo de dato actual de columnas de la B.D., o referenciarse a estructuras completas de filas de una tabla.

Como conclusión a estas características, PL/SQL redundará en beneficio de:

- A) Independencia de datos.
- B) Mantenimiento de programas.

Para entender mejor el funcionamiento del PL/SQL, obsérvese la figura II.4.1.2.

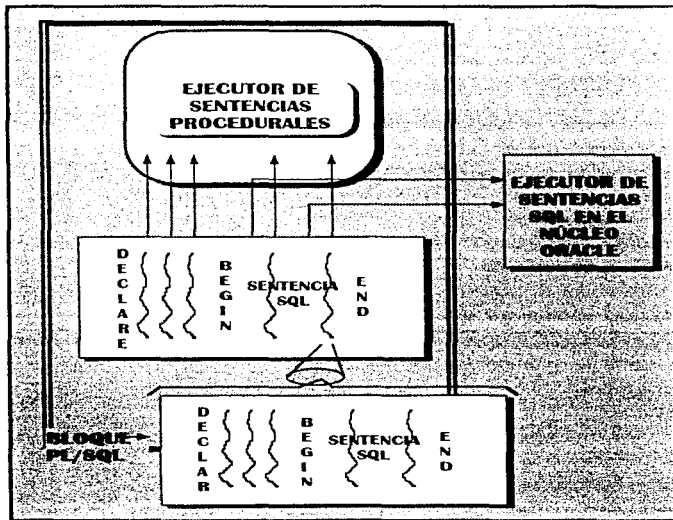


Figura II.4.1.3. Funcionamiento del PL/SQL

VENTAJAS DEL SQL

No procedural

Independencia de Datos

Fácil de aprender

Potente.

DESVENTAJAS

No se puede ejecutar una sentencia basándose en los resultados obtenidos.

No se pueden manipular los datos seleccionados.

Algunos procedimientos requieren lenguajes de alto nivel (user-exit, Pro*SQL).

Algunos problemas se resolverían mejor con un lenguaje procedural.

NOTA: **PL/SQL**, supera todas estas desventajas, tomándolas como ventajas extras en el manejo de este lenguaje.

BLOQUE PL/SQL

Un bloque PL/SQL está formado por una secuencia de sentencias PL/SQL.

Existen dos tipos de sentencias:

SQL (SELECT, INSERT, UPDATE, DELETE, COMMIT).

NO-SQL (IF, FOR, WHILE, OPEN,...)

El bloque es escrito en SQL*PLUS o en otra herramienta ORACLE como por ejemplo SQL*FORMS.

La estructura general de un Bloque PL/SQL es:

DECLARE

<variables y definiciones de constantes>

BEGIN

<sentencias ejecutables>

EXCEPTION

<control de excepciones>

END;

II.4.1.3.1.1 UTILIZACIÓN DEL PL/SQL EN SQL*FORMS

Los tipos de bloques PL/SQL que existen en SQL*FORMS son:

Bloques simples

Bloques de procedimiento (funciones globales)

Bloques Simples

No tienen nombre

Están asociados a triggers definidos

No hace falta definir Begin y End.

Se definirá BEGIN y END si el bloque PL hace referencia a variables locales, cursores o nombres definidos de excepciones.

Bloques de Procedimientos (Globales)

Requieren un nombre propio.

Requieren sintaxis completa de PL/SQL.

Incluyen BEGIN y END.

Estas funciones pueden ser llamadas desde triggers en la forma actual.

II.4.1.3.2. BLOQUE 2, HERRAMIENTAS EN EL DESARROLLO DE APLICACIONES

Dentro de este bloque se puede observar una serie amplia de productos ORACLE, de los cuales abarcaremos solo aquellos que se emplearon para el desarrollo de esta tesis:

SQL* FORMS.

SQL*PLUS.

ORACLE GRAPHICS.

SQL*REPORT WRITER.

SQL*FORMS

El generador de aplicaciones de ORACLE ó SQL*FORMS es un conjunto de herramientas:

IAG : Interactive Application Generator ó Generador Interactivo de Aplicaciones, quien se encarga de generar una aplicación.

IAP: Interactive Application Processor ó Procesador Interactivo de Aplicaciones, quien permite ejecutar una aplicación.

IAD: Interactive Application Designer ó Diseñador Interactivo de Aplicaciones, manipula las descripciones de aplicaciones almacenadas en la B.D.

IAC: Interactive Application Converter ó Convertidor Interactivo de Aplicaciones.

Obsérvese la figura IL4.1.4. para mejor entendimiento.

SQL*FORMS ofrece las siguientes VENTAJAS:

- Una aplicación permite la consulta y/o la modificación interactiva de la base a través de formatos de pantalla.

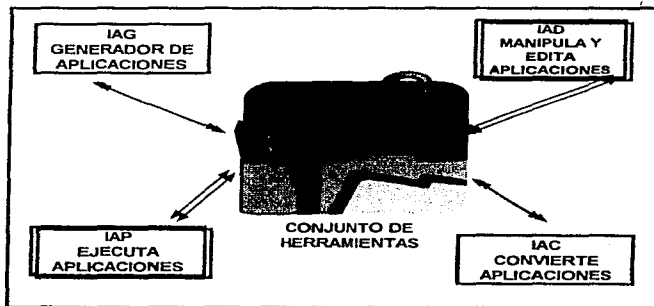


Figura IL4.1.4. Componentes de SQL*FORMS.

- Es posible integrar en ella todos los controles necesarios para mantener la integridad de la base.
- Por fin, una utilidad, FASTFORM, permite crear en unos segundos una aplicación estándar.
- Basta con indicar las tablas que deben figurar en la aplicación.
- FASTFORM generará los formatos de pantalla y los controles basados en las descripciones de tablas del Diccionario de Datos.

- Con solo oprimir doble clic en **FORMS** en el submenú de **SQL*FORMS**, se permite acceder, de una forma transparente para el usuario a todas las herramientas por medio de un formato de pantalla.
- **SQL*FORMS** permite crear, modificar, ejecutar o salvaguardar una aplicación.
- La estructura general de estos menús es la siguiente (Figura II.4.1.5):

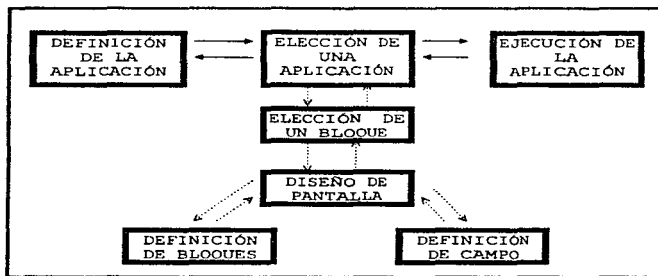


Figura II.4.1.5. Estructura general de los Menús en SQL*FORMS

- **SQL*FORMS** permite crear automáticamente un "Bloque por Defecto"; basta con indicar:
 - a) El nombre de la tabla (y eventualmente de las columnas) que deben figurar en el bloque.
 - b) Cuantos registros de la tabla deben figurar sobre la pantalla.
- **SQL*FORMS** consultará el Diccionario de Datos para obtener la descripción de los campos de la tabla que deben figurar en el bloque.

SQL*PLUS

Al estar manejando ORACLE, usted se podrá dar cuenta que al teclear un comando SQL se obtendrá el resultado deseado inmediatamente. Estamos dentro de SQL Interactivo de ORACLE, llamado SQL*PLUS.

Éste es una herramienta muy útil; y no se debe confundir el lenguaje SQL que puede ser utilizado desde todas las herramientas ORACLE, con los comandos de SQL*PLUS.

FUNCIONES DE SQL*PLUS.

- Trabajo Interactivo con ORACLE mediante SQL.
- El acceso a SQL*PLUS es permitido tras la correcta introducción del nombre de usuario y la palabra clave correspondiente.
- Los comandos SQL pueden ser ejecutados inmediatamente que son almacenados en un buffer y pasados a ORACLE.
- Dispone de ayuda interactiva.

EL BUFFER DE SQL

- La sentencia SQL introducida por el usuario es almacenada en el buffer de comandos SQL.
- La sentencia SQL puede ser:
 - Ejecutada inmediatamente (añadiendo un punto y coma (;) al comando)
 - Editada y listada antes de su ejecución.
 - RUN.- Se visualiza el contenido del buffer y se ejecuta.
 - /.- No se visualiza el comando previamente a la ejecución.
- El buffer mantiene una sentencia SQL y es sobrescrito cuando se introduce una nueva sentencia SQL.

FUNCIONES DE EDICIÓN

- SQL*PLUS también nos permite editar la sentencia SQL actual (almacenada en el buffer) mediante un sencillo editor de línea.
 - LIST** Lista el contenido completo del buffer
 - LISTn** Lista la línea n para su posterior edición, convirtiéndola en la línea actual.

APPEND Añade una cadena a la línea actual.

DEL Borra la línea actual.

INPUT Espera la introducción de líneas adicionales que serán colocadas inmediatamente después de la línea actual.

CHANGE Reemplaza en la línea actual la cadena_1 por la cadena_2. Dentro de cadena_1 puede utilizarse '....' para representar cualquier cadena de caracteres, incluyendo las de valor nulo.

- Todos los comandos, exceptuando DEL, pueden ser abreviados utilizando tan sólo su primera letra.

POSIBILIDADES DEL SQL*PLUS

- Tiene capacidad para crear archivos de comando SQL*PLUS, y generar informes de alta complejidad.
- Existe una interfaz de SQL*PLUS con el editor del sistema operativo muy buena.
- Utiliza archivos SPOOL y DE COMANDO.
- Genera informes.

INTERFAZ CON EL EDITOR DEL SISTEMA.

- SQL*plus, creará por defecto un archivo, que almacenará el contenido de la memoria intermedia.
- Simultáneamente a la modificación del archivo, se va alterando el contenido del buffer y al abandonar el editor, se devuelve el control a SQL*PLUS.

UTILIZACIÓN DE ARCHIVOS

ARCHIVOS DE SPOOL

- Los resultados generados por SQL*PLUS pueden ser capturados permanentemente con el comando SPOOL. Esta permite copiar cuanto aparece en pantalla a un archivo del sistema operativo.
- La acción de SPOOL continuará hasta que se utilice un nuevo comando SPOOL.
- SPOOL OFF Cierra el archivo creado por SPOOL.

- **SPOOL OUT** Envía directamente el contenido del archivo SPOOL a la impresora del sistema.
- Cuando acabe con una sesión de SQL*PLUS, todos los archivos creados con la instrucción SPOOL, se cerrarán automáticamente.

ARCHIVOS DE COMANDO

Un archivo de comando es un conjunto de comandos de SQL que pueden ir acompañados de otros comandos de SQL*PLUS almacenados en un archivo del sistema operativo, y que se ejecutan secuencialmente uno tras otro, al lanzar la ejecución del archivo.

Estos se crean de la siguiente manera:

- SQL*PLUS permite salvar el contenido de la memoria intermedia (que es una instrucción SQL) mediante el uso del comando SAVE.
- El archivo "salvado" puede ser recuperado y ejecutado inmediatamente utilizando el comando START.

II.4.1.3.3. BLOQUE s, PRODUCTOS DE CONECTIBILIDAD.

Dentro de estos productos encontramos el SQL*NET, quien es un mecanismo de ORACLE, para trabajar como interface con el protocolo de comunicación utilizado por la red que facilita el proceso distribuido y las B.D. distribuidas. Los protocolos de comunicación definen el camino por el cual el dato es transmitido y recibido a través de la red.

En un medio ambiente de red, un servidor de B.D. ORACLE comunicado con una estación de trabajo cliente y otros servidores de B.D. ORACLE, utilizan un software llamado SQL*NET, quien soporta comunicaciones sobre la mayoría de los protocolos de red, incluyendo los soportados por PC LAN usados por el más largo sistema de Mainframes.

Usando SQL*NET, la aplicación desarrollada no tiene que relacionarse con las comunicaciones de red en una aplicación de B.D. Si un nuevo protocolo es usado, el administrador de la B.D. realiza algunos cambios inadvertidos, mientras la aplicación continua funcionando sin necesidad de modificaciones ó si la necesita, estas son muy pequeñas. SQL*NET, usa protocolos de comunicación ó interfaces de aplicación programable (APIs), soportada por el amplio rango que poseen las redes en una B.D. distribuida ó en un proceso distribuido para ORACLE. Un protocolo de comunicaciones es un conjunto de estándares, en un

software implementado que gobierna la transmisión del dato a través de la red, esto significa que se establecen remotamente procesos a procesos vía protocolo de comunicaciones.

COMO TRABAJA SQL*NET

Los manejadores de SQL*NET proveen una interface entre los procesos ORACLE de un servidor de la B.D. y un usuario de procesos, sobre las herramientas ORACLE que se encuentran en otras computadoras de la red.

Los manejadores de SQL*NET toman parámetros de SQL de la interfaz de las herramientas ORACLE y los empaqueta para transmitirlos a ORACLE, por medio de uno de los protocolos soportados en estándares industriales con un nivel alto de aceptación. Todo esto esta realizado independientemente del sistema operativo de la red. Obsérvese la Figura II.4.1.6. SQL*NET para comprender mejor.

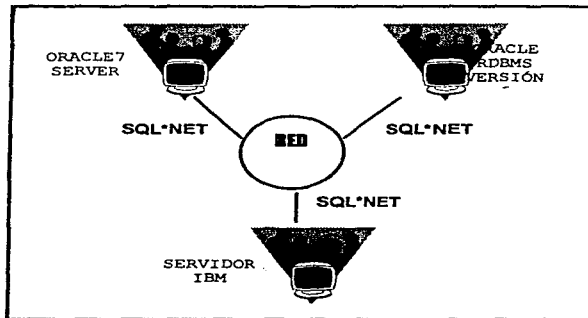


Figura II. 4.1.6. SQL*NET

II.4.1.3.4 BLOQUE 9, UTILIDADES DE LA BASE DE DATOS.

SQL*DBA

SQL*DBA es una herramienta que ayuda a la administración y monitoreo de la base de datos. Éste se puede utilizar en el desarrollo de las siguientes tareas:

- Principiar y Finalizar una sesión en ORACLE.
- Montar, desmontar, abrir y cerrar una B.D.
- Monitorear el tiempo real y el rendimiento de una B.D.
- Realizar respaldos y recobrar archivos y datos de una B.D.
- Ejecutar cualquier sentencia SQL.

Todas estas operaciones pueden ser realizadas tanto en una B.D. local como remota. Debido a que el manejo del SQL*DBA, puede dañar la seguridad de la base, pocos usuarios deben tener acceso a esta. Usualmente el principal usuario de esta aplicación es el ADMINISTRADOR.

Algunos comandos de SQL*DBA son privilegios del sistema y requieren de un acceso especial al sistema. Por ejemplo:

- STARTUP
- SHUTDOWN
- CONNECT INTERNAL.

Cuando un usuario invoca algún comando del SQL*DBA, éste checa si su acceso es permitido para esto no utiliza el USERNAME, pues este no ha sido introducido, sino que checa el sistema operativo actual en el que corre el SQL*DBA.

II.4.2. ESTRUCTURA DE LA BASE DE DATOS.

Esta sección describe la arquitectura de la B.D. de ORACLE, incluyendo la estructura física y lógica que la constituyen.

Esto provee de soluciones entendibles de ORACLE para controlar su disponibilidad, la separación de la estructura de datos lógica y física, y un control detallado del manejador de espacio.

CAPITULO II

Sistema manejador de B.D. El sistema Manejador de B.D. ha desarrollado desde los modelos jerárquicos hasta los relacionales y de red. Hoy en día el más aceptado es el modelo relacional Este modelo tiene 3 aspectos.

Relacionales

ESTRUCTURAS: Las estructuras son bien definidas como objetos que almacenan datos de la B.D. Las estructuras y los datos contenidos dentro, pueden ser manipulados por operaciones.

OPERACIONES: Las operaciones, son acciones definidas claramente para manipular datos y estructuras de la B.D. Las operaciones en una B.D. deben unirse en un predefinido conjunto de reglas de integridad.

REGLAS DE INTEGRIDAD: Estas son leyes que deciden que operaciones son permitidas sobre el dato y sobre la estructura de la B.D..

El sistema manejador de B.D. ofrece beneficios como:

- Independencia del almacenamiento físico del dato y la estructura lógica de la B.D.
- Acceso fácil y variado a todos los datos.
- Flexibilidad completa en el diseño de la B.D.
- Reducción de almacenamiento de datos y de las redundancias.

ORACLE es una colección de datos que son tomados como una unidad . El propósito general de ésta, es almacenar y recoger información relacionada entre si.

Base de datos Abierta y Cerrada: ORACLE, puede ser abierta o cerrada, es decir accesible ó no accesible. Normalmente es una B.D. abierta, lista para usarse. Pero en algunos casos es cerrada, para específicas funciones administrativa, que requieran de datos protegidos.

Estructura de B.D. ORACLE, cuenta con estructuras tanto lógica como física. El almacenamiento físico del dato puede ser manejado sin

Estructura física de la B.D.

afectar el acceso del almacenamiento lógico de las estructuras. La estructura física de la B.D. de ORACLE esta determinada por los archivos del SO que la constituyen. Cada una, esta compuesta por 3 tipos de archivos: Uno ó más archivos de dato, 2 ó más archivos de bitácora (redo log files), y uno ó mas archivos de control. Los archivos de la B.D. dan el almacenamiento físico para información de la B.D.

II.4.2.1. ESTRUCTURA LÓGICA DE LA BASE DE DATOS

Esta se determina por:

- Una ó más tablas de espacios (TABLESPACES).
- El esquema de objetos de la B.D. (Tablas, vistas, índices, clusters, procedimientos almacenados , secuencias).

La estructura de almacenamiento lógico incluyendo las tablas de espacio, segmentos y extensiones, ordenan el espacio físico que usará la B.D. el esquema de objetos y la relación entre ellas, forman el diseño relacional de la B.D.

ESTRUCTURA LÓGICA:

A continuación se explicará la estructura lógica, incluyendo las tablas de espacio, los esquemas de objeto, los bloques de dato, extensiones y segmentos.

TABLAS DE ESPACIO (TABLESPACES):

Una B.D. esta dividida en unidades de almacenamiento lógico, llamadas TABLESPACES (Tabla de espacios). Estas son utilizadas para agrupar estructuras lógicas relacionadas entre si. Por ejemplo, comúnmente una tabla de espacios agrupa todos los objetos de las aplicaciones para simplificar ciertas operaciones administrativas

II.4.2.1.1. BASES DE DATOS, TABLAS DE ESPACIO Y ARCHIVOS DE DATO.

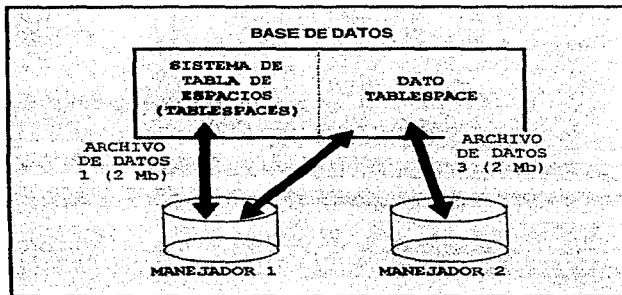


Figura II.4.1.7 Estructura Lógica De Una B.D. (TABLESPACES)

La figura anterior ilustra:

Cada B.D. es lógicamente dividida en una ó más TABLESPACES.

Uno ó más archivos de dato son explícitamente creados por cada TABLESPACE para almacenar físicamente el dato en todas las estructuras lógicas de un TABLESPACE.

El tamaño combinado de un archivo de datos de un TABLESPACE, es el total de la capacidad almacenada de el TABLESPACE (Un sistema tiene 2 Mb como capacidad de almacenaje, mientras un dato tiene 4 Mb).

La capacidad de almacenamiento de un TABLESPACE de la B.D. , es la capacidad total de almacenamiento de una B.D. (6 Mb).

**ESQUEMAS Y
ESQUEMAS DE
OBJETO**

Un esquema, es una colección de objetos. Los Esquemas de Objetos, son estructuras lógicas que hacen referencia directamente al dato de la B.D. Incluyen estructuras como tablas, vistas, secuencias, procedimientos de almacenamiento,

sinónimos, índices, clusters y ligas entre B.D. (No existe una relación entre una B.D. y un esquema; los objetos en el mismo esquema pueden estar en distintos TABLESPACES y los TABLESPACES, pueden contener objetos de diferentes esquemas.

TABLAS: Una tabla es una unidad básica de almacenamiento de datos en una B.D. de ORACLE. Los datos de la tabla son almacenados en renglones y columnas. Cada tabla es definida con un nombre de tabla y un conjunto de columna. A cada columna se le asigna un nombre y un tipo de dato (caracter, numérico, etc.) Una vez que la tabla esta creada, pueden ser insertados datos válidos dentro de ella, mismos que pueden ser borrados, buscados ó actualizadas.

VISTAS: Una vista es una presentación exacta de un dato en una ó más tablas. Una vista también puede ser considerada como una "búsqueda de almacenamiento".

Las vistas no pueden contener o almacenar datos, estas reflejan los datos de las tablas en que son basadas. Las tablas Base, pueden a su vez ser tablas ó vistas.

Como las tablas, las vistas pueden realizar búsquedas.

Una vista es utilizada comúnmente para:

- Proveer y adicionar niveles de seguridad a la tabla para restringir el acceso a un determinado conjunto de renglones y columnas de la tabla.
- Encubrir la complejidad del dato. Por ejemplo, una vista simple puede ser usada para crear una relación y será desplegada en una relación de columnas ó renglones en tablas múltiples.
- Simplifica comandos para el usuario. Las vistas permiten al usuario seleccionar información de tablas múltiples sin acudir al usuario que actualmente sabe como se desarrolla una sub-

busqueda.

- Presenta los datos en diferentes perspectivas de la tabla base. Por ejemplo en una vista se permite renombrar columnas sin afectar la tabla en donde se esta basando esa vista.
- Almacena búsquedas múltiples. Por ejemplo, una búsqueda debe desarrollar cálculos extensos con la información de una tabla. Para salvar esta búsqueda como vista, los cálculos son ejecutados solo cuando la vista es solicitada.

SECUENCIAS:

Una secuencia genera una lista consecutiva del número exclusivo de columnas de la tablas de la B.D.. La secuencia simplifica la aplicación, programando la generación automática de un valor numérico único, para los renglones de una tabla ya sea simple ó múltiple.

Por ejemplo, asumimos que 2 usuarios simultáneamente insertan un nuevo empleado en la tabla EMP, Usando una secuencia para generar un número único de empleado en la columna EMPO, ninguno de los usuarios tendria que esperar para obtener el número siguiente. La secuencia automáticamente genera el valor correcto para cada usuario.

Las secuencias numéricas, son independientes de las tablas, entonces la misma secuencia puede ser usada por más de una tabla. Después de su creación, una secuencia puede ser accedada por varios usuarios para generar un número actual de la secuencia.

UNIDADES DE PROGRAMA:

Este término, lo utilizaremos para referirnos a: procedimientos almacenados, funciones y paquetes.

Una **FUNCIÓN** ó **PROCEDIMIENTO**, es un conjunto de declaraciones de SQL y PL/SQL agrupadas en una unidad ejecutable para desarrollar una tarea específica. (SQL es abarcado en el apéndice de esta tesis y PL/SQL se trato

previamente en este capítulo).

Los PROCEDIMIENTOS y las FUNCIONES permiten combinar la facilidad y flexibilidad del SQL con la funcionalidad de los procesos de un lenguaje de programación estructurado. Usando PL/SQL se permite que los procedimientos y funciones sean definidos y almacenados en la B.D. para su uso continuo.

Un procedimiento y una función, son idénticos, exceptuando que la función siempre regresa un valor único donde fué llamada, y que el procedimiento no.

**PAQUETES
(PACKAGES):**

Los paquetes cuentan con un método para encapsular y almacenar procedimientos relacionados, funciones y otras construcciones de paquetes, conjuntamente como unidad, en la B.D. Mientras un paquete, proporciona una administración de la B.D. ó un beneficio organizacional para el desarrollo de una aplicación, también incrementan la funcionalidad de la B.D.

SINÓNIMOS:

Un sinónimo, es un alias para una tabla, vista, secuencia ó unidad de programa. Un sinónimo no es un objeto por si solo, pero en cambio, es una referencia directa a un objeto.

Se utiliza para:

- Ocular el nombre y propietario real de un objeto.
- Permitir acceso público a un objeto.
- Permitir una transparente ubicación de tablas, vistas ó unidades de programas, desde una B.D. remota.
- Simplifica las instrucciones de SQL para los usuarios de la B.D.

Estos puede ser públicos ó privados, un usuario individual puede crear un SINÓNIMO PRIVADO, que será utilizado solo por el. El Administrador de la B.D., debe crear sinónimos públicos que hacen los objetos del esquema base accesibles en general para cualquier usuario de la B.D.

ÍNDICES y CLUSTERS: Estos, son estructuras opcionales asociadas con tablas que pueden ser creadas para incrementar el desarrollo de la recuperación de datos.

ÍNDICES Ó ÍNDICES: Esos son creados para incrementar el desarrollo de la recuperación de datos. Cualquier índice, ayuda a localizar información específica más rápido que cuando éste, no existe, en ORACLE un índice accesa más rápido al path de la tabla de datos. Cuando procesa una solicitud, ORACLE puede usar algunos o todos los índices permitidos para eficientar la localización de los renglones solicitados. Los índices son creados en una ó más columnas de la tabla. Una vez creados se mantienen y usan automáticamente por ORACLE. Los cambios a la tabla de datos, ya sea altas, bajas ó actualizaciones, son incorporados automáticamente a los índices, con transparencia total para el usuario.

Los índices son física y lógicamente independientes de los datos. Éstos pueden ser borrados y creados en cualquier momento sin afectar las tablas ó cualquier otro índice.

CLUSTERS: Estos son un método opcional para almacenar datos en una tabla. Los clusters son grupos de una ó más tablas físicas almacenadas juntas pues comparten columnas en común y por lo regular se utilizan juntas. Debido a que renglones en común son físicamente almacenados juntos, el tiempo de acceso en disco se incrementa.

Las columnas relacionadas de una tabla en un cluster son llamadas "cluster key" (cluster llave).

El cluster llave es indexado, entonces los renglones del cluster son recuperados con un mínimo de entradas y salidas permitidas. Debido a que el dato en un CLUSTER LLAVE dentro de un

CLUSTER INDEXADO es almacenado solo una vez para muchas tablas, un cluster almacena un conjunto de tablas más eficientemente que cuando se almacenan las tablas individualmente. La figura siguiente ilustra como son almacenados los datos físicamente en un cluster y fuera de el .

Los clusters, también pueden incrementar la recuperación de los datos, dependiendo de la distribución de los datos y de que operaciones de SQL se desarrollan continuamente en un dato que se encuentre en cluster. En particular, las talas en cluster que son buscadas en una relación, benefician al usuario de los clusters, pues los renglones en común de las tablas relacionadas son recuperados en la misma operación de entrada y salida. Al igual que los índices, los cluster no afectan el diseño de la aplicación. Si la tabla es ó no parte de un cluster, son transparentes para el usuario y sus aplicaciones., los datos almacenados en una tabla en cluster, son accesados via SQL de igual forma que son almacenados en una tabla fuera de un cluster.

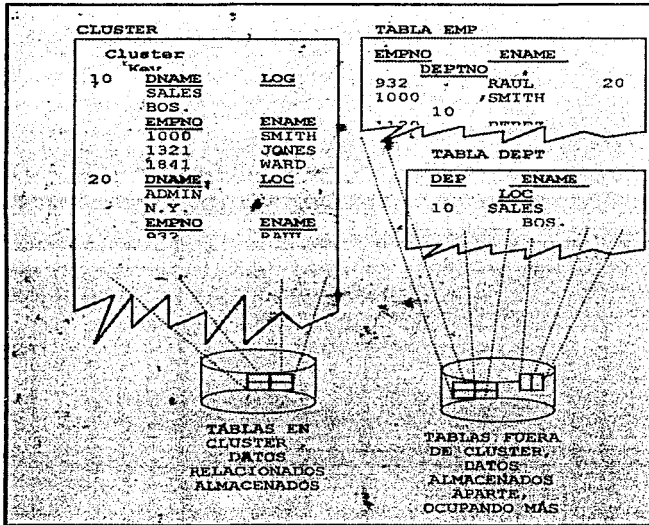


Figura IL4.1.8. Tablas en Cluster y fuera de ellos.

LIGAS EN UNA BASE DE DATOS.

Una liga en la B.D. es un objeto que describe un "path" de una B.D. a otra. Una liga es usada implícitamente cuando una referencia es hecha a un objeto global en una B.D. distribuida (ver B.D. DISTRIBUIDA).

BLOQUES DE DATO, EXTENSIONES Y SEGMENTOS.	ORACLE, permite un control detallado de el uso del espacio en disco a través de las estructuras de almacenamiento lógico, incluyendo bloques de dato, extensiones y segmentos.
BLOQUES DE DATO ORACLE:	En un nivel detallado el dato en una B.D. ORACLE, es almacenado en un bloque de datos. Un bloque de datos corresponde a un número específico de bytes en un espacio físico en disco. El tamaño de un bloque de datos es especificado por cada B.D. ORACLE, cuando ésta se crea.
EXTENSIONES	El nivel siguiente al espacio lógico en una B.D. es llamado extensión. Ésta es un número específico de bloques de datos continuos que se encuentran en una locación sencilla usando para el almacenamiento un tipo específico de información.
SEGMENTOS	<p>El nivel de almacenamiento lógico en una B.D. arriba de la extensión, es conocido como segmento.</p> <p>Un segmento, es un conjunto de extensiones permitidas por una cierta estructura lógica.</p> <p>Los diferentes tipos de segmento incluyen:</p> <p>SEGMENTO DE DATOS: Cada tabla que no se encuentra en cluster tiene segmentos de datos.</p> <p>SEGMENTOS INDEXADOS. Cada índice tiene un segmento indexado en donde almacena sus datos.</p> <p>SEGMENTOS DE RECUPERACIÓN O ROLLBACK: Uno ó más segmentos de este tipo, son creadas por el administrador de la B.D. para un almacenamiento temporal información de respaldo.</p> <p>SEGMENTOS TEMPORALES: Estos son creados por ORACLE, una instrucción de SQL necesita una área de trabajo temporal para completar una ejecución. Las extensiones de segmentos temporales son regresadas al sistemas para usos posteriores.</p>

ORACLE trabaja espacio para todos los tipos de segmento en extensiones, debido a esto, pueden o no estar seguidas en el disco.

II.4.2.2. ESTRUCTURA FÍSICA DE UNA BASE DE DATOS

A continuación se explicarán las estructuras físicas de ORACLE, archivos de dato, archivos bitácora (REDO LOG FILES) y archivos de control que son las partes que forman la estructura física de una B.D.

ARCHIVOS DE DATO. Las B.D. ORACLE, tienen uno ó más archivos de dato físicos. Un archivo de datos contiene todos los datos de la B.D. Los datos de la estructura lógica de la B.D. como tablas e índices, son almacenados físicamente en los archivos de dato asignados para la B.D.

Las siguientes son CARACTERÍSTICAS de los archivos de dato:

- Los archivos de dato, pueden ser asociados con solo una B.D.
- Una vez creado, no puede cambiar su tamaño.
- Uno ó más archivos de dato de la unidad lógica de una B.D. almacenada recibe el nombre de TABLESPACE, de la que ya se habló anteriormente.

ARCHIVOS BITÁCORA (REDO LOG FILES) Cada B.D. ORACLE, tiene un conjunto de 2 ó más archivos REDO LOG. El conjunto de este tipo de archivos, es conocido comúnmente como el REDO LOG de las B.D. Todos los cambios hechos a la B.D. son grabados en el REDO LOG. Como una prevención de fallas, las modificaciones a los datos son permanentemente escritas en los archivos de dato, los cambios pueden ser recuperados del REDO LOG y así los trabajos nunca se perderán.

ARCHIVOS DE CONTROL: Todas las B.D. ORACLE, cuentan con archivos de control. Un archivo de control almacena la estructura física de la B.D. Por ejemplo, estos

contienen la información siguiente:

- Nombre de la B.D.
- Nombre y localización de los archivos de dato y redo log (archivos bitácora) con los que cuenta la B.D.
- Señala el tiempo de creación de la B.D.

Al igual que los archivos bitácora (REDO LOG), ORACLE permite que los archivos de control sean respaldados para su misma protección.

**EL
DICCIONARIO
DE DATOS.**

Cada B.D. cuenta con un Diccionario de datos, éste es un conjunto de tablas y vistas que son usadas como de solo-lectura y hacen referencia a la B.D. El diccionario almacena información de la estructura tanto física como lógica de la B.D. y además almacena información como:

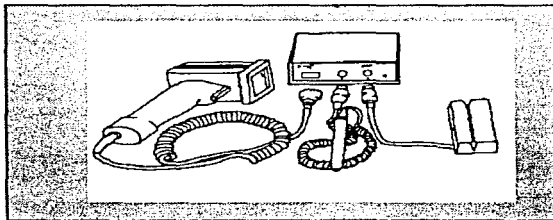
- Usuarios válidos para la B.D. ORACLE.
- Información sobre la integridad definida por las tablas en la B.D.
- Cuanto espacio es asignado para el esquema de objetos y cuanto de éste será usado.
- Un Diccionario de Datos, es creado cuando la B.D. es creada. Con exactitud, refleja el estado de la B.D. en todo momento, el Diccionario de Datos es básico para la operación de la B.D., quien se basa en éste para guardar, verificar y llevar a cabo el buen funcionamiento del sistema. Por ejemplo, durante la operación de la B.D. ORACLE, lee el dato del diccionario para verificar que existe en el esquema de objetos y que el usuario tenga permitido el acceso.

II.5 INTERFACES.

II.5.1. BAR CODE AND MAGNETIC STRIPE READER (MODEL 2000/2002).

El modelo 2000/2002 del lector de código de barras y cinta magnética combina características de varios lectores en un sistema sencillo y compacto. El lector se instala rápidamente entre el teclado y la computadora. Los datos se introducen a la computadora como si estos fueran introducidos desde el teclado. No son necesarios cambios de software o hardware.

El lector acepta una gran variedad de dispositivos de entrada tales como LASER, CCD, LAPIZ, LECTOR DE CINTA MAGNETICA, SLOT READER, y ENTRADA SERIAL RS-232. El modelo 2000/2002 acepta hasta cuatro dispositivos de entrada, maximizando la versatilidad del sistema lector. Todas estas características hacen del modelo 2000/2002 un dispositivo colector de datos ideal de gran rendimiento.

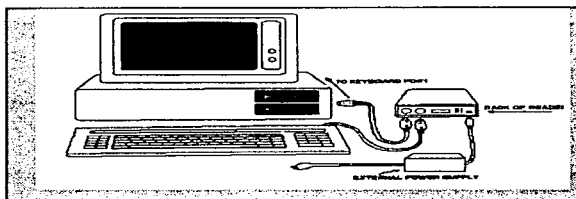


Aditamentos.

Características:

- Para la computadora el código de barras, cinta magnética, y datos seriales parecen como datos introducidos desde el teclado.
- El lector reconoce y lee automáticamente los siguientes tipos de códigos de barras:

- Código 39
- Código 39 extendido (Full ASCII)
- Interleaved 2 de 5
- UPC-A, UPC-E(0), UPC-E(1)
- EAN-8, EAN-13
- Suplementos UPC & EAN (Carácter 2 y 5)
- La luz Power/Ready indica el estatus de scanner(examinar)
- Lector de cintas magnéticas (Tarjetas de crédito, etc).
- Soporta múltiples dispositivos de entrada (Laser, CCD, lápiz, Slot reader)
- Acepta datos del puerto de entrada RS-232 (escalas electrónicas, terminales portátiles, etc).



Conexión con la computadora.

LECTURA DE CODIGO DE BARRAS.

Lápiz: El lápiz debe sostenerse con una inclinación de 10° a 30° de la vertical (figura II.4.1.9) y pasarse rápidamente a través de la etiqueta como si se tratara de dibujar una línea en medio de esta. Este movimiento puede hacerse tanto de izquierda a derecha como de derecha a izquierda y los datos se introducirán de manera correcta.

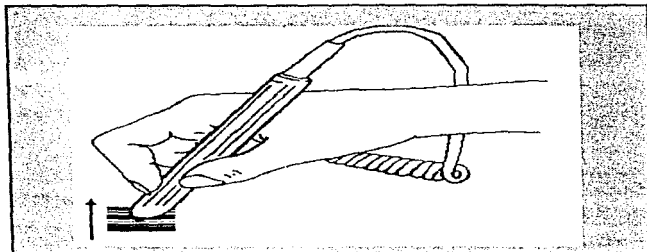


Figura IL4.1.9. Lectura de código de barras con lápiz.

Si el movimiento se ha hecho de manera correcta, se escuchará un pequeño "BEEP". Cuando la luz de Power (energía) se torna verde el lector está lista para procesar otra etiqueta.

Gatillo tipo Laser: Se debe colocar la pistola a una distancia de tres a seis pulgadas. Sin embargo, la distancia máxima varía dependiendo de la densidad de la etiqueta. Generalmente estas distancias varían entre 3 y 18 pulgadas.

Una vez que se ha presionado el gatillo se emitirá un rayo de luz roja. Este rayo debe cubrir completamente la etiqueta y una parte de área blanca en cada uno de los datos para que el lector pueda identificar el fin de la etiqueta en ambos extremos.

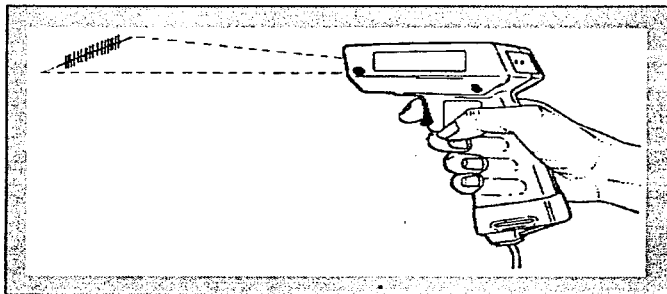


Figura II.4.1.10. Lectura de código de barras con gatillo tipo láser.

En este caso también se escuchará un pequeño "BEEP", y la luz de "DECODE" (decodificar) se encenderá. Entonces el código de barras se transmitirá a la computadora. Una vez hecho esto, la luz de "POWER" cambiará a verde y el lector estará listo para procesar otra etiqueta.

Si el láser no ha leído la etiqueta en un lapso aproximado de un segundo, el rayo de luz roja se apagará.

Lector de ranura (lector de distintivo): Se debe colocar la tarjeta que contiene el código de barras de forma que este código quede hacia abajo y de frente a la flecha junto a la ranura del lector. Entonces se debe correr la tarjeta a través de la ranura del lector. Este movimiento puede hacerse ya sea de derecha a izquierda o de izquierda a derecha y los datos se introducirán correctamente sin ningún problema. En todo momento la tarjeta debe mantener el contacto con la base del lector.

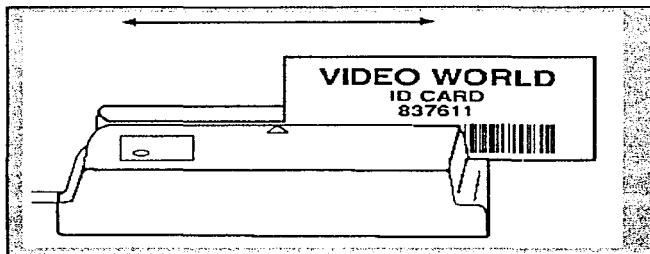


Figura IL4.1.11 Lectura con lector de ranura.

Después de esto se escuchará un pequeño "BEEP" y los datos serán transmitidos a la computadora. Cuando la luz de "POWER" se torne verde, el lector estará listo para procesar otra tarjeta.

Se debe tener cuidado de que el centro del código de barras esté a media pulgada del borde inferior de la tarjeta.

CONFIGURACION DEL MODELO 2000/2002.

El modelo 2000/2002 trae configuraciones por default para:

- | | |
|-----------------------|---------------------------|
| - CODIGO 39 | - PUERTO SERIAL |
| - UPC | - DIAGNOSTICOS |
| - EAN | - PREAMBULOS |
| - SUPLEMENTOS UPC/EAN | - CARACTER DE TERMINACION |
| - INT. 2 DE 5 | - CINTA MAGNETICA # 1 |

- CODABAR
- CODIGO 128
- CODIGO 93
- MSI/PLESSEY
- CODIGO 11
- OPCIONES # 1 Y # 2
- CINTA MAGNETICA # 2
- BEEP
- TIPO DE COMPUTADORA
- EDITOR DE DATOS
- VELOCIDAD DE TRANSMISION

Todas estas configuraciones de default pueden ser cambiadas de acuerdo a las necesidades que se tengan. Esto se puede hacer fácilmente con la ayuda de una hoja auxiliar que viene como suplemento del manual del lector.

Mediante dicha hoja uno puede elegir entre las distintas opciones disponibles para cada una de las configuraciones anteriormente mencionadas. Toda esta configuración se hace mediante código de barras, es decir, el suplemento contiene código para configurar cada una de las opciones con sólo utilizar alguno de los dispositivos de lectura antes mencionados.

Cuando se necesita dar de alta varios lectores con la misma configuración, se tiene la ventaja de que se puede configurar el primer lector y después copiar esta información a los demás. Esta información puede ser copiada de computadora a lector y de lector a lector utilizando el puerto serial RS-232. En el primer caso se necesita además un software de comunicaciones.

El puerto serial RS-232 localizado en el panel del lector acepta datos seriales ASCII y los retransmite a la computadora a través de una interface con el teclado. La entrada serial de datos aparece como si fuera introducida desde el teclado.

Con las teclas de función habilitadas, el decodificador puede aceptar un determinado caracter ASCII y transmitir a la computadora su tecla de función correspondiente.



CAPITULO III

PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

CAPITULO III PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

III.1. ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA.

La Coordinación de Servicios de Cómputo como entidad integrante de la Universidad Autónoma Metropolitana unidad Xochimilco, tiene por objetivo proporcionar los servicios de cómputo consciente de la necesidad de racionalizar y optimizar la infraestructura disponible para apoyar el desarrollo armónico de la comunidad universitaria.

ORGANIZACIÓN.

La Coordinación de Servicios de Cómputo consta de tres secciones y una área las cuales son:

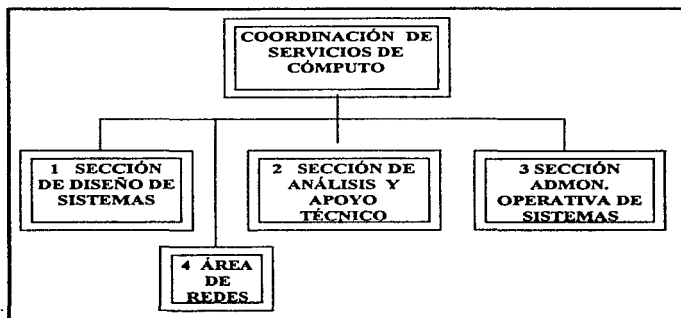


Figura III.1.1 Organización del Centro de Cómputo

Sus funciones según el número correspondiente en el diagrama son:

- 1.-Es la encargada del análisis, diseño, implementación, actualización y mantenimiento de los sistemas de cómputo que requiera la Institución.

CAPITULO III

- 2.-Esta sección adquiere, actualiza y adecua paquetes de software para el equipo, organiza cursos de apoyo a las áreas académicas y asesora a todos los usuarios.
- 3.- Es la encargada de mantener el buen funcionamiento del equipo, así como el asignar los recursos necesarios a los usuarios, conforme a las normas y procedimientos establecidos.
- 4.-El área de redes, se encarga de mantener en buen funcionamiento, adecuación y administración de las redes digitales con que se cuenta en toda la Universidad.

El centro de cómputo se encarga de:

- Asignación de números de cuenta con acceso a INTERNET a:
 - Profesores
 - Investigadores y sus colaboradores.
 - Alumnos inscritos
- Reservación y reasignación de equipo a:
 - Usuarios con número de cuenta e identificación.
- Préstamo de material de consulta :
 - Usuarios con número de cuenta e identificación.
- Impresiones remotas en laser y a color.
- Digitalizaciones de imágenes y textos.
- Impresión de planos:
 - Impresiones en Design - Jet a color.
- Detección y eliminación de virus informaticos.

En la actualidad los trámites para la prestación de algún servicio en el centro de cómputo son realizados manualmente o semiautomatizados lo que genera lentitud y pérdida de tiempo a los usuarios y a los prestadores del servicio, por lo cual se ve la necesidad de la automatización de dichos trámites.

III.2. REQUERIMIENTOS DEL USUARIO.

Desde la antigüedad el hombre a tenido necesidad de información, a inventado dispositivos que le ayudan a calcular y procesar los datos, aplicando varios métodos para la planeación y resumiéndolos a un área específica para la solución de un determinado problema.

Hoy en día toda institución tiene una logística (planes y políticas) así como una estrategia que debe satisfacer las necesidades de la misma.

Esto trae como consecuencia la necesidad de llevar un control riguroso de sus recursos materiales (por ejemplo: El Centro de Cómputo.), para saber cuanto se tiene y donde se tiene.

Por ello requiere de un proceso de sistematización y automatización que la lleve a una mejora en las actividades que realiza, éstas deben satisfacer las necesidades de la institución.

Concretándonos al estudio en cuestión, el Centro de Cómputo para su automatización nos presenta los requerimientos descritos enseguida:

- Un sistema confiable.

Con el que se asegura la integridad de los datos que maneja el centro de cómputo

Con claves de acceso y protección de datos.

Con eficiencia en el préstamo de servicios o en las operaciones que realiza.

- Rápido

Evita la pérdida de tiempo tanto para los usuarios como para los prestadores de servicio por la simplicidad de sus operaciones.

Lo que en el sistema actual tarda de 10 a 30 minutos se puede hacer en un par de minutos con el lector de código de barras y una estructura simple de programación.

- Fácil de manipular.

Que utilice secuencia de pasos sencillos, fáciles de aprender y aplicar; además de permitir un mantenimiento y/o una actualización que no requiera de procedimientos complejos.

- Que aproveche los recursos con los que cuenta el centro de cómputo.

Actualmente los procesos que han sido automatizados, no aprovechan a plenitud los recursos que el Centro de Cómputo ha adquirido:

- MAINFRAME HP9000 827s con sistema operativo UNIX versión 9.14

- ORACLE versión 4.9 manejador de base de datos.

- BAR CODE AND MAGNETIC STRIPE READER MODELS 2000/2002

- Que aproveche el código de barras con que cuentan las credenciales del plantel.

CAPITULO III

Para facilitar el manejo de datos y evitar la pérdida de tiempo en los trámites que se realizan (puntos mencionados con anterioridad) mejorando la calidad del servicio que se presta.

- Mejorar la calidad del producto final.

Los reportes y los comprobantes de servicio deben ser de mayor calidad en su presentación y contenido, permitiendo un fácil manejo de los mismos.

Además de que se solicita otro tipo de reportes que no se manejan actualmente para ayudar a la administración del centro (histórico y estadístico).

III.3. RECOPIACIÓN DE INFORMACIÓN.

Las técnicas que utilizamos para obtener información del usuario fueron las de entrevistas y aplicación de cuestionarios.

Se aplicó un cuestionario al responsable del centro en cuestión, y otro a las personas que realizan las actividades necesarias para la prestación de servicio, así como para el administrador del Sistema HP9000. Sus nombres y cargos son listados a continuación.

- Raúl Urban Ruiz

Coordinador del Centro de Cómputo.

Se encargan de asignar tiempo y máquinas en las salas de usuarios:

- Srita. Norma Breña Ramírez.

Auxiliar de Mesa.

- Sra. María del Rosario Breña Ramírez.

Auxiliar de Mesa

Se encargan del mantenimiento de equipo de cómputo de las diferentes coordinaciones del plantel:

- Ing. Demetrio Pérez Aguilera.

Jefe del SAOS (Sección de Administración Operativa y de Sistemas).

- Sr. Carlos Rincón Guillén.

Asesor técnico.

- Sr. Jorge Rocha Arellano.

Operador.

Se encargan de llevar un control de cuentas de usuarios y proyectos:

- Sra. Patricia Camacho Solís.
Responsable de Mesa.
- Sra. Graciela Martínez.
Responsable de Mesa.

Se encarga de llevar la administración de las salas de cómputo:

- Sr. Ismael Rivera Sandoval.
Jefe del SAT (Sección de Análisis y Apoyo Técnico).

A continuación se incluyen los cuestionarios que se aplicaron para este fin.

1.- CUESTIONARIO APLICADO AL COORDINADOR DEL CENTRO DE CÓMPUTO

- 1.-¿ CUALES SON LAS FUNCIONES QUE DESEAN AUTOMATIZAR?**
- 2.-¿ EXISTE UN PROCEDIMIENTO PARA CADA UNA DE LAS FUNCIONES A AUTOMATIZAR?**
- 3.-¿ CUANTAS PERSONAS ESTÁN DESIGNADAS PARA DESARROLLAR ESAS ACTIVIDADES?**
- 4.-¿ CADA PERSONA REALIZA UNA FUNCIÓN ESPECÍFICA?**
- 5.-¿ CUANTAS DE ESAS PERSONAS CONOCEN LOS PROCEDIMIENTOS Y REALIZAN TODO EL PROCESO?**
- 6.-¿ CUALES SON LOS PROBLEMAS QUE PRESENTAN LOS PROCEDIMIENTOS ACTUALES ?**
- 7.-¿ CON QUE EQUIPO DE COMPUTO SE CUENTA PARA EL DESARROLLO DEL SISTEMA ?**
- 8.-¿ QUE NUMERO DE USUARIOS ACCESAN NORMALMENTE EL EQUIPO ?**
- 9.-¿ CON QUE SOFTWARE SE CUENTA PARA EL DESARROLLO DEL SISTEMA ?**
- 10.-¿ ACTUALMENTE SE CUENTA CON ALGÚN SISTEMA PARA PROCESAR LA INFORMACIÓN ?**

11.- EN CASO DE EXISTIR ALGÚN SISTEMA ¿CUALES SON SUS CARACTERÍSTICAS ?

En resumen, se tuvieron este tipo de respuestas:

- 1) El control de cuentas de usuarios y proyectos , la asignación de material de computo a usuarios, la captura de los servicios de mantenimiento, el control de las salas de cursos y el control de inventarios
 - 2) Se puede decir que si pero no en todas esta documentado.
 - 3) Por actividad dos o tres personas y en el inventario participan mas de tres personas.
 - 4) Se puede decir que si, pero en algunas actividades la realizan igualmente varias personas.
 - 5) Las personas asignadas a alguna actividad conocen todo el proceso e igualmente lo realizan.
 - 6) Lentitud en el servicio e información deficiente.
 - 7) Un mainframe HP9000, PCs, así como un lector de código de barras que estamos por adquirir.
 - 8) El equipo esta destinado para el área de desarrollo de sistemas que esta conformada por 7 personas y separa todos los docentes del plantel generalmente lo ocupan para correo electrónico.
 - 9) ORACLE que esta instalado en la HP9000 así como el software de comunicaciones y emulador de terminal Pathway.
 - 10) En algunas de las actividades que se desean automatizar si.
 - 11) Se desarrollaron en una plataforma HP3000 con SPEEDWARE pero el principal problema con este es que esta empezando ha ser obsoleto por la información que se maneja actualmente además de que las cuentas se deben dar de alta en la HP9000.
- 2.- CUESTIONARIO APLICADO A PRESTADORES DE SERVICIO.**

- 1.- ¿ QUE PROCESO CONSTITUYE SU ACTIVIDAD ?
- 2.- ¿ CUANTO TIEMPO TARDA SU ACTIVIDAD ?
- 3.- ¿ SE PRESENTA RETRASOS Y PORQUE ?

- 4.- ¿ SE SATISFACEN LOS OBJETIVOS ESPECÍFICOS DE SU ÁREA, POR QUE ?
- 5.- ¿ QUE TIPO DE INFORMACIÓN MANEJA ?
- 6.- ¿ COMO MANEJA ESTA INFORMACIÓN ?
- 7.- ¿ SI EXISTE ALGÚN SISTEMA QUE REALICE SU ACTIVIDAD, QUE MEJORAS SUGIERE ?
- 8.- ¿ QUE CARACTERÍSTICAS LE GUSTARÍA QUE TUVIERA SU SISTEMA EN CASO DE AUTOMATIZARSE SU ACTIVIDAD ?

De este cuestionario se puede concluir lo siguiente:

1)

- Dar cuentas, entregar impresiones laser, entregar listados.(A usuarios en general).
- Asignar, verificar y dar de baja cuentas a los alumnos, docentes y proyectos trimestralmente, otorgar duplicados de las cuentas y actualizar proyectos.
- Asignar tiempos a los alumnos , así como préstamo de manuales.
- Asignar tiempos
- Mantenimiento a equipos y periféricos apoyo técnico a usuarios.
- Mantenimientos correctivos y preventivos a las computadoras del plantel.

2)

- No hay un tiempo promedio.
- El tiempo para dar de alta una cuenta es de aproximadamente 4 a 5 minutos
- 4 minutos por alumno.
- de 4 a 6 minutos.
- Unos 10 minutos en llenar la orden de servicio; y en realizar el mantenimiento depende del problema del equipo.
- Unos 6 minutos en capturar los datos del servicio y el tiempo en realizar la reparación del equipo depende de la falla.

3)

- Por que algunas veces falla el sistema o el programa de captura.
- Los retrasos que se presentan son por la actualización de los archivos cada trimestre en la HP9000.

CAPITULO III

- Si algunas ocasiones los alumnos no traen los requisitos que se les pide.
- Si , por que algunas ocasiones algunos alumnos no traen sus documentos completos.
- Si , por que el programa que expide las ordenes de servicio es muy complicado de manejar.
- Si , por que el sistema de captura de ordenes de servicio es muy poco amigable y esta incompleto.

4)

- Si porque se da la atención a todos los usuarios de la unidad.
- El proceso es lento pero se cumplen los objetivos de mi área de trabajo.
- Si , porque a los alumnos se les asigna su hora en las computadoras.
- Si, ya que a todos se les otorga el préstamo de equipo.
- Si, ya que se realiza el servicio a todos los equipos.
- La etapa de captura de las ordenes es lenta, pero se les presta este servicio a todos los equipos del plantel.

5)

- Credenciales o identificaciones vigentes, tiras de materias en si todos los datos personales de los usuarios del centro de computo.
- Matriculas de alumnos, números económicos de profesores y trabajadores administrativos y un catalogo de departamentos y divisiones.
- Los diferentes software que están instalados en las máquinas, así como la configuración de estas.
- Horario de servicio, el software instalado, tipo de equipo, que uso se le da a la cuenta.
- Datos del equipo como son marcas, modelos , números de serie, a quien pertenece , falla, etc.
- Los números de serie e inventario de los equipos, datos del responsable del equipo , falla del equipo , control de garantías del equipo, refacciones necesarias, etc.

6)

- Con un programa de computadora se registran las cuentas de usuarios y lo que no hace el programa lo almacenamos en archiveros.+
- Con un programa en la HP3000 y con archiveros.
- Verbal y en algunas ocasiones escrita.

- Verbal y escrita con la ayuda de archiveros.
 - Con un programa hecho en clipper para el control de este servicio.
 - Con un sistema desarrollado para la captura e impresión de ordenes de servicio.
- 7)
- Que permita la elaboración de reportes mas completos y que mejore en la parte de alta de cuentas además debe permitir llevar un control de las impresiones.
 - Que al ingresar una matricula aparezca automáticamente el dueño de la misma, como lo es el trimestre al que pertenece, si es alumno o docente, etc.
 - No existe ninguno.
 - No existe.
 - Base de datos SAOS(CLIPPER). Se requiere hacer una base de datos de fallas para hacer mas ágil la detección de las mismas.
 - Que permita realizar cambios en los registros de servicios, que permita elaborar reportes de fallas mas comunes para poder prevenirlas y que sea mas simple en su manejo.
- 8)
- Que las cuentas queden registradas en la IIP9000 para agilizar el servicio y un programa para el control de vales de impresión.
 - Que el funcionamiento fuera mas rápido, que se aproveche el código de barras de las credenciales para la agilización de atención a los alumnos.
 - Que sea rápido.
 - Velocidad y fácil manejo.
 - Que fuese fácilmente intuitivo para que los prestadores de servicio social se familiaricen rápidamente con el manejo de este. Que brinde información confiable y oportuna para la realización de reportes y estadísticas.
 - Que permita la elaboración de reportes y que mantenga la integridad de la información , además debe de ser de fácil manejo.

3.- CUESTIONARIO APLICADO AL ADMINISTRADOR DEL SISTEMA IIP9000

1.- ¿ CON QUE MECANISMOS SE CUENTA PARA PROTECCIÓN
CONTRA FALLAS DE HARDWARE ?

- 2.- ¿ QUE MÉTODOS SE TIENEN PARA RESPALDO DE LA INFORMACIÓN MANEJADA ?
- 3.- ¿ EXISTEN MÉTODOS PARA EVADIR EL SISTEMA ? ¿ POR QUE SE PRESENTAN ?
- 4.- ¿ QUE ÁREAS NECESITAN UN CONTROL ESPECIFICO ?
- 5.- ¿ CUALES SON LOS MÉTODOS DE CONTROL UTILIZADOS ?
- 6.- ¿ QUE MÉTODOS SON UTILIZADOS PARA DETECTAR LA VIOLACIÓN DE INFORMACIÓN ?
- 7.- ¿QUE MÉTODOS TIENE PARA UNA ACTIVIDAD IMPROPIA?

En resumen, las respuestas a este cuestionario son:

1) Las fallas de hardware pueden ser muy variadas, desde daño en algunos sectores del disco duro hasta la inutilización del disco duro.

Cuando el daño esta localizado en sectores del disco se detectaría por el mal funcionamiento de algunos programas del sistema de información o bien por la recuperación de datos inesperados al efectuar alguna consulta o recuperación de transacciones. Cuando esto pasa es necesario *resetear* la máquina y esperar que al hacer el *boot* los propios programas de arranque detecten el daño físico y lo corrijan en forma *semiautomática*. Se dice que es en forma *semiautomática* porque el programa (en este caso se llama *fsck*, file system check) es interactivo, es decir, solicita respuesta del operador para poder continuar con la reparación. Posterior a la revisión del o de los discos, es necesario *bajar* el respaldo más actualizado.

Cuando el daño es en la totalidad del disco sólo reemplazándolo se puede corregir el daño. (Nota: existen discos conocidos como *mirror* que precisamente actúan como espejo para tener una copia *paralela* del disco duro con el que normalmente se trabaja y sería el método ideal para tener una seguridad total de que el sistema siempre estará operando en forma óptima).

2) En un sistema HP9000 S827 se cuentan con comandos u órdenes a nivel sistema operativo tales como *tar*, *cpio* y *fsbackup*, además de un programa llamado SAM (System Administrator Manager).

Los comandos `tar`, `cpio` y `fbckup` nos sirven para efectuar respaldos desde la línea de comando, son muy flexibles y seguros, permiten realizar respaldos incrementales y totales, incluyendo y/o excluyendo rutas o directorios que así lo requieran.

El programa SAM efectúa las mismas operaciones señaladas arriba sólo que en forma *amigable*, debido al manejo de una interfaz gráfica.

3) Para poder acceder al sistema se requiere pasar dos etapas, la aceptación del *user name* y también la aceptación del *password* correspondiente al mismo usuario. Además como reglas de un sistema seguro se debe primero establecer una estructura de archivos con los permisos adecuados para propietarios y grupos, segundo forzar a los usuarios a cambiar su *password* periódicamente y por último establecer auditorías para detectar que usuarios y desde que nodo han tratado de *switchear* a modo de administrador del sistema.

4) Las áreas funcionales que requieren un control específico en un sistema de cómputo (HP9000S827) son el acceso al sistema a través de cuentas para los usuarios, monitoreo de procesos, respaldo y restauración de archivos, detección y corrección de errores en los *file systems*.

5) En lo referente al control de acceso a usuarios se asignan cuentas con el único requisito que sea para un alumno inscrito en el trimestre actual, el sistema determinará los permisos de sus archivos y directorios además de los programas y utilerías a las cuales tiene acceso determinado usuario.

El monitoreo de procesos se efectúa utilizando cualquiera de las siguientes tres utilerías: *glsnce*, *monitor* y *top*. Con estas utilerías se establecen prioridades de ejecución para los procesos y de acuerdo a tales prioridades se elevarían unas, se disminuirían otras, etc. de manera que procesos claves se desarrollen en forma óptima.

El respaldo y restauración de archivos se hace empleando el comando `tar` y `fbckup` y *frestore* como tales o a través de un programa llamado SAM; los puntos a tener en cuenta para desarrollar estas operaciones son principalmente ¿qué se va a respaldar?, ¿cuándo se va a respaldar? y ¿cómo se va a respaldar?.

CAPITULO III

La detección y corrección de errores en file systems es un asunto importante en el desempeño del sistema, debido a que datos corruptos o perdidos ocasionarían desde mal funcionamiento del sistema hasta su caída, para corregirlos se emplea el programa fsck, y en los caso extremos en que se deba formatear el disco se utiliza mediainit.

6) En mi caso específico, no manejo bases de datos por lo que desconozco herramientas para detección de violación de la información. En lo que se refiere al manejo en general del sistema la violación de la información entendida como corrupción de archivos, se detecta a través del mal funcionamiento del sistema. Además se pueden emplear utilerías para detectar problemas en red (ifconfig lan0, ping, landiag), para problemas en disco (fsck), para montaje de file systems (mount), etc.

7) Si se considera una actividad impropia aquello que afecte el desempeño correcto del sistema, los métodos para evitar esto consistirían en el monitoreo de procesos de los usuarios para determinar que actividades están realizando y con que fines, esto daría pauta para modificar prioridades de los procesos, transferir tiempo de ejecución de procesos a momentos de menor demanda del sistema, etc.

III.4. ANÁLISIS DE LA INFORMACIÓN OBTENIDA.

Del punto anterior se concluye que en la actualidad el centro de computo de la UAM XOCHIMILCO presta los siguientes servicios y el cual posee muchas desventajas que se pretenden corregir con este trabajo de tesis.

REGISTRO DE USUARIOS, PROFESORES Y PROYECTOS.

El registro de alumnos y profesores se realiza con la finalidad de llevar un control de usuarios, estos tienen derecho al uso del equipo y manuales con que cuenta el centro de computo, así como acceso a los diversos cursos que ofrece.

El registro de proyectos, es un servicio que se presta únicamente a las coordinaciones del plantel con el fin de asignar prioridades en la disposición de equipo a las personas que se dediquen a un proyecto determinado.

Ambos procesos se realizan sobre un sistema desarrollado en una plataforma HP3000 con el software speedware, el cual recopila datos personales del usuario asignándole password y login generando a su vez la credencial de acceso.

El sistema presenta las siguientes desventajas:

- La captura resulta tediosa y frecuentemente presenta errores, lo que genera pérdida de tiempo.
- La estructura de los diferentes menús es difícil de manejar debido a su complejidad.
- La presentación del sistema no es atractiva para el usuario.
- La aplicación esta desaprovechando las ventajas que ofrece el equipo adquirido recientemente.

ASIGNACIÓN DE EQUIPOS Y PRÉSTAMO DE MANUALES.

Esta acción se realiza para administrar los recursos con que cuenta el Centro de Computo:

- Impresoras
- PC.
- Macintosh
- Power - Macintosh.
- Estaciones de trabajo.
- Terminales
- Manuales.
- Vacunas.

Esta actividad es manual, utilizando una forma que contiene los siguientes datos:

- Fecha.
- Número de máquina.
- Tipo de máquina.
- Hora.
- Clave de usuario.

Las desventajas de este proceso son las siguientes:

- Lentitud en los trámites.
- Errores en la asignación de equipos.
- Pérdida de material.

CAPITULO III

- Asignaciones de tiempo en el uso de equipo a corto plazo.
- Dificultad en el control de fallas de equipos.

ASIGNACIÓN DE SALA DE CURSO.

Con la finalidad de administrar las diferentes salas de cursos el jefe de la Sección de Análisis y Apoyo Técnico, realiza una bitácora manual en la cual plasma datos como:

- Nombre del profesor.
- Nombre del curso.
- Duración.
- Hora en que se impartirá
- Número de alumnos
- Sala del curso

Este tipo de control nos lleva a:

- Malas asignaciones de las salas.
- Duplicación de horarios
- Falta de material necesario
- Sobrecupo de las salas.

CONTROL DE REPARACIÓN DE EQUIPO.

Esta actividad actualmente se realiza mediante un sistema que tienen como objetivo registrar los equipos que requieran algún tipo de mantenimiento (preventivo ó correctivo).

Dicho sistema esta desarrollado en CLIPPER y requiere la siguiente información:

- Nombre del solicitante.
- Coordinación a la que pertenece.
- Departamento o sección.
- Ubicación física del equipo.
- Número de serie.
- Marca.
- Modelo.

- Falla.
- Accesorios.

Este sistema genera un comprobante de recibo y entrega de equipo.

DESVENTAJAS:

- Dificultad en la captura de datos.
- Dificultad para modificar la información.
- Reportes deficientes e incompletos.
- Falta de algún control estadístico.

CONTROL DE INVENTARIO

Anexo a lo anterior se realiza manualmente un control de inventario, obsoleto ya que con el tiempo se traspapela o pierde la información y esto pone en riesgo el material con el que cuenta el centro de cómputo.

Además esta actividad es muy tediosa y redundante y genera pérdidas de tiempo trimestralmente, al mismo tiempo no se lleva un control histórico y estadístico de la información que puede servir para tomas de decisiones posteriores en cuanto adquisiciones de equipo o contratación de personal. Esto crea la necesidad de automatizar en su totalidad dichos trámites.

III.5. PROPUESTA DE SOLUCIÓN

Para la automatización de dichos trámites se va a desarrollar un sistema con las siguientes herramientas:

- MAINFRAME HP9000 827s con sistema operativo UNIX versión 9.14
- ORACLE versión 7, manejador de base de datos.
- BAR CODE AND MAGNETIC STRIPE READER MODELS 2000/2002
- PC's 486DX emulando terminales de la HP9000
- RED ETHERNET con protocolo de comunicaciones TCP/IP

Se propone se realice con las siguientes condiciones:

- El sistema deberá tener como dispositivos de entrada de datos el lector de código de barras y teclado y como dispositivos de salida la pantalla así como la impresora.

CAPITULO III

- El sistema deberá llevar el control de reservaciones y reasignaciones de equipo y salas de cursos por día y hora.
- Deberá llevar el control de las ordenes de servicios de reparación y mantenimiento de equipo de cómputo.
- Permitirá el manejo de inventario de equipo y usuarios del centro de cómputo.
- Generará estadísticas acerca de demandas de servicios de cómputo en fechas, horarios y equipos ; para ayudar en la toma de decisiones en la administración del centro de cómputo.
- Permitirá obtener reportes de cada una de las actividades mencionadas en la parte de análisis de información.
- Se realizará un enlace entre las PC's y un HP9000 mediante las utilerías que presenta el software para red Pathway versión 3.
- Posteriormente se llevará a cabo una decodificación de los datos de entrada con el lector de código de barras, para realizar una interfaz que permita su entrada a una base de datos de tipo relacional. Esta base de datos será escrita en ORACLE y operará dentro de la plataforma HP9000 que a su vez esta conectada a la red Ethernet con protocolo de comunicación TCP/IP.



CAPITULO IV

DISEÑO E
IMPLEMENTACIÓN



CAPITULO IV. DISEÑO E IMPLEMENTACIÓN.

IV.1 NOTACIONES DE DISEÑO.

La Universidad Autónoma Metropolitana cuenta con una serie de notaciones predefinidas para todos sus planteles, dichas notaciones son detalladas a continuación:

DEFINICIÓN DE PRESENTACIÓN DE MENÚS.

- Justificada a la derecha aparecerá la fecha en la forma "DD/MM/AA".
- En forma centrada aparecerá el nombre del menú actual.
- Apareceran las posibles opciones a elegir, estas deben de escribirse a partir de la columna 11 a la columna 70 justificadas a la izquierda.

PANTALLAS DE AYUDA.

•El recuadro de ayuda, debera presentarse en cualquier momento que se desee dentro de la forma.

•Para cada campo existente en la forma, debera existir un campo de ayuda.

ENTIDADES.

Para denominar una entidad se utilizará una letra que identifique el Servidor de la Unidad Académica donde reside físicamente la entidad, seguida de hasta trece letras que identifiquen claramente la entidad, por ejemplo: XBECAS (Entidad de Becas que reside en el servidor de Xochimilco). La primer letra de la identificación de una entidad debe ser distinta de "O" y "U".

UNIDADES

Azcapotzalco	A
Iztapalapa	I
Xochimilco	X
Rectoría	R

CAMPOS

Deben ser autodescriptivos con una longitud máxima de once caracteres, seguido del tipo de campo de acuerdo a la tabla que a continuación se especifica:

Alfabético	_AL
Númérico	_NU
Cadena	_XX
Fecha	_FE
Lógico	_LO

CAPITULO IV

Descripción	_DE
Porcentaje	_PO
Clave	_CL
Nombre	_NO
Acumulador	_AC
Bandera	_BA
Hora	_HO
Importe	_IM
Indice	_IN
Imagen	_IG

Por ejemplo: EMPLEADO_CL (especifica la clave del empleado)

CLASES

Para definir una clase se utilizará hasta un máximo de catorce letras que la identifiquen claramente, por ejemplo: ALPROY_CLAS (define el nombre que será heredado a los campos que necesiten de ella).

ETIQUETAS

Para identificar un etiqueta se utilizará una número de tres dígitos (el cual deberá inicialmente incrementarse de 10 en 10, para posteriormente poder insertar alguna etiqueta intermedia), seguido de "_" y una descripción de longitud máxima de diez caracteres y debe terminar con ":", por ejemplo 010_INI_CONTEO: (Etiqueta que indica el inicio de un ciclo de conteo).

LIBRERIAS

Para identificar una librería se utilizarán las letras que identifican la aplicación, seguidas de dos dígitos que indican la versión de la biblioteca y terminando con el sufijo "_LIB", por ejemplo: AA01_LIB (Versión uno de la Biblioteca del Subsistema de Administración Académica).

ARCHIVOS DE DATOS

Existirán tres niveles de archivos de datos , los cuales se describen a continuación:

A NIVEL SIUAM.

Los archivos de datos que sean utilizados por mas de un subsistema del SIUAM se denominarán iniciando con el prefijo "S_" seguido del nombre del archivo, de una longitud de

ocho caracteres, terminando con el sufijo ".dat". Por ejemplo: S_plazas.dat (Archivo de plazas de la UAM contenidas en el SIUAM).

A NIVEL SUBSISTEMA

Los archivos de datos que sean utilizados por mas de un módulo del subsistema se denominarán iniciando con las dos letras que identifican el subsistema seguidos de "_" y el nombre del archivo, de una longitud de siete caracteres, terminando con el sufijo ".dat". Por ejemplo: AA_profes.dat (Archivo de profesores de la UAM contenidas en el subsistema de Administración Académica).

A NIVEL MÓDULO

Los archivos de datos que sean utilizados exclusivamente por un módulo se denominarán iniciando con las dos letras que identifica el módulo, seguido del nombre del archivo, de una longitud de siete caracteres, terminando con el sufijo ".dat". Por ejemplo: BE_profes.dat (Archivo de profesores que tienen becas del módulo de becas).

IV.2. DIAGRAMAS DE FLUJO DE DATOS

Son diagramas que se utilizan para escribir un diseño de sistemas de alto nivel; muestran como se transforman los datos al pasar de un componente del sistema a otro.

Los diagramas de flujo de datos constan de tres componentes:

- 1.- Flechas con anotaciones.
- 2.- Burbujas con anotaciones.
- 3.- Los operadores * y ⊕

Las burbujas con anotaciones representan centros de transformación en los que la anotación específica la transformación. Las flechas representan el flujo de datos hacia adentro y afuera de los centros de transformación, donde las anotaciones dan nombre al flujo de datos. Los diagramas de flujo de datos describen cómo una entrada se transforma en una salida. No deben incluir información de control o sucesión de la información. Cada burbuja se puede considerar como una caja negra independiente que transforma sus entradas en salidas. Los operadores * y ⊕ se utilizan para unir flechas - * significa AND ("y" lógico) y ⊕ significa EXCLUSIVE OR ("o" exclusivo lógico).

CAPITULO IV

Una de las ventajas de los diagramas de flujo de datos es que muestran las transformaciones sin hacer ninguna suposición sobre su aplicación.

La preparación de los diagramas de flujo de datos se enfoca mejor si se tienen en cuenta las entradas al sistema y se trabaja hacia las salidas. Cada burbuja debe representar una transformación distinta: su salida debe, de alguna manera, ser diferente de su entrada. No hay reglas para determinar la estructura total del diagrama, y construir un diagrama de flujo de datos es uno de los aspectos creativos del diseño de sistemas.

A continuación se muestran los diagramas de flujo que se van a manejar, con la finalidad de ilustrar las funciones principales del sistema que se describen como:

1) VALIDACIÓN EN BASE DE DATOS. Este proceso verifica que el interesado, este inscrito como alumno del trimestre que transcurre.

2)VALIDACIÓN EN BASE DE DATOS DEL CENTRO DE CÓMPUTO. Este proceso verifica que el alumno este inscrito al C.C. y que no adeude material.

3)DAR DE ALTA. . Una vez que se realicen las validaciones anteriores, en este proceso se le da su nueva clave de acceso con la que podrá laborar a lo largo del trimestre.

Esta clave podrá ser de tres tipos:

4) DAR DE ALTA ESTATUS NORMAL. Esta indica que el alumno podrá laborar en el C.C en horarios restringidos.

5)DAR DE ALTA ESTATUS PROYECTO. Indica que el alumno podrá laborar en el C.C. únicamente si se encuentra inscrito en algún proyecto y tendrá un horario especial asignado previamente por el Líder del Proyecto.

6)DAR DE ALTA ESTATUS AMBOS. Indica que el alumno laborará en el centro de cómputo con ambos horarios.

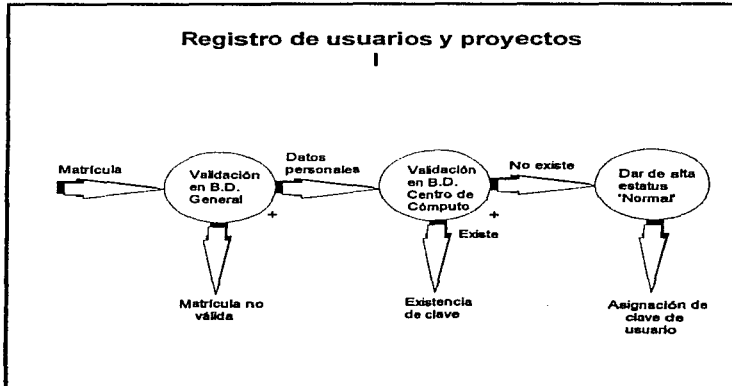
7) CONSULTA DE EQUIPOS, SALAS Y MATERIAL SIN ASIGNAR. Verifica salas, equipos y/o material que están disponibles para asignación ó prestamo.

8)REVISIÓN DEL PERÍODO DE PRÉSTAMO. Verifica que el alumno no adeude material previo a su último requerimiento.

9)GENERACIÓN DE ORDEN DE PAGO. Si el alumno adeuda material, entonces se generará una orden de pago que deberá liquidar antes de hacer otra solicitud.

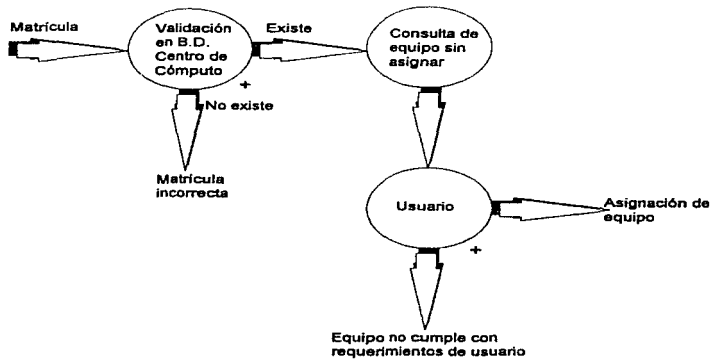
REPARACIÓN DE EQUIPO

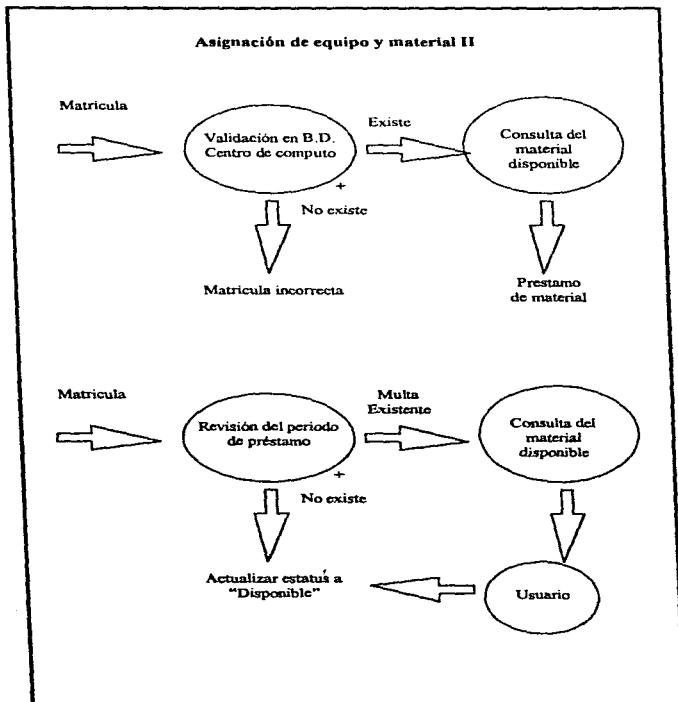
10) **CAMBIAR ESTATUS DE REPARACIÓN.** Verifica que el equipo que será enviado a reparación, exista en la base de datos, de ser así cambia su estatus a **reparación**, si no, se da de alta con un estatus de **reparación exterior**; posteriormente se genera un comprobante que se intercambiará cuando el equipo este reparado.



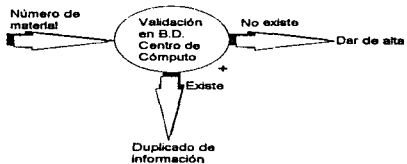


Asignación de equipo y material

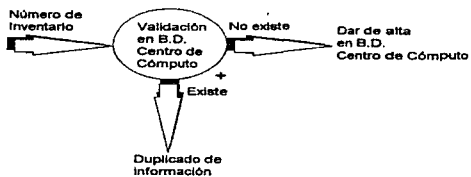


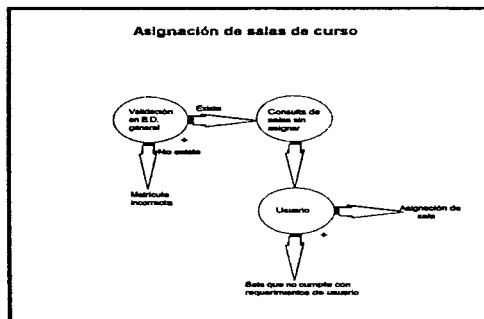
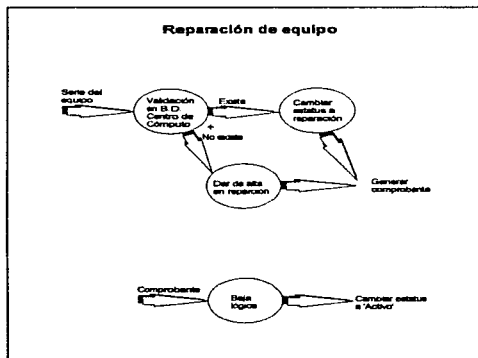


Asignación de equipo y material III



Inventario





IV.3. MODELOS ENTIDAD-RELACION

En las páginas siguientes se muestran los diagramas del modelo Entidad-Relación para el sistema de control del Centro de Cómputo. Cada diagrama representa una o más transacciones del Centro de Cómputo. El último diagrama corresponde al esquema general del sistema con todas las entidades involucradas.

Cada rectángulo representa una entidad y dentro de este se encuentran los atributos que corresponden a la entidad en turno. Aquí se indican también las llaves primarias (#) y las llaves foráneas (fk).

Por razones de espacio, en el esquema general sólo se muestran las llaves primarias.

Las relaciones entre entidades están denotadas como sigue:

Uno a muchos: \longrightarrow

Muchos a uno: \longleftarrow

ASIGNACION DE EQUIPO

INSCRIPCION
* Matricula
fk No. Material
Password
Status_inscripcion



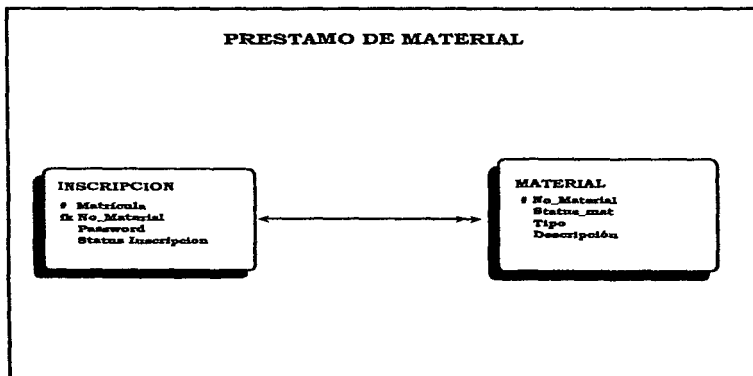
CONEQUIPO
* No_maquina
* Hora
* Fecha
Matricula
Sala

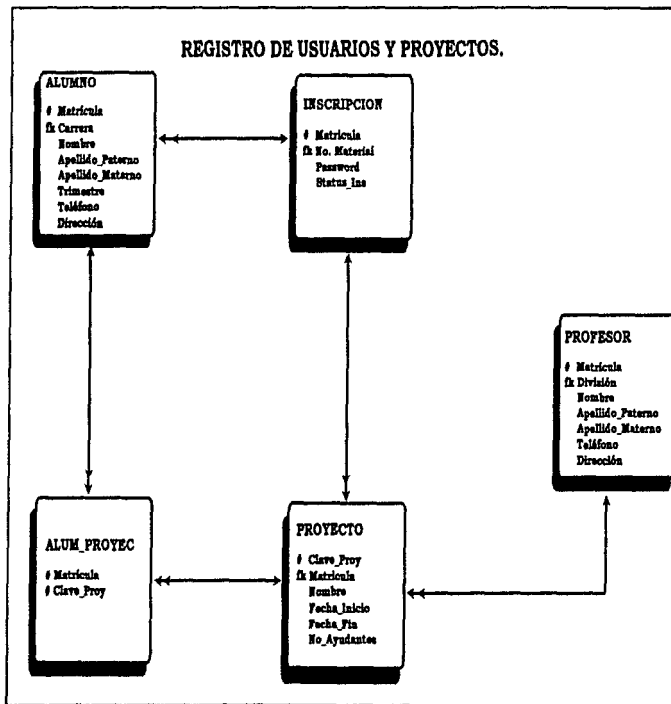


EQUIPO
* No_maquina
fk No_inventario
Prestamo

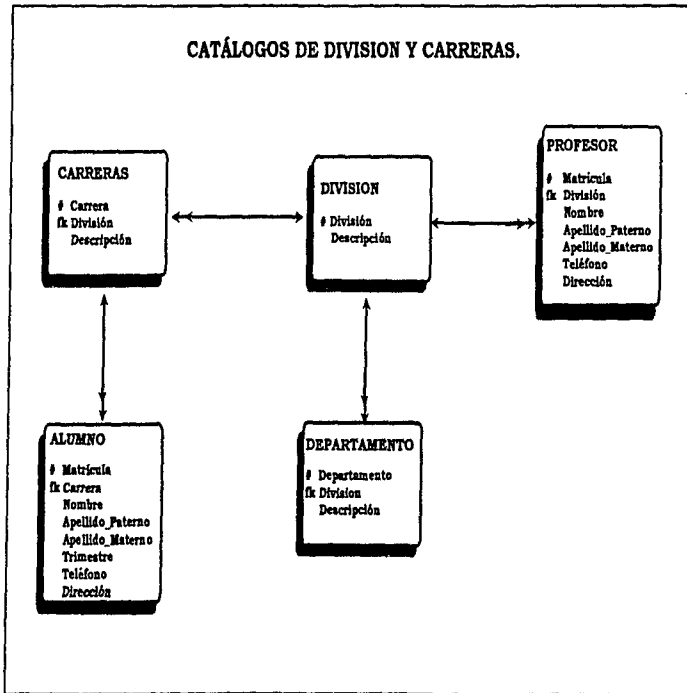


INVENTARIO
* No. de Inventario
fk No. de Serie
fk Marca
Descripción
Fecha

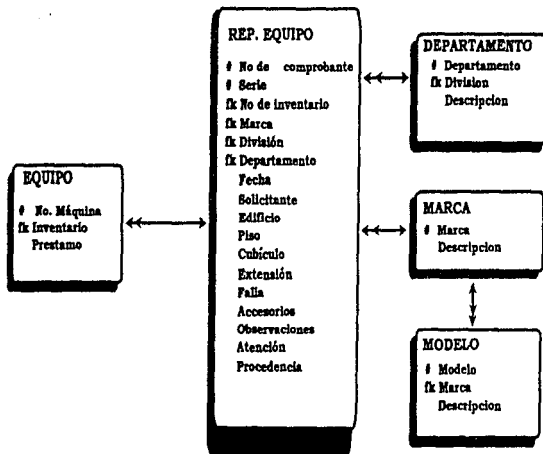




CATÁLOGOS DE DIVISION Y CARRERAS.



REPARACIÓN DE EQUIPO



PRESTAMO DE SALAS

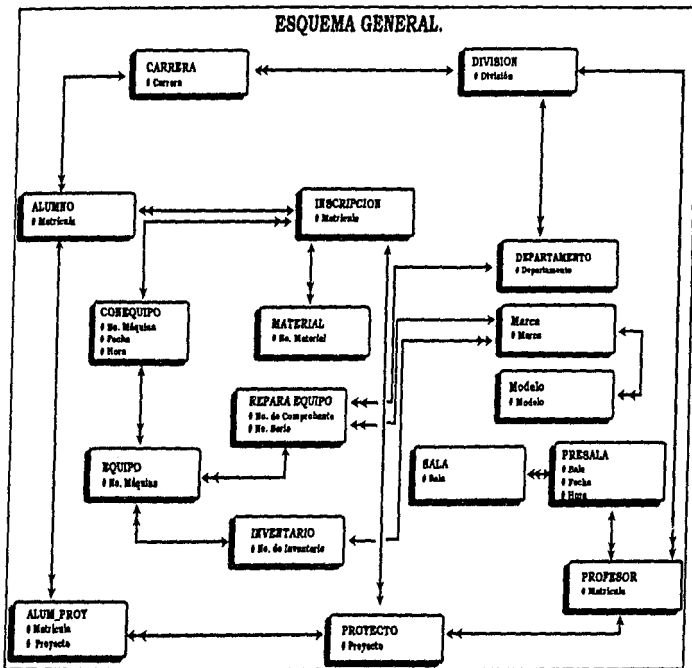
SALA
Clave_de_sala
Descripción
No. de máquinas



PRESTALO
Clave_de_sala
Fecha
Hora
fk Matricula
Curso



PROFESOR
Matricula
fk Division
Nombre
Apellido_paterno
Apellido_materno
Telefono
Direccion



IV.4. DICCIONARIO DE DATOS.

A continuación se muestran las tablas del sistema indicando el número, nombre, tipo y longitud del campo.

ESTRUCTURA DE TABLA : XALUMNO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MATR CL	NUMÉRICO	8
2	NOMB NO	CARACTER	20
3	APPA XX	CARACTER	20
4	APMA XX	CARACTER	20
5	DIRE XX	CARACTER	40
6	TELE XX	CARACTER	15
7	CARE CL	NUMÉRICO	2
8	TRIM NU	NUMÉRICO	2

ESTRUCTURA DE TABLA: XCARRERAS

CAMPO	NOMBRE	TIPO	LONGITUD
1	#CARE CL	NUMÉRICO	2
2	DESC XX	CARACTER	50
3	DIVI CL	NUMÉRICO	2

ESTRUCTURA DE TABLA: XPROFESOR

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MATR CL	NUMÉRICO	8
2	NOMB NO	CARACTER	20
3	APPA XX	CARACTER	20
4	APMA XX	CARACTER	20
5	DIRE XX	CARACTER	40
6	TELE XX	CARACTER	15
7	DIVI CL	NUMÉRICO	2

ESTRUCTURA DE TABLA: XDIVISION

CAMPO	NOMBRE	TIPO	LONGITUD
1	#DIVI CL	NUMÉRICO	2
2	DESC XX	CARACTER	50

CAPITULO IV**ESTRUCTURA DE TABLA: XINSCRIPCION**

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MATR CL	NUMÉRICO	8
2	PAWR XX	CARACTER	10
3	NOMATE CL	NUMÉRICO	3
4	STINS NU	NUMÉRICO	1

ESTRUCTURA DE TABLA: XPROYECTO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#PROY CL	NUMÉRICO	8
2	MATR CL	NUMÉRICO	8
3	NOMPROY NO	CARACTER	30
4	INFECH FE	FECHA	10
5	FINFECH FE	FECHA	10
6	AYUNO NU	NUMÉRICO	2

ESTRUCTURA DE TABLA: XEQUIPO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#NOMAO CL	NUMÉRICO	2
2	INVE CL	CARACTER	15
3	PRST NU	NUMÉRICO	1

ESTRUCTURA DE TABLA: XINSALUMPROY

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MATR CL	NUMÉRICO	8
2	#PROY CL	NUMÉRICO	8

ESTRUCTURA DE TABLA: XPRESALA

CAMPO	NOMBRE	TIPO	LONGITUD
1	#SALA CL	CARACTER	10
2	#FECH CL	FECHA	10
3	#HORA CL	FECHA	10
4	MATR CL	NUMÉRICO	8
5	CURS XX	CARACTER	30

ESTRUCTURA DE TABLA: X SALA

CAMPO	NOMBRE	TIPO	LONGITUD
1	#SALA CL	CARACTER	10
2	MAQS NU	NUMÉRICO	2
3	DESC XX	CARACTER	50

ESTRUCTURA DE TABLA: XCONEQUIPO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#NOMAO CL	NUMÉRICO	2
2	#FECH CL	FECHA	10
3	#HORA CL	FECHA	10
4	MATR CL	NUMÉRICO	8
5	SALA XX	CARACTER	10

ESTRUCTURA DE TABLA: XINVENTARIO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#INVE CL	CARACTER	15
2	DESC XX	CARACTER	50
3	FECH FE	FECHA	10
4	NOSERI CL	CARACTER	15
5	MARC CL	NUMÉRICO	2

ESTRUCTURA DE TABLA: XREPAREQUIPO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#COMP CL	NUMÉRICO	8
2	#NOSERI CL	CARACTER	15
3	FECH FE	FECHA	10
4	SOLINOM XX	CARACTER	35
5	DIVI CL	NUMÉRICO	2
6	DEPT CL	NUMÉRICO	2
7	EDIF XX	CARACTER	1
8	PISO NU	NUMÉRICO	1
9	CUBI NU	NUMÉRICO	2
10	FALLA XX	CARACTER	50
11	ACCESO XX	CARACTER	50
12	OBSERV XX	CARACTER	50
13	ATNCION XX	CARACTER	45
14	MARC CL	NUMÉRICO	2

CAPITULO IV

15	PROCE XX	CARACTER	7
16	INVE CL	CARACTER	15
17	EXT NU	NUMÉRICO	4

ESTRUCTURA DE TABLA: XMATERIAL

CAMPO	NOMBRE	TIPO	LONGITUD
1	#NOMATE CL	NUMÉRICO	3
2	PRESMO NU	NUMÉRICO	1
3	TIPOMA XX	CARACTER	15
4	DESC XX	CARACTER	50
5	MATR CL	NUMÉRICO	8

ESTRUCTURA DE TABLA: XDEPTO

CAMPO	NOMBRE	TIPO	LONGITUD
1	#DEPT CL	NUMÉRICO	2
2	DIVI CL	NUMÉRICO	2
3	DESC XX	CARACTER	50

ESTRUCTURA DE TABLA: XMARCA

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MARCA CL	NUMÉRICO	2
2	DESC XX	CARACTER	50

ESTRUCTURA DE TABLA: XMODEL

CAMPO	NOMBRE	TIPO	LONGITUD
1	#MODEL CL	NUMÉRICO	2
2	DESC XX	CARACTER	50
3	MARCA CL	NUMÉRICO	2

A continuación se muestra el detalle de cada una de las tablas, sus relaciones y los campos que contienen.

Nombre	XALUMNO																														
Alias	ALUMNO																														
Llave primaria	MATR_CL																														
Descripción	Tabla para almacenar los datos referentes a los alumnos inscritos en la U.A.M. Xochimilco.																														
Campos que involucra	<table border="0"> <tr> <td>Campos</td> <td></td> <td>Tabla</td> </tr> <tr> <td>MATR_CL</td> <td>←</td> <td>XINSCRIPCIÓN</td> </tr> <tr> <td></td> <td>→</td> <td>XINSALUMPROY</td> </tr> <tr> <td>NOMB_NO</td> <td></td> <td></td> </tr> <tr> <td>APPA_XX</td> <td></td> <td></td> </tr> <tr> <td>APMA_XX</td> <td></td> <td></td> </tr> <tr> <td>DIRE_XX</td> <td></td> <td></td> </tr> <tr> <td>TELE_XX</td> <td></td> <td></td> </tr> <tr> <td>CARE_CL</td> <td>←</td> <td>XCARRERAS</td> </tr> <tr> <td>TRIM_NU</td> <td></td> <td></td> </tr> </table>	Campos		Tabla	MATR_CL	←	XINSCRIPCIÓN		→	XINSALUMPROY	NOMB_NO			APPA_XX			APMA_XX			DIRE_XX			TELE_XX			CARE_CL	←	XCARRERAS	TRIM_NU		
Campos		Tabla																													
MATR_CL	←	XINSCRIPCIÓN																													
	→	XINSALUMPROY																													
NOMB_NO																															
APPA_XX																															
APMA_XX																															
DIRE_XX																															
TELE_XX																															
CARE_CL	←	XCARRERAS																													
TRIM_NU																															

Nombre	XPROYECTO																								
Alias	PROYECTO																								
Llave primaria	PROY_CL																								
Descripción	Tabla para almacenar los datos referentes a los proyectos que elaboran los investigadores en las diferentes divisiones académicas de la U.A.M. Xochimilco.																								
Campos que involucra :	<table border="0"> <tr> <td>Campo</td> <td></td> <td>Tabla</td> </tr> <tr> <td>PROY_CL</td> <td>→</td> <td>XINSALUMPROY</td> </tr> <tr> <td>MATR_CL</td> <td>←</td> <td>XPROFESOR</td> </tr> <tr> <td></td> <td>←</td> <td>XINSCRIPCIÓN</td> </tr> <tr> <td>NOMPROY_NO</td> <td></td> <td></td> </tr> <tr> <td>INIFECH_FE</td> <td></td> <td></td> </tr> <tr> <td>FINFECH_FE</td> <td></td> <td></td> </tr> <tr> <td>AYUNO_NU</td> <td></td> <td></td> </tr> </table>	Campo		Tabla	PROY_CL	→	XINSALUMPROY	MATR_CL	←	XPROFESOR		←	XINSCRIPCIÓN	NOMPROY_NO			INIFECH_FE			FINFECH_FE			AYUNO_NU		
Campo		Tabla																							
PROY_CL	→	XINSALUMPROY																							
MATR_CL	←	XPROFESOR																							
	←	XINSCRIPCIÓN																							
NOMPROY_NO																									
INIFECH_FE																									
FINFECH_FE																									
AYUNO_NU																									

CAPITULO IV

Nombre	XCARRERAS	
Alias	CARRERA	
Llave primaria	CARE_CL	
Descripción	Tabla para almacenar el catálogo de las carreras que se imparten en la U.A.M. Xochimilco.	
Campos que involucra	Campos	Tablas
	CARE_CL →→	XALUMNO
	DIVI_CL ←←	XDIVI
	DESC_XX	

Nombre	XDIVISIÓN	
Alias	DIVISION	
Llave primaria	DIVI_CL	
Descripción	Tabla para almacenar el catálogo de las diferentes divisiones que existen en la U.A.M. Xochimilco.	
Campos que involucra	Campo	Tabla
	DIVI_CL →→	XCARRERAS
	→→	XPROFESOR
	→→	XDEPTO
	DESC_XX	

Nombre	XINSCRIPCIÓN	
Alias	INSCRIPCIÓN	
Llave primaria	MATR_CL	
Descripción	Tabla para almacenar los datos referentes a los usuarios del centro de cómputo.	
Campos que involucra :	Campo	Tabla
	MATR_CL →→	XALUMNO
	→→	XPROYECTO
	←←	XCONQUIPO
	PAWR_XX	
	NOMATE_CL →→	XMATERIAL
	STINS NU	

Nombre	XEQUIPO		
Alias	CONTROL DE EQUIPOS ASIGNADOS		
Llave primaria	NOMAQ_CL		
Descripción	Tabla para almacenar los datos referentes a las máquinas que están asignadas a algún usuario del centro de cómputo.		
Campos que involucra :	Campo		Tabla
	NOMAQ_CL	↔←	XCON EQUIPO
	INVE_CL	↔←	XINVENTARIO
		↔←	XREPAREQUIPO

Nombre	XINSALUMPROY		
Alias	INSCRIPCIÓN DE ALUMNO A PROYECTO		
Llave primaria	PROY_CL + MATR_CL		
Descripción	Tabla para almacenar los datos referentes a los alumnos que trabajan como ayudantes de un investigador que realiza un proyecto.		
Campos que involucra :	Campo		Tabla
	PROY_CL	↔←	XPROYECTO
	MATR_CL	↔←	XALUMNO

Nombre	XPRESALA		
Alias	CONTROL DE PRÉSTAMO DE SALAS DE CURSOS		
Llave primaria	SALA_CL + HORA_CL + FECHA_CL		
Descripción	Tabla para almacenar los datos referentes a las salas de cursos que están asignadas a algún profesor del plantel.		
Campos que involucra :	Campo		Tabla
	SALA_CL+	→→	XSALA
	FECH_CL+		
	HORA_CL		
	MATR_CL	→→	XPROFESOR
	MAQS_NU CURS_XX		

CAPITULO IV

Nombre	XMATERIAL	
Alias	Control del préstamo de material	
Llave primaria	NOMATE_CL	
Descripción	Contiene los datos del material (Inventario Material)	
Campos que involucra	Campo	Tabla
	NOMATE_CL ←	XINSCRIPCION
	STINS_NU	
	TIPOMA_XX	
	DESC_XX	

Nombre	XCONEQUIPO	
Alias	CONTROL DE PRÉSTAMO DE EQUIPO	
Llave primaria	NOMAQ_CL + FECH_CL + HORA_CL	
Descripción	Contiene la matrícula de la persona que solicita el equipo, así como la fecha y hora de ésta y la sala en donde se trabajará	
Campos que involucra	Campo	Tabla
	NOMAQ_CL →	XEQUIPO
	FECH_CL	
	HORA_CL	
	MATR_CL →	XINSCRIPCION
	SALA_XX	

Nombre	XSALA	
Alias	CATALOGO DE SALAS DE CURSO	
Llave primaria	SALA_CL	
Descripción	Contiene la información de cada una de las salas de curso.	
Campos que involucra	Campo	Tabla
	SALA_CL ←	XPRESALA
	MAQS_NU	
	DESC_XX	

Nombre	XREPAREQUIPO	
Alias	Contiene los datos del equipo que se repara.	
Llave primaria	COMP_CL + NOSERI_CL	
Descripción	Contiene número de comprobante y el número de serie del equipo a reparar, así como otros datos del equipo.	
Campos que involucra	Campo	Tabla
	COMP_CL NOSERI_CL FECH_FE SOLINOM_XX DIVI_CL DEPT_CL ← XDEPTO EDIF_XX PISO_NU CUBI_NU FALLA_XX ACCESO_XX OBSERV_XX ATNCION_XX MARC_CL ← XMARCA PROCE_XX INVE_CL ← XEQUIPO EXT_NU	

Nombre	XPROFESOR	
Alias	PROFESOR	
Llave primaria	MATR_CL	
Descripción	Contiene los datos personales del profesor.	
Campos que involucra	Campo	Tabla
	MATR_CL → XPROYECTO ← XPRESALA NOMB_NO APPA_XX APMA_XX DIRE_XX TELE_XX DIVI_CL ← XDIVISIÓN	

CAPITULO IV

Nombre	XINVENTARIO	
Alias	INVENTARIO	
Llave primaria	INVE_CL	
Descripción	Contiene los datos técnicos y descriptivos de todos los equipos que pertenecen al Centro de Cómputo.	
Campos que involucra	Campo	Tabla
	INVE_CL	—————→ XEQUIPO
	DESC_XX	
	FECH_FE	
	NOSERL_CL	
MARC_CL	—————→ XMARCA	

Nombre	XDEPTO	
Alias	DEPARTAMENTO	
Llave primaria	DEPT_CL	
Descripción	Catálogo de los departamentos de cada división del plantel.	
Campos que involucra	Campo	Tabla
	DEPT_CL	—————→ XREPAREQUIPO
	DIVI_CL	←———— XDIVISION
	DESC_XX	

Nombre	XMARCA	
Alias	MARCA	
Llave primaria	DEPT_CL	
Descripción	Catálogo de los todas las marcas de computadoras que se manejan en el Plantel.	
Campos que involucra	Campo	Tabla
	MARCA_CL	—————→ XREPAREQUIPO
		←———— XINVENTARIO
		—————→ XMODEL
	DESC_XX	

Nombre	XMODEL	
Alias	MODELO	
Llave primaria	MODEL_CL	
Descripción	Catálogo de los modelos de las marcas de computadoras que se manejan en el Plantel.	
Campos que involucra	Campo	Tabla
	MODEL_CL	
	MARCA_CL	← XMARCA
	DESC_XX	

Ahora se describirá el detalle de los campos utilizados en el sistema, las tablas en las que son utilizados, su longitud y tipo.

Nombre	DESC_XX
Alias	DESCRIPCIÓN
Descripción	Campo para almacenar las descripciones de las claves de algunas entidades utilizadas en el diseño
Formato	Dato de tipo alfanumérico Longitud de 50 caracteres
Observaciones	Este campo se encuentra en las tablas: XCARRERAS XDIVISIÓN XMARCA XMODEL XSALA XINVENTARIO XDEPTO XMATERIAL

CAPITULO IV

Nombre	MATR_CL
Alias	MATRICULA
Descripción	Campo para almacenar la matricula de alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo numérico Longitud de 8 enteros
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR XPROYECTO XINSCRIPCIÓN XINSALUMPROY XCONEQUIPO XPRESALA XMATERIAL

Nombre	NOMB_NO
Alias	NOMBRE
Descripción	Campo que se utiliza para almacenar el nombre de alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo alfanumérico Longitud de 20 caracteres
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR

Nombre	APPA XX
Alias	APELLIDO PATERNO
Descripción	Campo para almacenar el apellido paterno de los alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo alfanumérico Longitud de 20 caracteres
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR

Nombre	APMA XX
Alias	APELLIDO MATERNO
Descripción	Campo para almacenar el apellido materno de los alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo alfanumérico Longitud de 20 caracteres
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR

Nombre	DIRE XX
Alias	DIRECCIÓN
Descripción	Campo para almacenar la dirección de los alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo alfanumérico Longitud de 40 caracteres
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR

CAPITULO IV

Nombre	TELE XX
Alias	TELÉFONO
Descripción	Campo para almacenar el teléfono de los alumnos y profesores de la U.A.M. Xochimilco.
Formato	Dato de tipo alfanumérico Longitud de 15 caracteres
Observaciones	Este campo se encuentra en las tablas: XALUMNO XPROFESOR

Nombre	CARE CL
Alias	CARRERA
Descripción	Campo para almacenar el número de la carrera de los alumnos de la U.A.M. Xochimilco.
Formato	Dato de tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en las tablas XALUMNO XCARRERAS

Nombre	TRIM NU
Alias	TRIMESTRE
Descripción	Campo para almacenar el trimestre en el que se encuentran los alumnos de la UAM-X
Formato	Dato de tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en la tabla: XALUMNO

Nombre	DIVICL
Alias	DIVISION
Descripción	Campo que se utiliza para almacenar el número de la División a la que pertenecen los usuarios del Centro de Cómputo
Formato	Dato de tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en las tablas: XCARRERAS XDIVISION XREPAREQUIPO XPROFESOR XDEPTO

Nombre	NOMPROY NO
Alias	NOMBRE DE PROYECTO
Descripción	Campo que se utiliza para almacenar el nombre de los proyectos a realizarse en la UAM-X con la ayuda de equipo existente en el Centro de Cómputo
Formato	Dato de tipo alfanumérico Longitud de 30 caracteres
Observaciones	Este campo se encuentra en la tabla: XPROYECTO

Nombre	INFECH FE
Alias	FECHA DE INICIO
Descripción	Campo que se utiliza para almacenar la fecha de inicio de un proyecto
Formato	Dato tipo fecha Longitud de 10 enteros
Observaciones	Este campo se encuentra en la tabla: XPROYECTO

CAPITULO IV

Nombre	FINFECH FE
Alias	FECHA DE FINALIZACIÓN
Descripción	Campo que se utiliza para almacenar la fecha de fin de un proyecto
Formato	Dato tipo fecha Longitud de 10 enteros
Observaciones	Este campo se encuentra en la tabla: XPROYECTO

Nombre	AYUNO NU
Alias	NÚMERO DE AYUDANTES
Descripción	Campo que se utiliza para almacenar el número de ayudantes de un profesor asignado a un proyecto
Formato	Dato tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en la tabla: XPROYECTO

Nombre	PROY CL
Alias	NÚMERO DE PROYECTO
Descripción	Identifica un proyecto de investigación desarrollado por académicos y alumnos de U.A.M. Xochimilco.
Formato	Campo de tipo numérico Longitud 8 enteros.
Observaciones	Este campo se encuentra en las tablas : XPROYECTO XINSALUMPRO

Nombre	PAWR_XX
Alias	PASSWORD DE ACCESO
Descripción	Clave de acceso utilizado para acceder el equipo del centro de cómputo.
Formato	Campo de tipo alfanumérico Longitud 10 caracteres
Observaciones	Este campo se encuentra en la tabla : XINSCRIPCION

Nombre	NÓMATE_CL
Alias	NÚMERO DE MATERIAL
Descripción	Este campo es un identificador numérico que nos indica que material es el que adeuda un usuario del centro de computo.
Formato	Campo de tipo numérico Longitud 3 enteros
Observaciones	Este campo se encuentra en las tablas : XINSCRIPCION XMATERIAL

Nombre	INVE_CL
Alias	NÚMERO DE INVENTARIO
Descripción	Este campo es un identificador que nos indica el número de inventario que tiene asignado un equipo.
Formato	Campo de tipo alfanumérico Longitud 15 caracteres
Observaciones	Este campo se encuentra en las tablas : XINVENTARIO XEQUIPO XREPAREQUIPO

CAPITULO IV

Nombre	NOSERI CL
Alias	NUMERO DE SERIE
Descripción	Este campo es un identificador numérico que nos indica el número de serie que asigna a un equipo el fabricante.
Formato	Campo de tipo alfanumérico Longitud 15 caracteres
Observaciones	Este campo se encuentra en las tablas : XINVENTARIO XREPAREQUIPO

Nombre	MARC CL
Alias	MARCA
Descripción	Este campo es un identificador que nos indica la clave de la marca del equipo.
Formato	Campo de tipo numérico Longitud 2 enteros
Observaciones	Este campo se encuentra en las tablas : XINVENTARIO XREPAREQUIPO XMARCA XMODEL

Nombre	MODEL CL
Alias	MODELO
Descripción	Este campo es un identificador que nos indica la clave del modelo del equipo.
Formato	Campo de tipo numérico Longitud 2 enteros
Observaciones	Este campo se encuentra en las tablas : XMODELO

Nombre	MAQS NU
Alias	NÚMERO DE MÁQUINAS
Descripción	Este campo es un identificador que nos indica el número de máquinas que tiene cada sala de cursos.
Formato	Campo de tipo numérico Longitud 2 enteros
Observaciones	Este campo se encuentra en la tabla : XSALA

Nombre	CURS XX
Alias	CURSO
Descripción	Este campo nos indica el tema del curso que se imparte en una sala.
Formato	Campo de tipo alfanumérico Longitud 30 caracteres
Observaciones	Este campo se encuentra en la tabla : XPRESALA

Nombre	NOMAQ CL
Alias	NÚMERO DE MÁQUINA
Descripción	Número que se asigna a cada máquina para llevar el control en el préstamo interno del Centro de Cómputo.
Formato	Dato de tipo numérico Longitud de 2 enteros..
Observaciones	Este campo se encuentra en las tablas: XCONEQUIPO XEQUIPO

CAPITULO IV

Nombre	FECH_CL
Alias	FECHA
Descripción	Campo para almacenar la fecha en que el equipo de cómputo es prestado.
Formato	Dato de tipo fecha Longitud de 10 enteros. DD-MM-AA (Día-Mes-Año)
Observaciones	Este campo se encuentra en las tablas: XCONEQUIPO XPRESALA

Nombre	FECH_FE
Alias	FECHA
Descripción	Campo de uso común para almacenar una fecha.
Formato	Dato de tipo fecha Longitud de 10 enteros. DD-MM-AA (Día-Mes-Año)
Observaciones	Este campo se encuentra en las tablas: XINVENTARIO XREPAREQUIPO

Nombre	HORA_CL
Alias	HORA
Descripción	Campo para almacenar la hora en que una sala o un equipo está reservada(o).
Formato	Dato de tipo alfanumérico Longitud de 5 caracteres 00:00 (horas: minutos)
Observaciones	Este campo se encuentra en las tablas: XCONEQUIPO XPRESALA

Nombre	SALA XX
Alias	SALA
Descripción	Campo para almacenar la sala en que se encuentra el equipo de cómputo en donde trabajará el solicitante
Formato	Dato de tipo alfanumérico Longitud de 10 caracteres
Observaciones	Este campo se encuentra en la tabla: XCONEQUIPO

Nombre	SALA CL
Alias	SALA
Descripción	Campo para almacenar la clave de la sala de cursos.
Formato	Dato de tipo alfanumérico Longitud de 10 caracteres
Observaciones	Este campo se encuentra en las tablas: XPRESALA XSALA

Nombre	TIPOMA XX
Alias	TIPO DE MATERIAL
Descripción	Campo para almacenar el tipo de material para préstamo
Formato	Dato de tipo alfanumérico Longitud de 15 enteros.
Observaciones	Este campo se encuentra en las tablas: XMATERIAL Podrá tomar dos valores: "DISCOS" y "MANUAL".

CAPITULO IV

Nombre	PRESMO NU
Alias	ESTATUS DE PRÉSTAMO
Descripción	Este campo es un identificador que nos indica si el material está prestado a algún alumno o no.
Formato	Campo de tipo numérico Longitud 1 entero
Observaciones	Este campo se encuentra en la tabla : XMATERIAL Podrá tomar dos valores: 0.- "PRESTADO" 1.- "DEVUELTO".

Nombre	COMP CL
Alias	NÚMERO DE COMPROBANTE
Descripción	Campo para almacenar el número de comprobante en una forma secuencial, con el que el solicitante de la reparación, podrá recoger su equipo una vez arreglado.
Formato	Dato de tipo numérico Longitud de 8 enteros.
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	SOLINOM XX
Alias	NOMBRE DEL SOLICITANTE
Descripción	Campo para almacenar el nombre de la persona que solicita la reparación de algún equipo de cómputo.
Formato	Dato de tipo alfanumérico Longitud de 35 caracteres
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	DEPT CL
Alias	DEPARTAMENTO DEL SOLICITANTE
Descripción	Campo para almacenar la clave del departamento al que pertenece la persona que solicite la reparación de algún equipo de cómputo.
Formato	Dato de tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en las tablas: XDEPTO XREPAREQUIPO

Nombre	EDIF XX
Alias	EDIFICIO DEL SOLICITANTE
Descripción	Campo para almacenar el edificio de procedencia de la persona que solicita la reparación de algún equipo de cómputo.
Formato	Dato de tipo alfanumérico Longitud de 1 carácter
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	PISO NU
Alias	PISO EN DONDE LABORA EL SOLICITANTE
Descripción	Campo para almacenar el piso en donde labora la persona que solicita la reparación de algún equipo de cómputo.
Formato	Dato de tipo numérico Longitud de 1 entero
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

CAPITULO IV

Nombre	CUBI NU
Alias	CUBICULO EN DONDE LABORA EL SOLICITANTE
Descripción	Campo para almacenar el cubiculo en donde labora la persona que solicita la reparación de algún equipo de cómputo.
Formato	Dato de tipo numérico Longitud de 2 enteros
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	FALLA XX
Alias	FALLA
Descripción	Campo para almacenar el tipo de falla que presenta el equipo de cómputo.
Formato	Dato de tipo alfanumérico Longitud de 50 caracteres
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	ACCESO XX
Alias	ACCESORIOS
Descripción	Campo para almacenar el tipo de accesorios que el equipo pudiese presentar al ingresar al departamento de mantenimiento.
Formato	Dato de tipo alfanumérico Longitud de 50 caracteres
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	OBSERV XX
Alias	OBSERVACIONES
Descripción	En este campo se almacena algún comentario que el encargado de la reparación tenga al respecto de ésta.
Formato	Dato de tipo alfanumérico Longitud de 50 caracteres
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	ATNCION XX
Alias	ATENCIÓN
Descripción	Se almacena el nombre de la persona que se encargará de atender la falla o daño del equipo.
Formato	Dato de tipo alfanumérico Longitud de 45 caracteres
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

Nombre	PROCE XX
Alias	PROCEDENCIA
Descripción	Aquí se contemplará el lugar de donde provenga el equipo a reparar, ya que se da mantenimiento a todo el plantel.
Formato	Dato de tipo alfanumérico Longitud de 7 caracteres.
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO Podrá tomar dos valores INTERNO ó EXTERNO

CAPITULO IV

Nombre	STINS NU
Alias	ESTATUS DE INSCRIPCIÓN
Descripción	En este campo se almacena el número de tipo de inscripción que el usuario tiene en el Centro de Cómputo
Formato	Dato de tipo numérico Longitud de 1 entero.
Observaciones	Este campo se encuentra en la tabla: XINSCRIPCION Podrá tomar tres valores: NORMAL, PROYECTO Y AMBOS por medio de un índice

Nombre	EXT NU
Alias	EXTENSION DEL SOLICITANTE
Descripción	En este campo se almacena el número de la extensión telefónica interna del solicitante de un servicio de reparación.
Formato	Dato de tipo numérico Longitud de 4 enteros.
Observaciones	Este campo se encuentra en la tabla: XREPAREQUIPO

IV.5. METODOLOGÍA DE LA PROGRAMACIÓN

Cuando comenzaron a surgir lenguajes de programación que permitían resolver problemas específicos mediante ordenador, al ser éste una máquina relativamente lenta y costosa, era muy importante diseñar los programas de manera que, aunque se tardara mucho tiempo en elaborarlos, el tiempo de ejecución en el ordenador no fuera excesivo.

Más tarde, con el avance tecnológico y la fabricación de ordenadores muy rápidos y con un precio accesible, resultó que lo que encarecía en gran medida la realización de un programa era el tiempo del programador.

Se planteaba, además, un grave problema cuando un programador diferente del autor de un programa tenía que modificarlo o corregirlo, debido a que uno lo realizaba según su criterio, sin modelo o estructura estándar alguna.

Todo esto condujo a crear una metodología de la programación que aportase las siguientes características:

- Adaptarse fácilmente a modificaciones.
- Propiciar una puesta a punto más rápida.
- Posibilitar una cómoda interpretación de los programas por diferentes programadores.

Así pues, la tendencia actual es tratar de conseguir, ante todo, programas simples y claros, que puedan ser mantenidos y actualizados fácilmente.

Para conseguir dichos propósitos se establecieron dos criterios generales de programación, conocidos con el nombre de Programación Modular y Programación Estructurada, que se desarrollarán en los apartados siguientes:

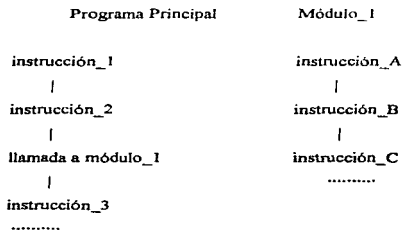
PROGRAMACIÓN MODULAR.

Esta técnica de programación consiste en dividir un programa en partes bien diferenciadas lógicamente, llamadas módulos, que pueden ser analizadas y programadas por separado.

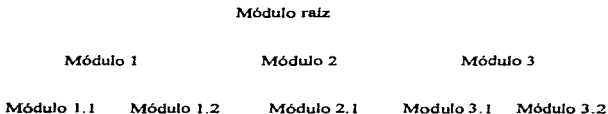
Un módulo se puede definir como un conjunto formado por una o varias instrucciones lógicamente enlazadas.

CAPITULO IV

A cada módulo se le asigna un nombre, que elige el programador, para poder identificarlo. Cuando en un punto de un programa se referencia un módulo, el programa le cede el control para que se ejecuten todas sus instrucciones. Finalizando el mismo, el control se devuelve al punto del programa desde donde se llamó, y se continúa con la ejecución de la instrucción siguiente a la que realizó la llamada. El orden de ejecución está representado gráficamente como sigue:



Si un módulo es lo suficientemente grande o complicado puede subdividirse en otros módulos, y éstos, a su vez, en otros, se puede representar un programa con un diagrama similar al siguiente:



Como se observa siempre existe un módulo raíz o principal, que es el encargado de controlar y relacionar a todos los demás.

Si los módulos que componen un programa son parte integrante del mismo, se les llamará rutinas, subrutinas, procedimientos o funciones. Si por lo contrario, son programas independientes, reclamados desde otros programas, se les denomina subprogramas.

No hay una norma fija para dividir un programa en módulos. Se hace a criterio del programador. Sin embargo, se deben seguir unas normas más o menos generalizadas:

- Cada módulo sólo puede tener un punto de entrada y otro de salida.
- El módulo principal debe ser conciso, mostrando claramente los módulos que lo componen y su relación. Es decir, debe de indicar la solución completa del problema.
- Los módulos deben tener la máxima independencia entre ellos.
- Un módulo debe representar por sí mismo una estructura lógica coherente y resolver una parte bien definida del problema.

La programación modular es, pues, una técnica que se basa en el desarrollo de programas de lo general a lo particular. Se comienza considerando que funciones deben realizar el programa desde un punto de vista muy general, dándolas como resueltas y dejando su diseño para un paso posterior. De esta manera se avanza hasta llegar al máximo nivel de detalle.

Es decir, si dentro de un módulo aparece una función cuyo desarrollo completo queremos posponer, esto lo representaremos indicando en ese punto el nombre que tendrá el módulo que lo desarrollará, ocupándonos posteriormente de dicho desarrollo.

- La programación modular aporta una serie de ventajas frente a la programación convencional:
- Los programas son más sencillos de escribir y depurar, pues se pueden hacer pruebas parciales con cada uno de los módulos.
- La corrección o modificación de un módulo se hace más cómoda y, en general, no tiene por qué afectar al resto de los módulos.
- Un programa se puede ampliar fácilmente con sólo diseñar los nuevos módulos necesarios.
- Un mismo módulo escrito una sola vez puede ser referenciado desde varios puntos del programa, evitando la repetición de instrucciones ya escritas.

PROGRAMACIÓN ESTRUCTURADA: ESTRUCTURAS DE CONTROL.

Para aumentar la eficiencia de la programación se necesita que los programas tengan una estructura fácil de interpretar, de modo que los haga más comprensibles, manejables, modificables y fiables.

La programación estructurada es un criterio de programación basado en el Teorema de la Estructura de Bohn y Jacopini que dice lo siguiente "Todo programa propio, es decir, con un solo punto de entrada y un solo punto de salida, que puede ser escrito utilizando únicamente tres tipos de estructuras de control: estructura secuencial , estructura secuencial y estructura repetitiva.

La programación modular y la programación estructurada no son criterios contrapuestos, sino complementarios. El primero tiende a dividir un programa en partes más pequeñas llamadas módulos, y el segundo se encarga de desarrollar estructuralmente cada una de esas partes.

Sin embargo una metodología de la programación sola no es suficiente para crear soluciones al diseñador. Debemos tener las herramientas idóneas, especialmente en forma de lenguajes de programación, esto con la finalidad de poder crear una aplicación buena.

En nuestro tema de tesis mencionamos lo importante que es llevar una metodología con el fin de demostrar que Personal Oracle (el manejador de la base de datos) es un software el cual nos permite llevar una estructura en el sistema, dicho de otra manera , es muy importante el que programemos en módulos para modificaciones posteriores , además de facilitar el entendimiento de los programas que van a integrar al sistema.

De acuerdo a los modelos anteriores de diagramas de flujo y entidad -relación, podemos darnos ya una idea de lo que es la programación y sobre todo de como lo estamos enfocando. La idea de llevar una metodología es básicamente para poder cumplir con las cuatro propiedades que son lo suficientemente generales como para ser aceptadas como metas de una disciplina íntegra de ingeniería de software. Estas cuatro propiedades son: modificabilidad, eficiencia, fiabilidad y comprensibilidad.

Modificabilidad.- Implica un cambio controlado, en el que algunas partes o aspectos permanecen iguales, mientras otras son alteradas. En resumidas cuentas, que cada camino nuevo que uno desee, resulte viable.

Podemos modificar el sistema por una de estas dos razones:

1. Tener que responder a un cambio en los requerimientos del sistema.

2. Tener que corregir un error que introducimos antes del proceso de desarrollo.

Siempre que diseñemos software, debemos tener la estructura de diseño de manera que sea clara y consistente. Para que la modificabilidad del sistema sea efectiva, debe enaltecer las estructuras existentes. Por otra parte estamos obligados a hacer correcciones en el sistema independientemente del diseño original. Después de muchas iteraciones, la estructura original empieza a ser obscura, además de combinar la dificultad de modificación del sistema.

Si los sistemas de software son modificables, sería posible introducir cambios sin incrementar la complejidad del sistema original.

Eficiencia.- Implica que el sistema sobre el que opera el software aprovecha el conjunto de sus recursos de una manera óptima. Podemos clasificar éstos recursos en dos grupos: **recursos de tiempo y recursos de espacio**. Podemos tener tiempo limitado de recursos si un proceso debe ejecutarse en un periodo dado, como por ejemplo, en sensores o en respuesta a un interruptor externo. Obviamente los recursos de tiempo son altamente dependientes de la base de Hardware, aunque nuestra elección de algoritmos de software estará afectada por el tiempo de ejecución. Los recursos de espacio, por otra parte, se refieren a aspectos físicos de la solución, como espacio de direccionamiento o número de periféricos que pueden aprovecharse.

Los conocimientos sobre una mayor comprensión de ubicación de un problema distante, tiene mayor impacto sobre la eficiencia que cualquier montón de piezas sobre una estructura defectuosa.

Fiabilidad.- Es una meta crítica para algunos sistemas que tienen que operar para largos periodos de tiempo sin la intervención del hombre. La fiabilidad debe impedir el fracaso en la concepción, diseño y construcción; al igual que restablecer y recuperar fallos desde operaciones o ejecuciones.

La fiabilidad solamente puede estar construida desde el principio; no puede ser añadida al final.

Comprensibilidad.- Es el puente entre nuestro problema particularmente de espacio y la solución correspondiente. En sistema para ser comprensible debe entenderse a primera vista. Si nosotros estamos reflejando nuestros recursos a un problema difícil, entonces debemos desarrollar una solución con una estructura de diseño clara.

Un sistema dado deberá ser comprensible porque influye en un número de niveles. En el nivel menor, la solución del software sería capaz de leer un resultado del propio código. En un nivel superior seríamos capaces de aislar las estructuras de datos (objetos) y algoritmos (operaciones) de la solución. La comprensibilidad es dependiente del lenguaje de programación que nosotros usemos como vehículo para expresar nuestra solución.

Para poder llegar a una programación adecuada y sobre todo entendible tendremos que tomar en cuenta los puntos anteriores, sin embargo tomemos además en cuenta algunos principios que harán que la modificabilidad, eficiencia, fiabilidad y comprensibilidad se realicen de manera óptima.

Dichos principios son:

- Abstracción
- Información Escondida
- Modularización
- Localización
- Uniformidad
- Integridad
- Confiabilidad

Abstracción e Información Escondida. La esencia de la abstracción consiste en extraer propiedades esenciales mientras omitimos detalles innecesarios. Cuando descomponemos nuestra solución en partes, cada módulo en la descomposición empieza con una parte de abstracción en un nivel dado. Además podemos aplicar la abstracción a la abstracción de algoritmos. Nuestro sistema en base de datos necesitará abrir, cerrar, leer, y escribir esos archivos físicos.

Modularización y Localización. Trata cómo la estructura de un objeto puede hacer más fácil el conocimiento de algún proyecto. Los principios de modularización consiguen descomponer una solución, creando módulos de programas.

Los módulos pueden ser funcionales (orientados a procedimientos) o declarativos (orientados a objetos). Una medida de modularización es la interactividad entre módulos, y de esta forma denominamos ajuste a la "capacidad de fuerza de interconexión" entre los diferentes módulos. Idealmente es deseable disminuir el ajuste entre módulos para que podamos tratar cada módulo relativamente independiente a los otros. Si tenemos un sistema que esta bien estructurado seremos capaces de comprender algún módulo dado relativamente independiente de otros módulos.

Conformidad, Integridad, Confirmabilidad. La conformidad significa que el módulo use una notación consistente y quite algunas diferencias innecesarias. La conformidad generalmente es resultado de un buen estilo de codificación, en el que aplicamos una estructura consistente de control.

Integridad y Confirmabilidad soporta los objetivos de confianza, eficiencia y modificabilidad. Para ayudarnos a desarrollar soluciones correctas. Considerando que la abstracción extrae los detalles esenciales de una entrada dada, la integridad asegura que todos los elementos importantes están presentes. En una palabra, la abstracción y la integridad nos ayudan a desarrollar módulos que sean necesarios y suficientes.

De acuerdo a todo lo anterior podemos llegar a darnos cuenta que tipo de metodología es la que se lleva para poder programar. Cabe mencionar que Personal Oracle nos da la facilidad para realizar de manera modular la programación. Cada uno de los diagramas de flujo de datos y los modelos entidad-relación con los que contamos se pueden dividir por módulos, esto nos lleva a una programación estructurada de tal manera que la alteración o descomposición de un módulo no provocará un paro total del sistema, es por ello que en el sistema se programará en módulos.

IV.6. IMPLEMENTACIÓN , PRUEBAS Y DEPURACIÓN DE PROGRAMAS.

Para realizar un buen sistema de base de datos es siempre necesario realizar pruebas a cada uno de los módulos que lo conforman , por lo que en nuestro caso el sistema de automatización de servicios de cómputo de la UAM Xochimilco consta de los módulos siguientes :

- Alumnos : esté módulo permite dar de alta a los alumnos que requieren de los servicios que el centro de cómputo presta.
- Proyectos : este módulo permitirá dar de alta a los alumnos que fungen como ayudantes de algún proyecto de investigación realizado por algún docente de la UAM y que requieran de los servicios que imparte el centro de cómputo.
- Salas de Cursos : este modulo se encarga de llevar el control de la asignación de cursos a las salas que para esto tiene el centro de cómputo.
- Reparación de Equipos : este modulo es el que llevara el control de los equipos que para su reparación son llevados al taller de reparación y mantenimiento del centro de cómputo.
- Asignación de Equipos : este modulo llevara el control de la asignación de equipos que el centro de cómputo tiene para el uso de alumnos y docentes así como el control de los prestamos del material didáctico con el que cuenta el centro de cómputo.
- Catalogo : Este modulo permitirá llevar el inventario de todos los recursos materiales con los que cuenta el centro de cómputo.

Como se puede notar a simple vista cada uno de estos módulos se encuentra ligado entre si por lo que fue necesario someter a cada uno a un periodo de pruebas antes de que el sistema en su totalidad fuera integrado.

Las pruebas que se realizaron al sistema fueron la siguientes :

- Pruebas de facilidad de uso : Este es un punto muy importante ya que muchos de los usuarios del sistema pidieron que los procedimientos para realizar tal o cual

tarea fueran muy sencillos y fáciles de aprender , por lo que el sistema no requiere un gran número de pasos para realizar alguna tarea lo que simplifica su uso. Para facilitar su uso cuenta con un sistema de ayuda en línea que permite al usuario consultar para saber como se debe realizar determinada tarea o para que sirve determinado icono que presenta el sistema.

- **Pruebas de integridad de información :** Un sistema de base de datos debe asegurar la integridad de la información que maneja no permitiendo información redundante o información corrupta por lo que, nuestro sistema se puso a prueba para evitar estos problemas de integridad de la información , asegurándonos de que las bases de datos no manejaran información redundante y corrupta además se maneja la clave de acceso para asegurar el manejo de información integra. Además se utilizaron algoritmos de validación para evitar el manejo de información corrupta o inconsistente.
- **Pruebas de presentación :** este punto es muy ignorado por algunos desarrolladores de sistemas pero en nuestro caso creamos planillas prototipo con el formato que manejaríamos de las pantallas y se los dimos al usuario para que eligieran el que mas les agrada e igualmente para los reportes que requerían que el sistema generara; ya que para nosotros es muy importante que el usuario se sienta a gusto con su herramienta de trabajo.

Durante las pruebas de cada modulo se realizaron algunas correcciones para el manejo de la información por que existía información redundante; además durante este periodo de pruebas los usuarios llegaron con nuevos requerimientos para anexarle al sistema por lo que el periodo de pruebas se alargó .

Una vez que cada uno de los módulos pasaron las pruebas de integridad de información y facilidad de uso se procedió a integrar cada uno de los módulos en un solo sistema y a darle una mejor presentación.

Cada una de las pruebas realizadas fueron de gran importancia pues con estas logramos dejar satisfecho al usuario permitiendo a la UAM Xochimilco llevar a cabo la automatización de los servicios de cómputo que presta.

IV.6.1. MANTENIMIENTO Y DEPURACIÓN DEL SISTEMA.

Normalmente la actividad de mantenimiento es bastante imprevisible, ya que se requiere de un periodo de observación del comportamiento del sistema en operación real, para determinar si se necesita corregir alguna imprecisión, optimizar algún proceso, incluir alguna función no prevista por el usuario, o bien prevenir futuras cargas mayores de trabajo.

No obstante, con el fin de realizar de la manera más adecuada las actividades de mantenimiento del sistema en cualquiera de sus variedades (correctivo, preventivo, adaptativo ó perfectivo) se considera como una tarea fundamental e indispensable, el hecho de mantener actualizadas las especificaciones conforme ocurran las modificaciones en el sistema a cualquier nivel, es decir si la modificación implica cambios a los documentos de análisis como los diagramas de flujo de datos, estos deben ser efectuados reflejándose también en sus niveles de mayor detalle.

Frecuentemente la urgencia de realizar alguna modificación en los sistemas, provoca que esta actividad quede relegada, sin embargo, es importante mantener actualizadas las especificaciones, ya que además de ser la base documental de soporte para comprender el funcionamiento y la anatomía del sistema, sirven incluso como base para modificaciones posteriores.

Específicamente, en lo que se refiere a mantenimiento adaptativo y perfectivo, desde la puesta en marcha y con la continua operación del sistema, se percibió la posibilidad de incluir ciertos aspectos que pudieran mejorar el rendimiento y ampliar las funciones realizadas, las cuales se describen a continuación:

- Un punto fundamental y frecuente es la creación de reportes no planeados con base en la información producida por el sistema. Al respecto, la medida contemplada consiste en analizar detalladamente cada reporte nuevo solicitado por el usuario, en cuanto a si éste será utilizado por una ocasión única, o bien será requerido constantemente en periodos determinados. En el primero de los casos, la situación será atendida usando generadores externos de reportes, ya

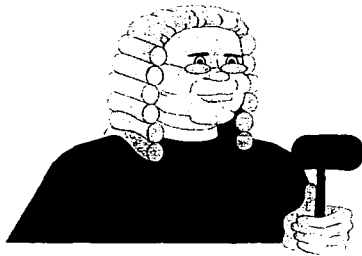
que los formatos de los archivos permiten esta facilidad, pues ORACLE, cuenta con una gran variedad de herramientas que proporcionan soluciones rápida. Por su parte, si los reportes deben ser definitivos, entonces se implantarán permanentemente al sistema (como los ya existentes). En esta última situación, si la premura del caso así lo amerita, la solución inmediata puede atenderse por medio del generador de reportes para incluirlo en el sistema.

- A pesar de que la información que produce el sistema, es un factor fundamental en la toma de decisiones, esta es presentada de acuerdo a los requerimientos operativos planteados inicialmente por el usuario, por lo que se ha detectado la necesidad de realizar informes con resúmenes entendibles del comportamiento de altas, bajas y cambios (manual de usuario) presentados en cuadros y gráficas para facilitar el entendimiento del usuario y hacer más agradable su lectura.

Los puntos mencionados sólo son algunos aspectos que podrían adicionarse al sistema a mejorar su operación o apilar su funcionalidad, sin embargo, las mejoras reales requeridas las irá detectando el usuario de acuerdo a su propia experiencia. Por otra parte, el mismo avance tecnológico es otro factor que marcará la pauta para la evolución del sistema de modo que se adapte a nuevas plataformas operativas, a efecto de prolongar su vida útil.



CONCLUSIONES



CONCLUSIONES.

Durante el desarrollo del presente sistema hemos visto que por muy sencillo que parezca un sistema de bases de datos, su desarrollo puede complicarse conforme se avanza en el proyecto, pues las dificultades técnicas, en realidad, son mínimas en comparación con los problemas que se van presentando en cuanto a la estructura del sistema. Estos problemas pueden ser ya sea de errores en el planteamiento inicial, ya que de todo el universo de posibilidades en el manejo del sistema, siempre habrá algo que quede fuera de nuestra vista. Además, el usuario no tiene una visión precisa de lo que quiere del sistema desde el inicio del proyecto. El usuario puede identificar sus requerimientos incluso aún después de que técnicamente haya sido concluido el sistema.

Es por esto que se debe emplear una buena planeación para tratar de minimizar tanto como sea posible este tipo de situaciones, ya sea durante el desarrollo del sistema y/o en la operación del producto.

Se deben considerar tanto las necesidades del usuario final como las necesidades de la organización, una descripción de como trabaja el sistema precedente y los procesos de los que debe de ser capaz el nuevo sistema. También se deben considerar todas las soluciones alternas y los recursos disponibles. Todo esto debe ser revisado cuidadosamente para evitar ambigüedades en la definición, ningún requisito debe de contradecir u obstaculizar a los demás.

A lo largo del desarrollo del análisis, diseño, implementación, prueba y mantenimiento, es conveniente ir generando documentos autorizados y firmados por el usuario para evitar pérdidas de tiempo en la elaboración de requerimientos que ya habían sido solicitados con anterioridad, evitar omisiones en la implementación del sistema o que el usuario exija que el sistema realice alguna operación que no haya sido especificada previamente.

Por otro lado, un sistema de este tipo es de gran utilidad para el usuario, ya que se evita la redundancia de información, ayuda a mantener la integridad en la información y puede llevarse un mejor control de los recursos (en este caso los recursos computacionales) de la organización. Esto se puede apreciar mucho mejor aún en el caso del Centro de Cómputo de la U.A.M. Xochimilco, dado que hasta antes de este sistema, todas las operaciones se llevaban a cabo manualmente o con escasa ayuda de dispositivos de cómputo.

En resumen, la planeación y el control son parte esencial en el desarrollo de este tipo de sistemas, ya que sin ellos se pueden sufrir grandes retrasos, los costos de implementación aumentan y esto en ocasiones provoca que el solicitante llegue incluso a cancelar el proyecto.



MANUAL TÉCNICO

MANUAL TÉCNICO

El presente manual se elaboró con la finalidad de ayudar al Líder de Proyecto para que en caso de encontrarse con alguna falla, ó desee modificar y/o actualizar alguna propiedad del sistema, tenga bases suficientes para poder hacerlo.

1. MANEJO DE FORMS.

A) Seleccione el icono CDE2 TOOLS.

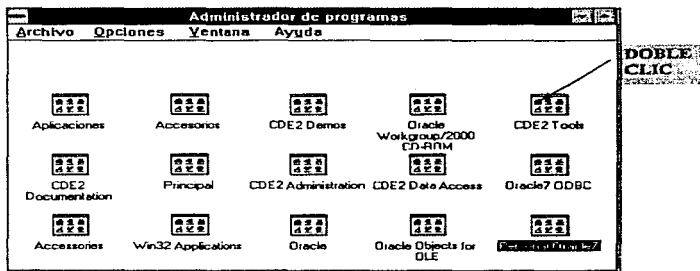


Figura 1.. CDE2 TOOLS

B) Seleccione icono "FORMS DESIGNER" , en el abrir una de las formas listadas a continuación, dependiendo de lo que necesite modificar:

- INSALUM.FMB
- REPARE1.FMB
- EQUIPRE.FMB
- PRE_MAT.FMB
- PRE_SALA.FMB

Ó bien alguno de los catálogos:

- ATENCI2.FMB

- EQUIP2.FMB
- INVEQUI2.FMB
- MARCA.FMB
- MATERJ.FMB
- MODEL.FMB
- SALAS.FMB

Para poder modificar estos, deberá conocer lo siguiente:

2. OBJECT NAVIGATOR.

A) Aquí se encontrarán todas las posibles opciones para poder modificar, ya sea a través de los iconos que aparecen a la izquierda, ó bien por medio del **BOTÓN DERECHO DEL MOUSE**. (Figura 2).

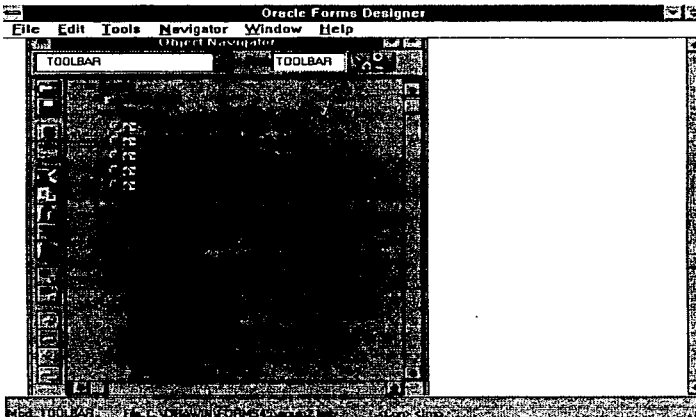


Figura 2. Object Navigator

B) Según la necesidad, podrá seleccionar el **EDITOR DE TEXTOS** ó **LAS PROPIEDADES DE CADA OBJETO**.

EDITOR DE TEXTOS. Con este se elaborará el código necesario para que el sistema realice lo que el usuario requiera. Todo bajo un lenguaje de cuarta generación **PL/SQL**.

El código generado, podrá manejarse a nivel Forma, Bloque ó Campo.

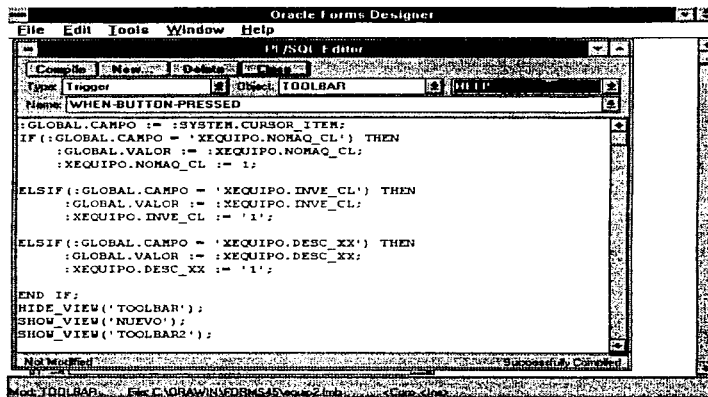


Figura 3. Editor De Texto.

TRIGGERS. Para hacer que el código de algún objeto se dispare en el momento deseado, nos auxiliaremos de los triggers, quienes están predefinidos para su uso, y se dispararán de acuerdo a su prioridad como se indica abajo:

- PRE.....
- POST.....
- ON.....
- WHEN
- KEY

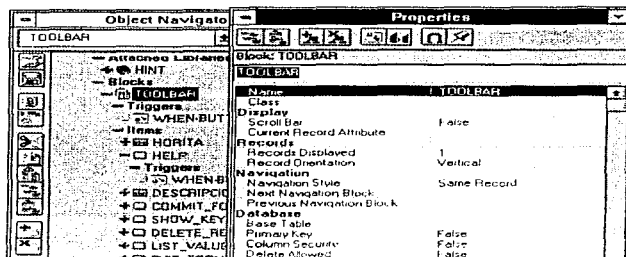


Figura 6. Propiedades.

3. LAYOUT EDITOR. Como todo Manejador de Base de Datos, ORACLE, en su herramienta FORMS, cuenta con el diseñador de pantallas quien nos muestra y permite modificar la forma de acuerdo a los requerimientos del usuario. Tomando en cuenta, que esta es la presentación previa a su ejecución.

Oracle Forms 4.5 (Runform)

Action Edit Block Field Record Query Window Help

INSCRIPCIONES

MATRICULA PASWORD UAM

ADEUDO TIPO INSCRIPCION

NOMBRE PATERNO MATERNO

DOMICILIO TELEFONO

CARRERA TRIMESTRE

CORNER



*MANUAL DE
USUARIO*

Este documento es una guía que tiene como finalidad ayudarlo en el manejo adecuado del sistema: "AUTOMATIZACIÓN DE LOS TRÁMITES PARA PRESTACIÓN DE SERVICIOS DEL CENTRO DE CÓMPUTO DE LA UNIVERSIDAD AUTÓNOMA METROPOLITANA UNIDAD XOCHIMILCO". También se incluye en pantalla un extenso sistema de ayuda (al que podrá acceder en el momento en que usted lo necesite).

ACCESO AL SISTEMA

Estando en ambiente WINDOWS:

1.- DAR DE ALTA LA BASE DE DATOS (B.D.).

A) Seleccione el icono "PERSONAL ORACLE7" (Figura 1)

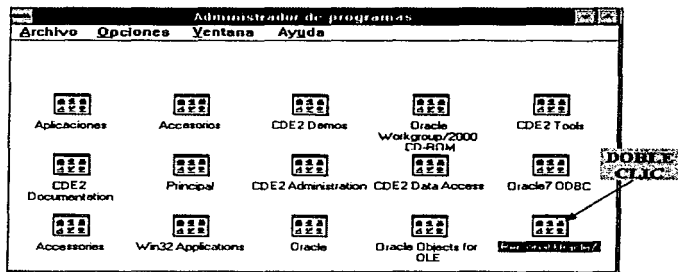


Figura 1. Personal Oracle7

B) Seleccione icono "DATA BASE MANAGER", donde se dará de alta la B.D..

C) Doble Clic en la palabra "START UP" (Figura 2).

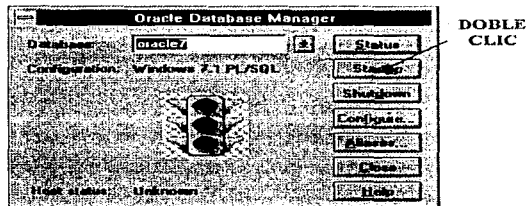


Figura 2. Dando de alta la Base de Datos

D) Teclear la palabra ORACLE que servirá como password para levantar la B.D., presione la palabra OK. (Figura 3.)

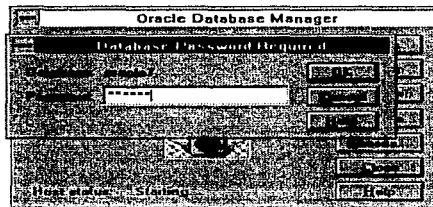


Figura 3. Password

E) Para saber que su B.D. esta levantada, el semáforo de la Figura 2, deberá estar en color verde.

F) Cierre esta ventana

2. Una vez dada de alta la B.D. usted podrá trabajar en el sistema sin ningún problema.
EJECUTANDO EL SISTEMA.

A) Seleccione icono "TESIS" del Administrador de Programas. (Figura 4).

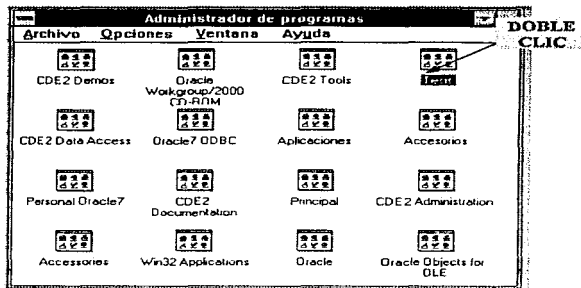


Figura 4. Icono Tesis

B) Doble clic al icono "SIS-UAM"

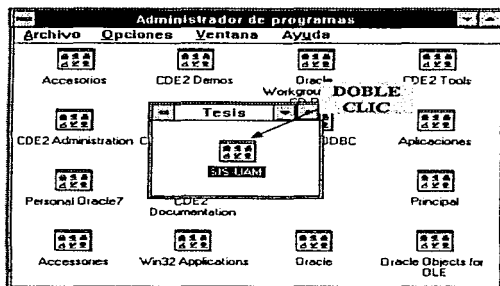


Figura 5. Sistema UAM.

C) Pedirá un PASSWORD para poder acceder al sistema que se encarga de proporcionar seguridad para evitar alguna alteración no autorizada en su B.D.

Los datos válidos a ingresar, son los siguientes:

LOGIN: TESIS
PASSWORD: TESIS
DATA BASE: T: TESIS/TESIS



Figura 6. Conectándose al sistema

D) Aparecerá un menú del cual podrá seleccionar la opción que usted necesite con un doble clic o bien con un ALT más la letra subrayada.



Figura 7. Menú Principal

E) Ejemplo: Si usted selecciona, la opción de **Inscripciones**, se encontrará con la siguiente forma:

Dentro de todas las formas, encontrara una serie de botones que se explican a continuación

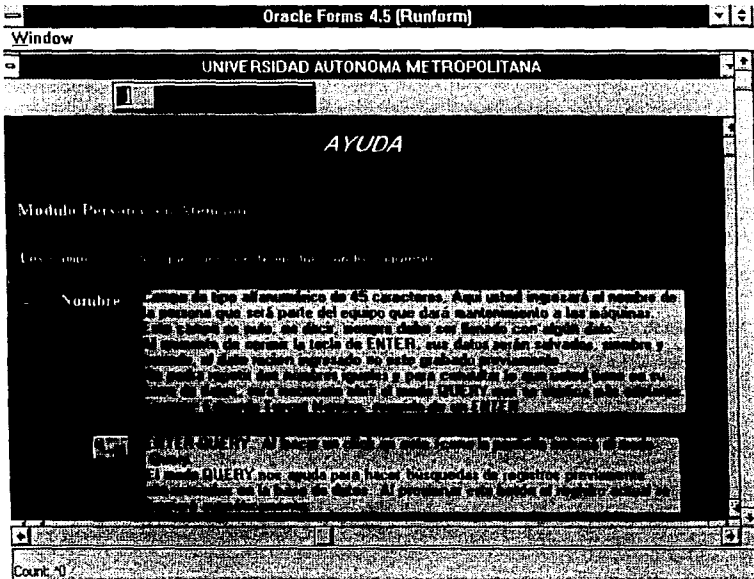


Figura 10. Ayuda

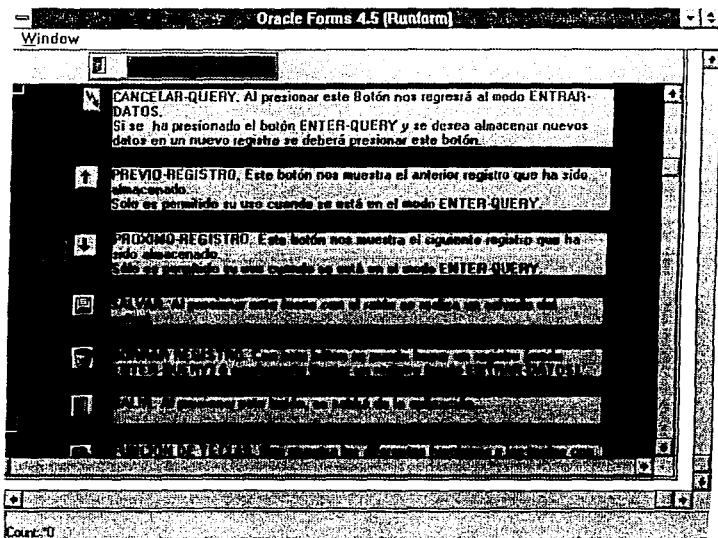
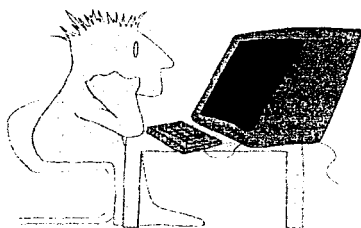


Figura 11. Explicación de iconos.

Los cuadros huecos que aparecen en la parte izquierda de la pantalla, tienen la finalidad de ayudarlo en el desplazamiento de esta pantalla sin ocupar el SCROLL BAR.



GLOSARIO



GLOSARIO

<i>Actualizar.</i>	Modificar datos en un archivo o base de datos. Los términos actualizar y editar se utilizan como sinónimos.
<i>Adaptador.</i>	Véase Tarjeta.
<i>Administración de Datos.</i>	Se refiere a varios niveles de manejo de datos, a partir de métodos de acceso hasta administradores de archivos y DBMS para manejar datos como un recurso organizacional.
<i>Administración de la Información.</i>	Disciplina que analiza la información como un recurso de la organización. Cubre las definiciones, los usos, el valor y la distribución de todos los datos e información dentro de una organización, bien sea procesados o no por una computadora.
<i>Administrador de Bases de Datos (DBA).</i>	Persona responsable del diseño físico y de la administración de la base de datos además de la evaluación, selección e implementación del DBMS.
<i>AIM</i>	Es una asociación que representa a los fabricantes y vendedores de equipos, sistemas y abastecimientos para identificación automática, incluyendo: código de barras, identificación por radiofrecuencia, cintas magnéticas, reconocimiento óptico de caracteres, reconocimiento de voz y sistemas de visión.
<i>Internacional (Automatic Identification Manufacturers).</i>	Utilización combinada de letras del alfabeto con números y caracteres especiales.
<i>Algoritmo.</i>	Conjunto de pasos ordenados para resolver un problema.
<i>Alias.</i>	Nombre alternativo usado para la identificación, como denominar un campo o un archivo.
<i>Analista de Sistemas.</i>	Persona responsable del desarrollo de un sistema de información. Realiza el diseño y modifica los sistemas transformando las necesidades del usuario en un conjunto de especificaciones funcionales que constituyen el programa detallado de acción del sistema. Diseña la base de datos, o ayuda en su diseño si se dispone de administradores de datos. Desarrolla los procedimientos manuales y de máquina, y las especificaciones detalladas de procesamiento para cada ingreso de datos, actualización, programa de consulta e informes en el sistema.
<i>Analizador.</i>	Instrumento electrónico de laboratorio para control de códigos de barras para preparación y verificación de impresiones, de muy alta sensibilidad y precisión, para leer, decodificar y analizar en detalle distintos tipos de códigos. Generalmente permite también analizar los colores, contrastes y reflectancias.
<i>Apertura del Scanner.</i>	Se refiere al orificio a través del cual el haz de luz reflejado en el objeto, retorna al scanner y su elección adecuada define los parámetros del haz para ajustarlos a las dimensiones físicas de las barras en el código sobre el objeto explorado, o sea el foco.
<i>Arquitectura Cliente / Servidor.</i>	Modelo de diseño para aplicaciones que corren redes de área local, en la que la mayor parte del proceso final a realizar -una búsqueda física en una base de datos, por ejemplo- se lleva a cabo en el servidor. El proceso de la fase delantera, que implica comunicación con el usuario, lo manejan varios programas pequeños, distribuidos en las estaciones de

GLOSARIO

- Arquitectura de Red.* Conjunto total de hardware, software y pautas de instalación de cables que estipula el diseño de una red.
- Arquitectura de Rendimiento Graduable (SPARC).* Unidad Central de Proceso RISC de 32 bits desarrollada por Sun y autorizada por SPARC International (Standard Planning and Requirements Committee). Se utiliza en estaciones de trabajo SPARCStation de Sun.
- Arquitectura.* Diseño total mediante el cual se interrelacionan los componentes individuales del hardware de una computadora.
- ASCII.* De las siglas en inglés de American Standard Code for Information Interchange. Es un código binario de datos que se usa en comunicaciones, en la mayor parte de las minicomputadoras y en la mayor parte de las computadoras personales. Sólo los primeros 128 caracteres (0-127) dentro de las 256 combinaciones de un byte constituyen el estándar ASCII. El resto se utiliza en forma diferente de acuerdo con la computadora.
- Atributo.* En administración de bases de datos relacionales, es un campo dentro de un registro.
- Banda Magnética.* Pequeño tramo de cinta que se adhiere a las fichas contables, tarjetas de identificación y de crédito. Las bandas magnéticas son leídas por lectores especializados que pueden incorporarse a máquinas de contabilidad y terminales. Debido al intenso uso, los datos en la banda magnética se encuentran en un formato de baja densidad que puede duplicarse varias veces.
- Barra.* Línea más larga que ancha, por lo general de color negro o muy oscuro, de ancho variable entre uno y varios módulos, capaz de absorber (y no reflejar) la luz del scanner y que es uno de los elementos del código.
- Base de Datos Distribuida.* Base de datos que está físicamente almacenada en dos o más sistemas computacionales. Aunque se encuentra geográficamente dispersa, administra y controla toda la base de datos como un conjunto único de datos.
- Base de Datos Relacional.* Método de organización de base de datos que determina las relaciones entre archivos cuando se requieren. En vez de tener enlaces fijos predeterminados, o apuntadores, entre archivos, una base de datos relacional enlaza los archivos mediante comparación.
- Base de Datos.* Conjunto de archivos interrelacionados creado y manejado por un sistema de gestión o de administración de bases de datos (DBMS).
- Binario.* Sistema alfanumérico que sólo utiliza dos elementos: "1 y 0".
- Bit de Inicio.* En comunicaciones asincrónicas, bit que se transmite antes de cada carácter.
- Bit de Paridad.* En comunicaciones asincrónicas y almacenamiento primario, bit extra que se agrega a una palabra de datos para la verificación de paridad.
- Bit de paro.* En comunicaciones en serie, bit insertado en el flujo de datos de datos para informar a la computadora receptora que en seguida aparecerá un byte de datos.
- Bit.* Dígito simple de un número binario (1o 0).
- Buffer.* Segmento reservado de memoria que se utiliza para almacenar datos

	mientras se procesan. Un buffer también puede ser un pequeño banco de memoria utilizado para fines especiales.
<i>Bug.</i>	Error persistente en el software o hardware. Si existe en el software, debe corregirse modificando el programa. Si existe en el hardware, deben diseñarse nuevos circuitos.
<i>Bus.</i>	Canal o ruta común entre dispositivos de hardware.
<i>Byte.</i>	Unidad común de almacenamiento en computación, desde micros hasta mainframe. Se compone de ocho dígitos binarios (bits).
<i>Cable Coaxial.</i>	En redes de área local, cable de conexión con amplitud de banda alta en el que corre un alambre aislado a lo largo de su centro. Un segundo alambre hecho de metal sólido o en forma de malla rodea al alambre aislado.
<i>Cable de Pares Trenzados.</i>	En redes de área local, cable de conexión de amplitud de banda baja empleado en sistemas telefónicos. El cable incluye dos alambres trenzados uno al rededor del otro para reducir la interferencia proveniente de otros cables.
<i>Campo</i>	Unidad física de datos que ocupa uno o más bytes. Una colección de campos forman un registro. Un campo también define una entidad de datos en un documento fuente, pantalla o informe. El campo es el común denominador entre el usuario y la computadora.
<i>Caracter Final.</i>	Indica al scanner el final del código, puede estar formado por un número, letra o símbolo según el código.
<i>Caracter Inicial.</i>	Indica al scanner el comienzo del código, puede estar formado por un número, letra o símbolo según el código.
<i>Caracter tipo OCR-B.</i>	Forman la línea de interpretación que permite al hombre leer los caracteres codificados o símbolos.
<i>Ciclo de Vida del Sistema.</i>	Vida útil de un sistema de información. Su duración depende de la naturaleza o volatilidad del negocio, así como de las herramientas de desarrollo de software utilizadas para generar las bases de datos y las aplicaciones. Con el tiempo, un sistema de información que ha sido remediado una y otra vez ya no posee una firmeza estructural suficiente como para ser expandido.
<i>Código Bidireccional.</i>	Es el código capaz de ser leído por el scanner en ambas direcciones aunque luego será decodificado electrónicamente en la dirección correcta.
<i>Código Continuo.</i>	Es aquel en donde cada caracter está a continuación del otro, sin que existan intervalos mudos, o sea que todos los espacios forman parte de la codificación (es lo opuesto al código discreto).
<i>Código de Barras.</i>	Código impreso utilizado para un reconocimiento mediante un lector óptico de barras. Los códigos de barras tradicionales unidimensionales utilizan el ancho de barra como el código, pero codifican sólo un número de identificación o de cuenta. Los sistemas bidimensionales como PDF 417 de Symbol Technology, retienen 1,800 caracteres en un área del tamaño de un timbre postal.
<i>Código Discreto.</i>	Es aquel en donde cada caracter es independiente y está separado del siguiente por una zona neutra llamada intervalo mudo que no forma parte del código (es lo opuesto al código continuo).

GLOSARIO

<i>Código fuente.</i>	Conjunto de sentencias de programación tal como fueron escritas por el programador. El código fuente debe convertirse en lenguaje de máquina mediante compiladores, ensambladores e intérpretes antes de ejecutarse en la computadora.
<i>Código Objeto.</i>	Ver lenguaje de máquina.
<i>Coefficiente de Primera Lectura.</i>	Es el porcentaje de lecturas correctas que producirá el scanner en un solo paso por el código explorado. Indica la velocidad con que podrá operar un scanner y un determinado símbolo impreso.
<i>Compatibilidad.</i>	Es la capacidad de un código de ser leído e interpretado en otro sistema distinto.
<i>Compilador.</i>	Software que traduce los lenguajes de programación de alto nivel a lenguaje de máquina. Un compilador habitualmente genera en primer lugar un lenguaje ensamblador y a continuación traduce este último a lenguaje de máquina.
<i>Conector.</i>	Cualquier cable que une dos dispositivos.
<i>Construcción de Prototipos.</i>	Creación de un prototipo de un nuevo sistema. Mediante lenguajes de cuarta generación, los analistas de sistemas y los usuarios pueden desarrollar un nuevo sistema en forma conjunta. Las bases de datos pueden crearse y manipularse mientras el usuario monitorea el progreso.
<i>Contraste.</i>	Es la acción y efecto de oposición entre el color de las barras y el de los espacios. El correcto funcionamiento del scanner se basa en el reconocimiento de este contraste entre el color de los elementos. Los colores y contrastes del código corresponden a especificaciones muy precisas y nunca deben ser elegidos sin consultar antes una guía de colores y contrastes.
<i>Cursor.</i>	Símbolo móvil en una pantalla que sirve como punto de contacto entre el usuario y los datos.
<i>Chip.</i>	Pequeña sección de un cristal sencillo de material semiconductor, generalmente silicio, que forma el sustrato sobre el que se fabrica un solo dispositivo semiconductor o todos los dispositivos individuales que constituyen un circuito integrado.
<i>Datos fuente.</i>	Datos originales manuscritos o impresos en un documento fuente, o introducidos en el sistema computacional mediante un teclado o terminal.
<i>Datos.</i>	Cualquier forma de información bien sea en papel o en forma electrónica.
<i>Deadlock</i>	Estancamiento que ocurre cuando dos elementos en un proceso están cada uno, esperando a que responda el otro. El software debe ser capaz de encargarse de esto.
<i>Densidad del Código.</i>	Es la relación entre la cantidad de caracteres codificados y la longitud que ocupan una vez impresos. Se expresa generalmente en caracteres por centímetro o pulgada.
<i>Desarrollador de Aplicaciones.</i>	Individuo que desarrolla una aplicación comercial; generalmente realiza los servicios de analista de sistemas y programador de aplicaciones.
<i>Diagrama de Flujo</i>	Representación gráfica de la secuencia de operaciones en un sistema de información o programa.
<i>Diccionario de</i>	Base de datos acerca de datos y bases de datos. Contiene el nombre, tipo,

<i>Datos.</i>	rango de valores, fuente y autorización para el acceso a cada elemento de datos en los archivos y bases de datos en la organización.
<i>Dígito de Verificación.</i>	Es un número incluido en el código, calculado por un algoritmo que emplea los restantes números de código. Su función es detectar errores durante la lectura o scanning para evitar las lecturas erróneas. También se le utiliza como clave para evitar adulteraciones.
<i>Dígito.</i>	Cada uno de los símbolos numéricos o alfabéticos, iguales entre sí por el criterio de codificación elegido, que difieren en su valor y forma de representación.
<i>Dimensión Nominal o Estandar.</i>	Es la longitud y superficie de un código cuando el factor de magnificación $fm=1$ (o 100%).
<i>Dirección.</i>	Número de una ubicación particular de memoria o de almacenamiento periférico. Cada byte de memoria y cada sector de un disco poseen su dirección.
<i>Dispositivo.</i>	Cualquier máquina electrónica o electromagnética, o componente, de un transistor a una unidad de disco. Un dispositivo siempre se refiere a hardware.
<i>Drive</i>	Dispositivo electromecánico que gira discos y cintas a una velocidad especificada. También se refiere a la unidad periférica completa.
<i>EAN (International Article Numbering Association).</i>	La unión de representantes de 12 países europeos para el estudio de un método unificado de codificación. Actualmente cuenta con 48 países miembros.
<i>Elementos del Código.</i>	Son las barras, espacios, zonas mudas; en general todos los elementos que se imprimen dentro del rectángulo formado por las cuatro señales de encuadre.
<i>Emulador</i>	Dispositivo que se construye para trabajar como otro. Una computadora puede ser diseñada para emular otra computadora y ejecutar software que fue escrito para ejecutarse en la otra máquina. El emulador puede ser de hardware, software o de ambos.
<i>Entidad</i>	En una base de datos, cualquier cosa acerca de la cual se pueda almacenar información. Generalmente se refiere a una estructura de registro.
<i>Equipo de Comunicación de Datos (DCE).</i>	Lado de una interfaz que representa al suministrador de una comunicación de datos en una norma tal como RS232C o X.25. Los DCE son módems analógicos o digitales, por regla general.
<i>Equipo de Terminación de Datos (DTE).</i>	Por lo general, una terminal o computadora, es un dispositivo de comunicaciones que representa la fuente o destino de señales en una red.
<i>Escala Patrón de Impresión.</i>	Está compuesta por líneas paralelas dispuestas en 11 grupos que se identifican con las letras A-K, en sentido longitudinal y las letras A'-K' en sentido transversal.
<i>Espacio.</i>	Línea, más larga que ancha, por lo general de color blanco o muy claro, de ancho variable entre uno o varios módulos, capaz de reflejar la luz del scanner, que generalmente es el fondo sobre el que están impresas las

GLOSARIO

<i>Especificación Funcional</i>	barras, es uno de los elementos del código. Proyecto para el diseño de un sistema de información. Provee documentación para los procedimientos de base de datos, humanos y de máquina, y todos los detalles de introducción, procesamiento y salida para cada entrada, consulta o actualización de datos y programas de informes en el sistema.
<i>Estándar.</i>	Conjunto de reglas y regulaciones acordado por una organización oficial de estándares (estándar legal) o por aceptación general en el mercado (estándar de hecho).
<i>Ethernet.</i>	Red de área local (IDEE 802.3) que transmite a 10Mbits/seg y puede conectarse en total hasta 1,024 nodos. El Ethernet estándar o "thick Ethernet" usa una topología de bus con una longitud de segmento máxima de 1,640 pies y 100 dispositivos.
<i>Factor de Magnificación.</i>	Adoptando como tamaño normal las dimensiones estándar de un símbolo y su factor de magnificación $m = 1$, es posible aumentar o reducir el tamaño relativo del mismo hasta 2.0 máximo y hasta 0.8 veces como mínimo.
<i>Flujo de Datos.</i>	El trayecto de los datos a partir del documento fuente al ingreso de datos, al procesamiento hasta los informes finales.
<i>Función</i>	En programación, rutina de software que hace una tarea particular. Cuando el programa pasa el control a una función, esta realiza la tarea y devuelve el control a la instrucción siguiente a la que la llamó.
<i>Ganancia de Impresión. Generador de Aplicaciones.</i>	Aumento o reducción de tamaño de la barra impresa, respecto a la película original. Software que genera programas de aplicación a partir de las descripciones del problema en vez de hacerlo con la programación tradicional.
<i>Hardware.</i>	Maquinaria y equipo (CPU, discos, cintas, módem, cables, etc.). En una operación, una computadora es tanto el hardware como el software. El uno no sirve sin el otro. El diseño de hardware especifica los comandos que puede seguir y las instrucciones que le dicen qué debe hacer.
<i>Host.</i>	Computadora central en un entorno de procesamiento distribuido. Por lo general, se refiere a una gran computadora de tiempo compartido o una computadora central que controla una red.
<i>Huecos.</i>	Son manchas claras en las barras causadas por defectos de impresión en el código.
<i>Independencia de Datos.</i>	Técnica de DBMS que separa los datos desde el procesamiento y permite cambiar estructuralmente la base de datos sin afectar los sistemas existentes.
<i>Índice o Indexación.</i>	En administración de datos, método más común para llevar un seguimiento de los datos en disco. Los índices son listados de directorios que mantienen el sistema operativo, el DBMS o la aplicación.
<i>Instituto Nacional de Estándares Americanos (ANSI).</i>	Coordina el desarrollo de estándares voluntarios a nivel nacional (E.U.) que incluye lenguajes de programación, telecomunicaciones y propiedades de medios de disco y cinta.
<i>Integridad de</i>	Proceso de evitar el borrado o adulteración accidental en una base de

<i>Datos.</i>	datos.
<i>Interconexión de Sistemas Abiertos (OSI).</i>	Estándar ISO para comunicaciones a nivel mundial que define una estructura con el fin de implementar protocolos en siete estratos o capas. El control se transfiere de un estrato al siguiente, comenzando en el estrato de aplicación de una estación, llegando hasta el estrato inferior, por el canal hasta la próxima estación y subiendo nuevamente la jerarquía.
<i>Interfaz de Usuario.</i>	Combinación de menús, diseño de pantalla, órdenes de teclado, lenguaje de comandos y pantallas de ayuda, que constituyen la manera como un usuario interactúa con una computadora.
<i>Intervalo Mudo.</i>	Es el espacio que separa un caracter de otro en un código de tipo discreto y no forma parte de la codificación.
<i>Láser.</i>	Es un haz de luz en el cual las ondas se propagan en forma coherente en fase y sin dispersarse, logrando una muy alta concentración de la energía.
<i>Lectura Errónea.</i>	Es cuando el scanner lee y decodifica la información que no concuerda con la originalmente codificada en memoria.
<i>Lectura Nula.</i>	Es cuando el scanner lee un código, pero no lo reconoce como tal.
<i>LED (Diodo Emisor de Luz).</i>	Es un dispositivo de estado sólido similar a un transistor, capaz de transformar pequeñas cantidades de energía eléctrica en lumínica.
<i>Lenguaje de Definición de Datos (DDL).</i>	Es la definición de la base de datos según el modelo de base de datos. Sirve para la declaración de las tablas y creación de ellas según el modelo de red jerárquico o relacional.
<i>Lenguaje de Manipulación de Datos (DML).</i>	Contiene lenguajes de tercera y cuarta generación además de funciones de procesamiento.
<i>Lenguaje de Máquina.</i>	Lenguaje natural de la computadora. Para ejecutar un programa, este debe estar en el lenguaje de máquina de la computadora que lo acciona.
<i>Lenguaje Natural.</i>	Lenguaje que ocurre en forma natural - como el español, el francés, el inglés, etc- y que es distinto al artificial, como un lenguaje de programación en computadora.
<i>Lenguaje Procedimental.</i>	Lenguaje de programación que requiere una disciplina en programación. Los programadores emplean estos lenguajes para desarrollar un orden adecuado de acciones para resolver un problema.
<i>Línea de Interpretación.</i>	Son los símbolos codificados en el código de barras que se imprimen en forma legible al ser humano, generalmente al pie del código.
<i>Longitud del Código.</i>	Fija: El ancho total es fijo y no depende de la información codificada.
<i>Llave.</i>	Datos que identifican un registro. Como identificador, cada valor de la llave debe ser único para cada registro.
<i>Mainframe.</i>	Computadora de gran capacidad de almacenamiento y procesamiento.
<i>Mantenimiento.</i>	El mantenimiento de programas o software es la actualización de programas de aplicación con el fin de satisfacer las cambiantes necesidades de información.
<i>Mapa.</i>	Conjunto de datos que tiene una relación de correspondencia con otro conjunto de datos.
<i>Mapeo.</i>	Proceso para convertir datos codificados en un formato o dispositivo a

GLOSARIO

<i>Marco de Soporte.</i>	otro formato o dispositivo. Es el marco o borde externo formado por una barra gruesa de forma rectangular que rodea a todo el código por fuera, no es interpretado por el scanner y sirve como soporte a las barras del código para emparejar su desgaste en las impresiones flexográficas y especialmente para la impresión de ciertos substratos como el cartón corrugado, previene errores de lectura.
<i>Memoria Virtual.</i>	Técnica que simula más memoria que la que realmente existe, al descomponer el programa en segmentos, llamados páginas, y llevar tantas páginas a la memoria como sea posible. El resto de las páginas quedan en disco hasta que se requieran.
<i>Memoria.</i>	Area de trabajo de la computadora. Es un recurso importante, puesto que determina el tamaño y la cantidad de programas que pueden ejecutarse al mismo tiempo, así como también la cantidad de datos que pueden procesarse en forma instantánea.
<i>Microprocesador</i>	UCP en un sólo chip. Para funcionar como una computadora, requiere suministro de energía, reloj y memoria.
<i>Mimicomputador</i>	Computadora a mediana escala que funciona como una sola estación de trabajo, o como un sistema multiusuario de hasta varios cientos de terminales.
<i>Mínima Diferencia de Reflectancia.</i>	Relación más pequeña, entre cantidad de luz reflejada por una superficie (el código) y la reflejada por un patrón estándar.
<i>Modelo de Datos.</i>	Descripción de los principios de organización de una base de datos.
<i>Modelo Entidad-Relación</i>	En una base de datos, modelo de datos que describe atributos de entidades y relaciones entre estas.
<i>Módem.</i>	Dispositivo que adapta una terminal o computadora a una línea telefónica. Convierte las pulsaciones digitales de la computadora en audio-frecuencias y vuelve a convertir estas en pulsaciones en el lado receptor.
<i>Módulo.</i>	Es el elemento más angosto, sea barra o espacio. Todos los elementos del código incluyendo zonas mudas y separadores, poseen un ancho, múltiplo del módulo por lo general. El tamaño del módulo define directamente la densidad, y es la dimensión "X" o nominal de un código.
<i>Multitarea.</i>	Ejecución de dos o más programas en una computadora al mismo tiempo. La multitarea se controla mediante el sistema operativo. La cantidad de programas que pueden realizar tareas múltiples efectivamente depende de la cantidad de memoria disponible, la velocidad de la UCP, capacidad y velocidad del disco duro, así como la eficiencia del sistema operativo.
<i>Multiusuario.</i>	Computadora compartida por dos o más usuarios.
<i>Nodo.</i>	En comunicaciones, punto de empalme o de conexión en una red. En administración de base de datos, elemento de datos al que se puede tener acceso por dos o más rutas.
<i>Normalización.</i>	Proceso de optimización de diseño de base de datos basado en la reestructuración de tablas para disminuir la redundancia de información.

<i>Organización Internacional de Estándares (ISO).</i>	Muchos estándares técnicos que incluyen el OSI para comunicaciones a nivel mundial. Ginebra.
<i>Palabra.</i>	Unidad interna de almacenamiento de la computadora. Se refiere a la cantidad de datos que puede contener en sus registros.
<i>Paquete.</i>	Grupo de bits o caracteres que se transmiten como una sola unidad.
<i>Parámetro.</i>	Cualquier valor que se suministre a un programa o rutina con el fin de modificarlo. Algunas veces el software se escribe para aceptar direcciones opcionales, que se introducen en la línea de comando con el nombre del programa cuando este se carga.
<i>Partición.</i>	Parte reservada del disco o de memoria que se deja aparte para algún propósito.
<i>Password.</i>	Palabra o código utilizado como un medio de seguridad contra el acceso no autorizado a los datos.
<i>Path.</i>	Camino, vía de acceso, ruta.
<i>Periférico.</i>	Cualquier dispositivo de hardware conectado a una computadora, como monitor, teclado, impresora, plotter, mouse, etc.
<i>Procesamiento de Datos.</i>	Captura, almacenamiento, actualización y recuperación de datos e información. Este término puede referirse a toda la industria de la computación o al procesamiento de datos en contraste con otras operaciones, como procesamiento de palabras.
<i>Procesamiento Múltiple.</i>	Procesamiento simultáneo con dos o más procesadores en una computadora, o dos o más computadoras que están procesando juntas. Cuando se usan dos o más computadoras, estas se unen a un canal de alta velocidad y comparten entre sí la carga de trabajo general. En caso de que uno falle, el otro se hace cargo.
<i>Procesamiento Paralelo.</i>	Una operación se realiza en muchos conjuntos de datos, o se trabaja en diferentes del trabajo de manera simultánea.
<i>Profundidad de Campo.</i>	Es la diferencia entre las distancias máxima y mínima del scanner al objeto codificado, para que pueda ser leído
<i>Programa.</i>	Conjunto de instrucciones que indican que debe hacer la computadora. Un programa se denomina software.
<i>Programación Estructurada.</i>	Variedad de técnicas que impone una estructura lógica en la escritura de un programa.
<i>Protocolo de Control de Transmisiones / Protocolo Internet (TCP/IP).</i>	Protocolos de comunicaciones desarrollados bajo contrato del Departamento de Defensa de los Estados Unidos para intercomunicar sistemas diferentes. Es un estándar UNIX de hecho pero está respaldado por sistemas operativos de micro a mainframe. Es utilizado por muchas corporaciones y casi todas las universidades y entidades federales.
<i>Protocolo.</i>	Acuerdo que dirige los procedimientos utilizados para el intercambio de información entre entidades que colaboran.
<i>Puente.</i>	En redes de área local, dispositivo que permite el intercambio de información entre dos redes (incluso las de topologías, alambrado o protocolos de comunicación diferentes).
<i>Query.</i>	Consultar una base de datos. Es un listado selectivo de elementos.
<i>Ranura de expansión.</i>	Receptáculo integrado al bus de expansión de la computadora y diseñado para aceptar adaptadores.

GLOSARIO

<i>Reconocimiento Óptico de Caracteres (OCR).</i>	Reconocimiento, por medio de una máquina, de caracteres impresos. Los sistemas OCR pueden reconocer muchos tipos diferentes de OCR, así como caracteres de máquinas de escribir e impresos por computadora.
<i>Red de Área Ancha (WAN).</i>	Red de comunicaciones que cubre amplias áreas geográficas, como estados y países.
<i>Red de Área Local (LAN).</i>	Enlace de computadoras personales y de otro tipo dentro de un área limitada por medio de cables de alto desempeño con el fin de que los usuarios puedan intercambiar información, compartir periféricos costosos y recurrir a los recursos de la unidad de almacenamiento masivo secundario, llamado servidor de archivos.
<i>Red.</i>	Sistema de intercambio de comunicaciones e información basado en computadora creado mediante la conexión física de dos o más computadoras.
<i>Reducción Ancho de Barra.</i>	Como resultado de las pruebas de impresión con la escala patrón se reducen o aumentan los anchos de cada barra en la película original para compensar la ganancia de impresión.
<i>Reflectancia. Registro.</i>	Es la relación entre el flujo lumínico incidente y el reflejado. Grupo de campos relacionados que almacenan datos acerca de un tema (registro maestro) o actividad (registro de transacción). Un conjunto de registros constituye un archivo.
<i>Relación Aumento / Reducción. Repetidor.</i>	Es la relación que existe entre los anchos de los elementos más anchos y más angostos del código. En redes de área local, dispositivo de hardware empleado para prolongar la extensión del cableado de la red, con el que se amplifican y transmiten los mensajes y pasan a través de la red.
<i>RS-232-C.</i>	Estándar de EIA (Electronics Industries Association) para una interfaz serial entre computadoras y dispositivos periféricos.
<i>Ruteador (Router).</i>	En comunicaciones, dispositivo que examina la dirección de destino de un mensaje y selecciona la ruta más efectiva. Los ruteadores se utilizan en redes complejas donde hay muchas vías de acceso entre usuarios.
<i>Scanner.</i>	Es el transductor que transforma la información impresa en un código, mediante la emisión y recepción de luz, en impulsos eléctricos digitales capaces de alimentar una computadora.
<i>Sector.</i>	Unidad más pequeña de almacenamiento leída o escrita en un disco. Los sectores son fijos en cuanto a longitud, y en una pista reside generalmente la misma cantidad de sectores. Sin embargo, el hardware puede variar la velocidad del disco para acomodar más sectores en las pistas localizadas en los bordes externos del plato del disco.
<i>Señales de Encuadre. Separadores.</i>	Son las marcas o puntos que limitan externamente a un código y sus elementos. Son caracteres auxiliares formados por barras y espacios que generalmente advierten al scanner los extremos del código y la dirección en que la información es recibida, permitiendo también la lectura bidireccional. También se utilizan separadores dentro del código para separar zonas.
<i>Servidor.</i>	En la red, computadora que es compartida por varios usuarios.

<i>Sesión.</i>	El uso de un programa de aplicación (periodo entre el arranque y la conclusión).
<i>Shell.</i>	Capa exterior de un programa que proporciona la interfaz del usuario, o medio para dar órdenes a la computadora. Las cápsulas, por lo general, son programas agregados, creados para sistemas operativos manejados por comandos.
<i>Sintaxis.</i>	Reglas que rigen la estructura de una sentencia en un lenguaje. Especifica la manera como se unen las palabras y los símbolos para formar una frase.
<i>Sistema de Administración de Base de Datos (DBMS).</i>	Software que controla la organización, el almacenamiento, la recuperación, la seguridad y la integridad de los datos en una base de datos. Acepta solicitudes de la aplicación y genera las órdenes al sistema operativo para que transfiera los datos apropiados.
<i>Sistema de Información.</i>	Aplicación comercial para la computadora. Está constituida por la base de datos, los programas de aplicación, los procedimientos manuales y automatizados, e incluye los sistemas computacionales que realizan procesamiento.
<i>Sistema Distribuido.</i>	Sistema de computadora diseñado múltiples usuarios que proporciona a cada uno una computadora funcionalmente completa. Un sistema distribuido está diseñado para facilitar la comunicación de las computadoras enlazadas y ofrecer acceso compartido a archivos centrales.
<i>Sistema Jerárquico de archivos.</i>	Método de organización de archivos que almacena datos en una estructura organizacional descendente. Todo acceso a los datos comienza desde arriba y procede a lo largo de los niveles de la jerarquía.
<i>Sistema.</i>	Grupo de componentes relacionados que interactúan para realizar una tarea.
<i>Sistemas.</i>	Término general para el departamento, personal o trabajo relacionado con las actividades de análisis y diseño de sistemas.
<i>Software.</i>	Instrucciones para la computadora. Una serie de instrucciones que realiza una tarea en particular se llama programa o programa de software. Las dos categorías principales son software de sistemas y software de aplicaciones.
<i>SQL</i>	Structure Query Lenguaje (Lenguaje estructurado de consulta) Es utilizado para interrogar y procesar datos en una B.D. relacional. Los comandos SQL pueden utilizarse para trabajar interactivamente una B.D. ó pueden incluirse en un lenguaje de programación para servir de interface en estas.
<i>Subrutina.</i>	Grupo de instrucciones que realizan una función específica. Una subrutina grande se denomina usualmente módulo o procedimiento; una pequeña, función o macro, pero todos los términos se identifican entre sí.
<i>Tabla.</i>	En programación, un conjunto de campos adyacentes. También denominada arreglo, una tabla contiene datos que pueden ser constantes dentro del programa o ser introducidos cuando el programa esté en ejecución. En administración de bases de datos relacionales, lo mismo que un archivo; un conjunto de registros.

GLOSARIO

<i>Tarea.</i>	Unidad de trabajo que se ejecuta en la computadora. Una tarea puede ser un programa o un grupo de programas que deben trabajar juntos.
<i>Tarjeta.</i>	Circuito electrónico cuyo diseño permite insertarlo en las ranuras de expansión de una computadora. Es sinónimo de adaptador.
<i>Terminador.</i>	Caracter de finalización; caracter de detención; caracter de parada. Símbolo que marca el final de una sentencia en un lenguaje de programación.
<i>Terminal.</i>	Dispositivo de entrada/salida de una computadora, que posee normalmente un teclado para la entrada y una pantalla de video o impresora para la salida.
<i>Tiempo de respuesta.</i>	Tiempo que tarda una computadora en satisfacer un pedido del usuario, como buscar un registro de cliente.
<i>Tiempo real.</i>	Respuesta inmediata. Se refiere al control de procesos y sistemas incorporados. También se refiere a sistemas de procesamiento rápido de transacciones.
<i>Tipo de Dato.</i>	Categoría de datos. Los tipos de datos comunes son numéricos, alfanuméricos, fechas y datos lógicos (booleanos).
<i>Topología.</i>	En una red de comunicaciones, patrón de interconexión entre nodos; por ejemplo, una configuración de bus, de anillo o de estrella.
<i>Transceiver (emisor-receptor).</i>	Transmisor y receptor de señales analógicas o digitales que viene en muchos formatos; por ejemplo, un receptor y transmisor de un satélite de comunicaciones o un adaptador de redes.
<i>Transparencia de Datos.</i>	Capacidad para tener acceso y trabajar con datos fácilmente, sin importar dónde estén localizados o qué aplicación los creó.
<i>Unidad Central de Proceso (UCP).</i>	También llamada procesador es la parte de cálculo o "cerebro" de la computadora, que está constituida por la unidad de control y la unidad aritmético-lógica. La unidad central de procesamiento obtiene sus instrucciones y datos de la memoria y contiene los circuitos que realizan las operaciones matemáticas y lógicas en los datos.
<i>Usuario.</i>	Cualquier individuo que interactúa con la computadora a nivel de una aplicación.
<i>Variable.</i>	En programación, estructura que contiene datos y que recibe un solo nombre dado por el programador. Mantiene los datos asignados a esta hasta que se le asigne un nuevo valor o hasta que termine el programa.
<i>Verificación Cíclica de Redundancia (CRC).</i>	Técnica de verificación de errores utilizada para asegurar la precisión de la transmisión de código digital a través de un canal de comunicaciones. Los mensajes transmitidos se dividen en longitudes predeterminadas, que usadas como dividendos, son divididas por un divisor fijo. El resto de la división es agregado al mensaje y enviado con el mismo. En el extremo receptor, la computadora recalcula el resto. Si no coincide con el resto transmitido, se detecta un error.
<i>Zonas Mudas.</i>	Zonas o márgenes reservadas, sin barras, formadas únicamente por espacios, a la izquierda antes del caracter inicial y a la derecha luego del caracter final.



BIBLIOGRAFÍA



BIBLIOGRAFÍA

- Amanda Marconi //ADMINISTRACIÓN ORACLE 7 // VOLUMEN 1 // ORACLE // FEBRERO 1993.
- Henry F. Korth y Abraham Silberschatz// FUNDAMENTOS DE LAS BASES DE DATOS. //Mc. Graw-Hill // 1987.
- Janis Furtek Costa // HIGH SPEED NETWORKS // PRENTICE-HALL // 1995.
- Ivan Somerville // INGENIERÍA DE SOFTWARE // ADDISON-WESLEY IBEROAMERICANA // 1993.
- Roger S. Pressman // INGENIERÍA DE SOFTWARE UN ENFOQUE PRACTICO // Mc. GRAW-HILL // 1992.
- Hewlett Packard // INTRODUCCIÓN AL MODELO RELACIONAL // ORACLE IBÉRICA, S.A. // FEBRERO 1993.
- Hewlett Packard // MANUALES DE SERIES 600/800 // HEWLETT PACKARD // 1992.
- Cooperative Server Technology for Transparent Data Sharing // ORACLE7 "SERVER CONCEPTS MANUAL" // ORACLE // Diciembre 1992.
- James Martin // ORGANIZACIÓN DE LAS BASES DE DATOS //Prentice Hall// 1977.
- Sal Ricardi // PC MAGAZINE "BASE DEVELOPMENT SYSTEMS" // Volumen 17 No 1// Enero 1992.
- Hewlett Packard // PL/SQL* // ORACLE IBÉRICA, S.A. // 1995.
- Thomas W. Madron // REDES DE ÁREA LOCAL // GRUPO NORIEGA EDITORES // 1993.
- Andrew S. Tanenbaum // REDES DE ORDENADORES // PRENTICE-HALL HISPANOAMERICA, S.A. // 1994
- Rick Grenhan and Standford Diehl // REVISTA BYTE "DATA BASE AT WORK" // Volumen 17, No. 1 // Enero 1992.
- RED // REVISTA RED. VARIOS ARTÍCULOS // RED S.A. DE C.V. // 1994 - 1995.
- Hewlett Packard // SQL* FORMS // ORACLE IBÉRICA, S.A. // 1995.