



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

“ARAGON”

32
2ej

“DIGITALIZACION DE IMAGENES”

T E S I S

Que para obtener el Título de:

INGENIERO EN COMPUTACION

Presenta:

MARIA DE LOURDES ZUÑIGA LEAL

Asesor: Ing. Víctor R. Velasco Vega

TESIS CON
FALLA DE ORIGEN

San Juan de Aragón, Edo. de Méx. 1993



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está 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.

INDICE

Objetivos	1
Introducción	2
CAPITULO I. Conceptos Básicos	4
Fases	5
Estructura	6
Digitalizadores	7
Scanners	8
Funcionamiento de un scanner de mano (ScanMan)	10
Partes de Scanner	11
Iniciando la exploración	12
Descripción del scanner ScanMan	14
Tamaño de la imagen	16
Resolución	16
Especificaciones técnicas	18
Otros Scanners comerciales	19
CAPITULO II. Formatos de Archivos	21
Unidad de salida (monitor)	21
Texto y Matriz de Video	23
Definiciones de caracteres gráficos	23
Almacenamiento de una imagen en la memoria de la computadora	24
Mapeo Binario	26
Formatos	26
ASCII	27
SDF	27
Estructura de un archivo en Dbase	27
Delimitado	29
DIF, SYLK y WKS	30
Binario	30
Otros	31

CAPITULO III. Técnicas de Compresión	32
Ventajas de las Técnicas de Compresión	32
Tipos de Compresión	34
Tipos de Compresión Física	35
Null Supression	36
Bit Mapping	37
Run Length	41
Half-Byte Packing	44
Diatomic Encoding	46
Statistical Encoding	48
Teoría	48
Código de Huffman	52
Otros	57
Pattern Sustitution	57
Relative Encoding	58
Forms Mode Operation	59
CAPITULO IV. Aplicación de los Sistemas Reales	61
Sistema en Basic (Primer Sistema)	62
Sistema utilizando Bases de Datos (Segundo Sistema)	64
Sistema empleando Compactación (Tercer Sistema)	71
Proceso de Compactación	71
Proceso para mostrar imagen (archivo compactado)	74
Implementación del Sistema	74
Costo del Sistema	80
CAPITULO V. Evolución de los Sistemas	82
Sistemas más recientes	84
Conclusiones	86
Bibliografía	88
Anexo	90

OBJETIVOS:

Actualmente se cuenta con dispositivos que pueden ser utilizados para mejorar la calidad de los sistemas de información por ejemplo los scanners. Dado que en la mayoría de establecimientos donde se tiene alguna computadora, se registran entre otras cosas, los datos correspondientes a sus empleados. Una forma de dar mejor presentación a esta información, es mostrarla haciendo uso de imágenes, pero a su vez minimizar el espacio ocupado por ellas para reducir costos en cuanto a almacenamiento, por todo esto los objetivos de este trabajo se resumen a lo siguiente:

- a) Mediante la digitalización de imágenes resolver o complementar los procesos de información computarizada.
- b) Compactación de imágenes a archivos que ocupan el menor espacio en discos.
- c) Procesamiento de imágenes a través de archivos directos e indexados.
- d) Elaboración de un sistema que permita realizar consultas de mapas, fotografías o cualquier imagen a través de los diferentes métodos de almacenamiento de la información, utilizando archivos compactados.

INTRODUCCION

En los últimos años la digitalización de imágenes ha adquirido gran importancia en el mundo de la computación, debido a que este proceso facilita los trabajos referentes a imágenes, y se puede contar con una mayor exactitud, obteniéndose con mayor calidad si se cuenta con un digitalizador cuya resolución sea muy buena.

La idea en realizar o desarrollar este proyecto, surge porque al contar con un "scanner" manual se pensó que podía servir para obtener un Sistema de Información que proporcionara al usuario una forma de presentar su banco de datos de tal manera que represente la mayor información posible; es decir, que además de tener datos como nombre, dirección y teléfono se pueda anexar una fotografía (imagen) y mostrarse en una pantalla todos los datos a la vez (esto si se trata de un sistema por ejemplo para una empresa que desee visualizar los datos de sus empleados) o también se puede tratar de un Sistema cuyos datos correspondan a instrumentos que fabrique o distribuya una empresa (catálogo), etc.; todo esto debido a la importancia que han adquirido las computadoras en los ámbitos educacional, comercial y de servicio.

Por otra parte, se proporcionan algunos puntos importantes en el formato de archivos para conjuntarlos en un sistema; es decir, cuáles de todos los datos incluidos en el archivo de imagen son útiles para tomarlos y los demás descartarlos, y en algunos casos agregar otros para convertirlos a otro formato como es el binario. Después de esto hacer uso de un método para enlazar el archivo que contiene datos y el archivo que contiene la Imagen.

Además el sistema incluye un programa de compactación del archivo de imagen utilizando un método denominado "Supresión de Nulos o Blancos" y otro de recuperación del mismo para mostrarlo en pantalla, reduciendo con esto el espacio de almacenamiento en disco lo que implica a su vez un ahorro en costos.

Una vez que se ha descrito el por qué de este proyecto y lo que incluye, para explicarlo se tiene este escrito que en sus dos primeros capítulos hace la descripción de una serie de conceptos y puntos importantes relacionados con el tema.

El capítulo tres está dedicado totalmente al desglose de los diversos métodos de compactación, tomando uno de ellos para el desarrollo del sistema explicado en el capítulo cuatro. Y el quinto capítulo que habla acerca de la tendencia de los sistemas en la actualidad y en un futuro.

Finalmente un anexo en el cual se incluyen los programas realizados en este proyecto.

CAPITULO I. Conceptos Básicos

El desarrollo de este trabajo implica el uso constante del concepto Digitalización de Imágenes, definiéndose como un proceso avanzado muy útil para la elaboración, tanto de sistemas como de documentos en los cuales se requiera de imágenes (fotografías, mapas, dibujos, etc.). Esto simplifica mucho el trabajo del usuario ya que sólo basta con utilizar un digitalizador (en este caso un scanner) y el software necesario para obtener la imagen, visualizarlo en pantalla y guardarla en un archivo para poderla manipular según las necesidades.

En este caso el proceso de Digitalización de Imágenes consiste en la utilización de un dispositivo llamado Scanner el cual permite a través de software como PaintShow, PaintBruch, etc. la conversión de una imagen "física" a un archivo o imagen digital¹.

Con este proceso se presentan mejoras en la visualización de información gráfica y como consecuencia la realización de proyectos con mejor presentación y más ilustrativos en un tiempo menor al requerido en otras circunstancias.

Haciendo un paréntesis diré que las imágenes digitalizadas pueden ser perfeccionadas para la percepción del ojo humano a través del Procesamiento Digital de Imágenes.

Este proceso, cuyas fases se enuncian en el siguiente punto, afecta la entrada, realce, análisis estadístico y la salida de tonos de gris y los datos del color

1 Las imágenes digitales tienden a ser muy grandes, son diferentes a los dibujos CAD, donde los datos pueden ser representados en la pantalla como una lista, o en una página de descripción de un lenguaje, en tanto que las imágenes pueden solamente ser descritas por píxeles. En 8 bits por píxel (256 tonos de gris) un 1024 X 1024 píxeles de una imagen forman 1 MByte de almacenamiento. Una imagen de color (24 bits por píxel: 8 bits para cada color de rojo, verde y azul) de un mismo tamaño requiere 3 MB de almacenamiento.

de la imagen, siendo las imágenes generalmente representaciones de objetos en dos dimensiones.

FASES

Las *fases* del Procesamiento Digital de Imágenes que operan en los píxeles (elementos de datos) de los arreglos de datos de una imagen, son:

La fase de *Entrada* que consiste en llevar la imagen dentro del sistema de procesamiento de imágenes. La entrada fuente incluye digitalización de video, un scanner, otros sensores digitales de salida, etc. poniendo la salida de una imagen a residir en un archivo.

La fase de *Procesamiento* asume que la imagen reside dentro de un "buffer". Un área de un buffer es acoplada generalmente, con un procesador de video, pero también se puede tener un área "virtual" de buffer, el cual existe en memoria independientemente de la pantalla. El concepto virtual se hace muy importante cuando se manejan imágenes más grandes que las pantallas estándar.

En esta fase se realizan procesos utilizando técnicas matemáticas como la Transformada de Fourier o bien Matrices para mejorar la apariencia visual de una imagen, siendo ésta más apropiada que la original.

Así, se tiene, que las transformadas pueden ser utilizadas para extraer características desde las imágenes. Por ejemplo, en la transformada de Fourier el término de Valor Medio o "d.c." es proporcional al Brillo Medio de la Imagen, y el término Alta Frecuencia da una indicación de la Amplitud y Orientación del límite dentro de una imagen. Otra aplicación es transformar la imagen en la cual una reducción del ancho de banda es logrado descartando o "engrosando" la magnitud de los coeficientes de la transformada. La reducción en computación es una tercera aplicación; determinando simplemente aquellos coeficientes que son pequeños para ser excluidos de las operaciones de procesamiento filtrándolas, quedando fuera del funcionamiento del procesamiento.

La fase de *Salida* de un procesamiento de imágenes puede ser la misma imagen. La salida puede ser solamente la pantalla de la imagen, salvando la imagen

en un archivo, generando una copia de la imagen en el disco duro (o flexible) o usando una impresora.

Para que estas fases puedan cumplir sus funciones se requiere de cierto equipo que a continuación se enuncia.

ESTRUCTURA

Un sistema típico en el Procesamiento Digital de Imágenes incluye una *computadora* con algunos o todos los siguientes componentes: un área de Buffer, un Monitor, un Digitalizador de Video, una Fuente de Video y un Acelerador de Aplicación.

Un Buffer se compone de una memoria de video de gran rapidez acoplada con un sistema de video. Puede contener más memoria que la que necesita para una pantalla simple; puede usar la memoria adicional para códigos y datos de almacenamiento; para desplegar varias páginas o para retener una imagen grande.

Un Sistema de Video normalmente consiste en buscar las tablas de salida para la asignación de los colores actuales o escala de grises a los valores de la imagen, y los convertidores D/A para convertir la imagen digital en una señal de video analógica para desplegarla.

Las características del monitor deben igualar los requerimientos del buffer. Una señal de televisión estándar (definida como 512 por 480 pixeles entrelazados) requiere un monitor de 15.75 Khz, mientras una señal de 1280 por 1024 pixeles requieren un monitor de 64 Khz. La medida física de un monitor no tiene un comportamiento directo en la señal (de video) de entrada.

La entrada de video puede llegar desde una señal de video analógica por una cámara o a través del Sistema Nacional de Televisión. Estos datos analógicos son convertidos, o digitalizados vía un convertidor A/D y es situado en el Buffer.

El "exploramiento" óptico puede ser usado también para generar imágenes. En este caso un arreglo de sensores, generalmente de 200, 300 o 400 puntos por pulgada, generan salidas digitales. El papel se mueve encima del sensor,

se mueve sobre la pieza de papel fija; o ambos, el sensor y el papel pueden ser estacionarios mientras los sensores ópticos se muevan. Estas líneas "exploradas" son reconstruidas como la imagen en el buffer.

Finalmente, un Acelerador de Aplicación es una "tarjeta" o subsistema que tiene capacidad de Procesamiento, independientemente del CPU.

DIGITALIZADORES

Retomando nuevamente el tema, los digitalizadores son periféricos gráficos de entrada que han aparecido gracias a los avances tecnológicos registrados en el campo del diseño y la fabricación más conocido como CAD/CAM, debido a los cuales han aumentado las aplicaciones de las computadoras en la ingeniería, en el control de procesos, en la fabricación, el diseño gráfico, etc.

Su utilización para almacenar planos en arquitectura, por ejemplo, permite guardarlos como están y de una manera muy rápida. Una vez memorizado el plano, dibujo, pieza, etc., pueden ampliarse, reducirse, verlo desde diferentes ángulos, en general modificarlos según se requiera o según lo permita el software utilizado.

Un digitalizador convierte una imagen en una representación numérica apropiada para ser la entrada de una computadora, aquellos que se usan son: Microdensímetros, Scanners, Disectores de Imágenes, Cámaras de video y Fotosensitivos de estado sólido. Los primeros dos dispositivos requieren que la imagen a ser digitalizada este en forma de una transparencia (negativo) o una fotografía; los disectores de imagen, cámaras de video y arreglos de estado sólido, pueden aceptar imágenes grabadas de este modo, pero tienen la ventaja adicional de ser capaces de digitalizar imágenes naturales que tienen suficiente intensidad de luz para excitar el detector.

Cabe señalar que cuando los digitalizadores nacieron estaban reservados a grandes instalaciones de computadoras debido a su elevado costo, pero en la actualidad ya existen digitalizadores adaptables a proyectos más reducidos.

La conexión con las computadoras, se efectúa a través de una interface RS232-C o IEEE488, HPIB, GPIB, ect., y la introducción de los datos en la

computadora normalmente se realiza mediante la conexión al digitalizador de un lápiz óptico.

El funcionamiento de un digitalizador puede describirse de la siguiente manera:

Quando el lápiz óptico toca cualquier punto de la tableta, el digitalizador sabe qué punto ha tocado y envía la información pertinente a la computadora, donde se almacenará o visualizará en cualquier momento. Supongamos que se tiene un dibujo para almacenar en la computadora, esta operación será muy fácil y rápida mediante el digitalizador; se deberán seguir simplemente las líneas del dibujo con el lápiz óptico del digitalizador, los datos relativos a todos los puntos que el lápiz óptico va siguiendo se irán transmitiendo a la computadora y ésta los podrá almacenar en un fichero, de tal modo que podrá visualizarlos, modificarlos y moverlos o almacenarlos cuantas veces se quiera. Normalmente, estos datos representan las coordenadas relativas a cada punto.

En microdensímetros la transparencia o fotografía es montada en un plano o enrollada en un tambor. El "exploramiento" es realizado enfocando un rayo de luz en la imagen y trasladando el plano o rotando el rodillo en relación al rayo (haz). En el caso de transparencias el rayo pasa a través del film; en tanto que en fotografías es reflejado desde la superficie de la imagen, en ambos casos el rayo es enfocado a un fotodetector y el nivel de gris en algún punto en la imagen es grabado por el detector basado en la intensidad del rayo.

Una imagen digital es obtenida permitiendo solamente valores discretos de intensidad y posición en la salida. Aunque los microdensímetros son dispositivos lentos, son capaces de un alto grado de exactitud en la posición, debido especialmente al continuo movimiento de translación mecánico usado en el proceso de digitalización.

SCANNERS

Los Scanners son lectores ópticos de imágenes, también operan con el principio de enfocamiento y transmiten o reflejan el rayo en un fotodetector. En este

caso, sin embargo, la imagen es estacionaria y la fuente de luz es un Tubo de Rayos Catódicos (CRT) en el cual un rayo de electrones, desviados por electromagnetismo, toca una superficie de fósforo fluorescente. El rayo de este modo produce una marca (mancha) de luz que se mueve en un patrón de "exploramiento" en la imagen del tubo. El hecho de que el rayo sea movido electrónicamente permite alta velocidad en este proceso.

Almacenar imágenes dentro de la computadora no es un procedimiento sencillo, la forma fácil y rápida de hacerlo es utilizando un periférico de este tipo. Los scanners leen imágenes en dos dimensiones y las convierten en bits. En el mundo del diseño, en el de la seguridad, en las finanzas, en las artes gráficas, se encuentran aplicaciones claras de esta tecnología.

Un scanner actúa como una impresora para imágenes gráficas, pero con un procedimiento invertido. Así, del mismo modo que la impresora, a partir de la introducción de unos datos, genera una imagen, un scanner genera unos datos a partir de una imagen. En realidad, para un scanner una imagen es una serie de paquetes cerrados de puntos.

La mayoría de scanners están diseñados para poder usarse en conjunción con minicomputadoras o con estaciones de trabajo inteligentes en campos como artes gráficas, ingeniería, etc. Actualmente se desarrolla esta tecnología para poder usar los scanners con las computadoras personales.

Un scanner no es estrictamente un lector óptico de caracteres. Los lectores ópticos de caracteres convierten las imágenes de texto analizadas en código de caracteres, mientras que los scanners aceptan imágenes de texto y gráficos en cualquier forma y tamaño. Un lector óptico de caracteres puede reconocer 2000 caracteres de una página convirtiéndolos en un total de 15000 bits aproximadamente. Si se comparan estas cifras con las que ofrece un scanner², comprenderemos la

2 Una imagen leída por un scanner ocupa más de 30 Kbytes, dependiendo de la resolución así como de su tamaño (para el caso del scanner utilizado en este proyecto).

diferencia que existe entre ellos. La cantidad de memoria que una computadora necesita para almacenar la imagen leída por un scanner es igual a la que esta misma computadora utilizaría para almacenar 400 páginas de texto leídas por un lector óptico de caracteres.

Funcionamiento de un Scanner de mano (ScanMan)

Un scanner de mano no es más que un dispositivo con un sensor que detecta la suma de luz reflejada o transmitida por un punto dado del área de trabajo. Un scanner permite aumentar la capacidad gráfica de nuestra computadora, con lo que se pueden tomar ilustraciones y gráficas de libros, revistas, periódicos, etc. que pueden ser usadas posteriormente en la presentación de proyectos. Como se sabe, los colores negros absorben todos los colores de luz. Cuanto más luz absorbe un objeto, lo negro (oscuro) aparece en la vista, cuanto más luz es reflejada por un objeto, aparecerá más blanco.

En el mundo real, prácticamente nada absorbe toda la luz que emiten, ni tampoco los espejos reflejan el 100% de la luz que reciben. Las cosas que no absorben grises en algún lugar, parte de la luz es absorbida y parte reflejada.

Los scanners de mano, parecidos a nuestros ojos, asumen la ventaja de los diferentes caminos, en los cuales los objetos absorben o reflejan la luz. Si la luz recibida por el sensor no excede el valor (umbral) predeterminado, el pixel es determinado como negro. A los valores mayores que el umbral le son asignados un número representativo de blanco, o en el caso de scanners con tonos de gris un número correspondiente a algún tono (nivel).

Estos scanners permiten el ajuste del umbral, hacia arriba o hacia abajo. Teniéndose este control se puede hacer que determinada información sea registrada o ignorada. El software del scanner permite alterar los valores asignados para especificar los niveles de luz, así como, ajustar lo que se denomina mapa de gris (gray map).

Los scanners de mano de color trabajan exactamente de la misma manera, excepto que separan los valores luz/obscuridad que son registrados para cada uno de los tres colores primarios.

Todos estos scanners capturan la información en píxeles, el sensor se mueve por encima del original hasta que se completa la exploración del área que será recorrida. Este proceso es muy sencillo, así que el original será explorado en pocos segundos.

Partes del Scanner (ScanMan)

Los mecanismos de este proceso varían, dependiendo del tipo de scanner. Sin embargo, en términos generales se puede decir que todos los scanners usan una fuente de luz, algunas formas de movimiento del sensor (generalmente, el movimiento de la mano) sobre la superficie del área de trabajo y circuitería para convertir la información capturada en forma digital.

Con los scanners de este tipo, el área a explorar es colocada usualmente en una superficie sólida, tal como un escritorio; el scanner es colocado hacia abajo encima del original. La luz es producida en el scanner desde una fuente (Figura 1) como un *diodo emisor de luz (LED)*. El color de la luz es determinado por el tipo de LED, que generalmente es roja, amarilla o verde.

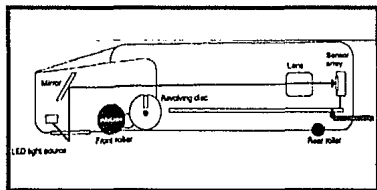


Figura 1: Corte transversal de un scanner

Los scanners de color necesitan una fuente de luz blanca, de tal manera que usan tubos fluorescentes muy pequeños. Mientras estos son más eficientes que las lámparas incandescentes, ellos usan más energía que un LED, así que los scanners de color son generalmente más grandes y exigen demasiada energía de la PC.

El *sensor de exploración* es localizado cerca de la superficie inferior del panel de plástico. Un conjunto de *rodillos* en la parte de abajo del scanner elimina casi toda la fricción que resultaría si simplemente se "arrastrara" el scanner a través del original.

En algunos scanners, uno de los rodillos también sirve para llamar al software cuando se mueve el scanner, haciéndolo fácil para relacionar el movimiento de la mano con los cambios que el sensor recibe de la imagen.

Iniciando la exploración

El proceso de exploración se inicia presionando el botón (scan button) del lado izquierdo del scanner. Algunos scanners también permiten el inicio, seleccionándolo desde el software, usando un comando de línea, menú o a través del mouse. El botón se mantiene presionado hasta terminar el proceso.

Cuando la exploración inicia, la fuente de luz, generalmente un arreglo de LED's, ilumina el área del objeto a través de una ventana, con luz verde, amarilla, roja o blanca³.

La luz es reflejada desde una porción del objeto hacia arriba a un espejo colocado en un ángulo en la cabeza (Figura 2). Esta luz es enfocada por lentes que se encuentran en la parte posterior del scanner encima de un arreglo de sensores, el cual es un fotodiodo o dispositivo de acoplamiento de carga⁴ que traslada el nivel de luz a una corriente eléctrica. Cuanto más luz es reflejada, mayor es el voltaje.

3 Como la luz del Scanner es roja no toma el color rojo (ni sus variaciones), por lo cual se debe evitar explorar imágenes que hayan sido dibujadas con él o bien que lo contengan. Dicho color es interpretado como blanco y no puede verse en la imagen explorada.

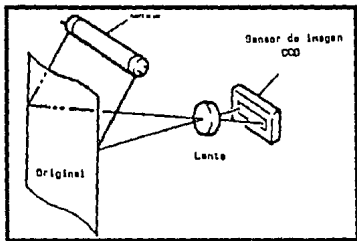


Figura 2: Diagrama de exploración de un documento.

Es decir, una vez que los lentes han concentrado la luz reflejada pasa al sensor que es un arreglo lineal; un scanner de mano captura una imagen, una línea a la vez, usando una serie de sensores diminutos ordenados en un renglón. Pueden ser 100 a 400 de estos por pulgada ó 400 a 1600 sensores por cada 4 pulgadas de un scanner de mano⁵. Produciendo típicamente resoluciones de 100 a 400 puntos por pulgada (DPI).

Un scanner de mano también captura un número equivalente de líneas por pulgada como se hace dibujando sobre un área, así una pulgada cuadrada de imagen tendrá 100 x 100 a 400 x 400 píxeles.

El dato es capturado en forma analógica, esto es, un rango continuo de voltajes (valores). Estos voltajes fijos necesitan ser convertidos a una forma digital para que la computadora pueda procesarlos.

- 4 Este dispositivo (CCD) es un sensor de imágenes de estado sólido que maneja poca corriente eléctrica y es de un tamaño muy reducido, y por lo consiguiente de poco peso.
- 5 Para el caso del ScanMan de Logitech, si mide 4.14"; haciendo una regla de tres, se tiene que utiliza 414 sensores para una resolución de 100 DPI, 828 para 200, 1242 para 300 y 1656 para 400 DPI.

Algunos scanners de mano, procesan la imagen antes de la conversión analógica-digital. Un chip puede ser incluido en la "tarjeta" para suministrar una forma de comparación llamada "corrección gama", la cual toma un promedio de la mayor sensibilidad del ojo para tonos oscuros. Otros scanners manejarán esta corrección después de que la información haya sido convertida en forma digital.

Esto es realizado a través de un dispositivo llamado convertidor Analógico-Digital, en este caso el CCD que durante la exploración, recibe los cambios de la intensidad de luz dando a la salida variaciones de voltaje en forma de pulsos digitales. En un scanner con escala de gris completa, el rango analógico de voltajes que es capaz de registrar es convertido a 256 valores digitales diferentes. Esto provee suficientes valores diferentes que algún tono de gris puede ser representado con muy buena exactitud.

Algunos otros scanners tienen limitado su rango de voltajes, de tal manera que sus convertidores concentran toda la información recibida dentro de dos valores: negro y blanco.

Esto es, si el pixel actual excede del nivel determinado, es almacenado como blanco, de otra manera como negro. Este "umbral" puede ser ajustado frecuentemente.

Finalmente, cuando ya se tiene la información en forma digital es enviada a la memoria de video para ser desplegada en la pantalla y poderla manipular.

Descripción del scanner ScanMan (Figura 3):

1. Scanning Window:

Es una ventana que permite la "exploración". Cuando el Scanner lee, la ventana se ilumina.

2. Scan Buttons:

Debe ser presionado durante la operación.

3. Switch Dither/Line Art

Modo Dither (Área punteada) es para una imagen que contiene diferentes colores, tonos de gris o sombreados. Si se deja el scanner en esta posición, la imagen será desplegada en pantalla o impresa en varias tonalidades.

Modo Line Art (Área negra) es para 'explorar' texto y dibujos (ilustraciones, diagramas, tablas, etc.), es decir, alguna imagen que no contenga sombreados solo blanco y negro.

4. Switch de Resolución

Permite seleccionar entre cuatro resoluciones diferentes: 100, 200, 300 y 400 puntos por pulgada (DPI)

El ajuste permite poner la resolución del scanner para igualarla con la resolución del dispositivo de salida (en este caso el monitor de la computadora), de esta manera dar mejor calidad a las imágenes.

5. Control de Contraste:

El dual o perilla de Contraste determina (como su nombre lo indica) el contraste entre las áreas claras y oscuras de la imagen. Esta perilla tiene una

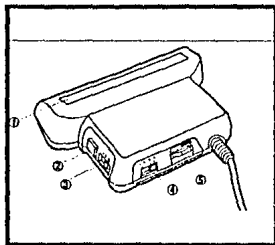


Figura 3: Diagrama del Scanner
ScanMan

"ranura" que marca el punto medio, normalmente es movida del centro hacia la izquierda (menor contraste); pero si se quiere un contraste elevado entonces la marca es girada del centro a la derecha.

Tamaño de la imagen

El ScanMan cuenta con el software WinScan el cual posibilita "explorar", salvar e imprimir imágenes dentro del ambiente WINDOWS, permitiendo un ancho de 4.14" (10.5 cm) como máximo de la figura.

El otro software es el PaintShow Plus; este proporciona una variedad de herramientas gráficas para modificar y por tanto mejorar las imágenes 'exploradas' (digitalizadas) y además cuenta con utilerías para convertir imágenes a los formatos de otros paquetes. El ancho máximo que permite este paquete es de 4.10" (10.4 cm).

El default de largo para ambos es de 6" (15 cm).

El largo máximo que se puede especificar con este scanner de mano depende de varios factores: el ancho especificado, el modo de video, la resolución del scanner y la memoria RAM disponible. (La memoria RAM disponible sin usar DOS, drivers, programas residentes o alguna otra aplicación). Por ejemplo, para una imagen de 4 X 1 (pulgadas) se necesitan 20 K de RAM disponible

Resolución

Dos de las características fundamentales de todo digitalizador son la Resolución y la Precisión. La Resolución se refiere al máximo número de puntos que el digitalizador puede reconocer por pulgada (2.54 cm), mientras que la Precisión indica la exactitud con que pueden reconocerse los puntos.

En otras palabras, la resolución es una proporción de Unidades-Imagen (puntos de la pantalla o pixeles) por área digitalizada (pulgadas, centímetros). Por ejemplo, 300 puntos por pulgada (DPI).

Por ejemplo, un sistema con 640 de RAM y configuración estándar (mouse, DOS, network) obtendrá una imagen con un largo máximo de 10.5" en resolución de 200 DPI.

DPI	LARGO MAXIMO	RAM Requerida
100	41.5	640
200	10.5	640
300	4.6	640
400	2.6	640

Figura 4: Tabla de comparación entre las diferentes resoluciones utilizadas.

Cada Unidad-Imagen usa un bit de Memoria. En alta resolución, más puntos son generados y más memoria es utilizada. Como resultado de esto, al incrementarse la resolución, la longitud (largo) máxima 'explorada' decrece.

Se observa entonces, que en alta resolución al ser más puntos para generar una imagen, ésta se alarga en la pantalla.

La combinación de Scanner (ScanMan) y el software PaintShow Plus permite 'explorar' una imagen e introducirla directamente en donde pueden hacerse cambios, trazos, redibujar, ponerles sombras, color, etc., o bien imprimirla.

Cuando se salva una imagen, ésta es guardada con el formato TIFF (Tagged Image File Format). Este archivo puede ser usado en otras aplicaciones que acepten dicho formato o bien pueden ser convertidos a otros con las utilerías existentes, como ya se mencionó anteriormente.

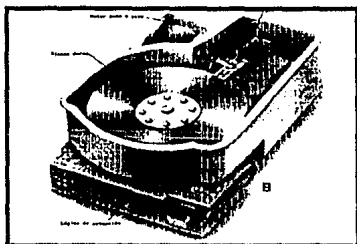


Figura 5: Imagen digitalizada a 200 dpi

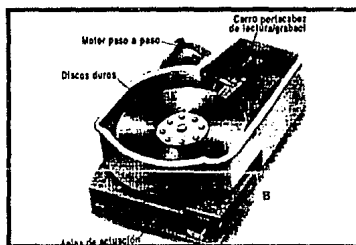


Figura 6: Imagen digitalizada a 300 dpi

Especificaciones Técnicas

Dimensiones	134 mm x 91 mm x 32 mm (5.3 x 3.6 x 1.3 pulgadas)
Peso	300 gramos (0.66 lb)
Ancho	105 mm (4.14 pulgadas)
Lectura por seg.	87 mm/s (100 DPI) 44 mm/s (200 DPI) 29 mm/s (300 DPI) 22 mm/s (400 DPI)

Píxeles por línea	420 píxeles/línea (100 DPI) 840 píxeles/línea (200 DPI) 1260 píxeles/línea (300 DPI) 1680 píxeles/línea (400 DPI)
Longitud del cable	1.8 mts. (7 pulgadas)
Conector	8 pin mini DIN
Consumo de corriente	400 mA máximo (a 12 V DC +/- 10%)
Fuente de luz	LED rojo
Modos	1 blanco y negro y 3 en tonos de gris
Niveles	32 niveles
Canal DMA	1, 2 o 3
Direcciones I/O	280h, 2A0h, 330h, 340h
IRQ	ninguna, 2, 3, 4, 5 o 7

OTROS SCANNERS COMERCIALES

Además de los Scanners manuales existen scanners de página completa a color como es el caso del Scanner HP ScanJet IIc, el cual es un digitalizador tipo Cama plana (Flatbed) y cuyas características se enuncian a continuación:

Color

- * *Reconocimiento de color de 24 bits, hasta 16 millones de colores.*
- * *La digitalización automática que asegura un registro superior y la digitalización rápida del color.*
- * *Compensa la forma en que los colores son interpretados en cada paso del proceso.*
- * *Identifica los colores dominantes en digitalizaciones de piezas en color y convierte cada pixel con uno de estos colores específicos.*

Escala de Grises

- Resolución óptica de 400 puntos por pulgada (DPI)
- 256 niveles de tonalidades de grises que aseguran excelentes imágenes en blanco y negro.

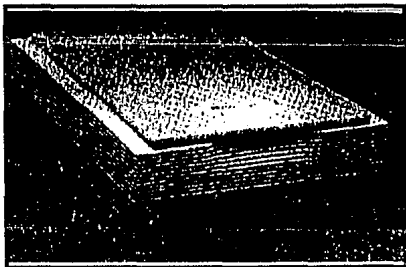


Figura 7: Scanner de cama plana

Reconocimiento óptico de caracteres

- La resolución óptica de 400 DPI tiene la capacidad de leer tipografía hasta de 6 puntos con mayor precisión que la que ofrecen menores resoluciones.

CAPITULO II. Formatos de Archivos

Uno de los problemas que se tuvieron para la utilización o conjunción de los archivos del sistema se presenta al no tener las mismas características o formato.

En este capítulo se incluye la descripción de formatos de archivos más comunmente usados en procesadores de texto, base de datos e imágenes con el fin de compararlos.

Pero antes de llegar a este punto es importante trabajar alrededor de los elementos que posibilitarán su desarrollo.

UNIDAD DE SALIDA (MONITOR)

Las unidades de monitor de pantalla (display) son dispositivos de salida en la estructura de una computadora, se usan para visualizar datos o información dados por ella a través de dispositivos de entrada como un teclado o un digitalizador.

Estas pantallas pueden ser:

- Tubo de rayos catódicos (CRT): para visualizar información dada por una computadora en este tipo de pantalla, es necesario que la computadora tenga capacidad para memorizar y generar imágenes en su propia memoria, para posteriormente reproducirlas en la pantalla.
- Pantalla de cristal líquido (LCD): el principio de este tipo de pantalla es muy parecido a un interruptor. Es decir, si se aplica voltaje, se bloquea el paso de la luz y, si no es aplicado, la luz pasa permitiendo la visualización de la información que se va a representar.

Estos dispositivos permiten visualizar caracteres y gráficos cuyos bloques de memoria de acceso aleatorio (RAM) y de sólo lectura (ROM) se distribuyen con el fin de mantener los datos necesarios para describir una imagen visual.

El potencial de una computadora de visualizar texto se describe en términos de filas y columnas (25 X 40 u 80). Esta división no es estándar, ya que cada computadora se adapta a las características de la pantalla que utiliza. En la actualidad, el estándar de las pantallas es de 14 pulgadas.

Cuando se dibujan imágenes por medio de mapas binarios, el tamaño de un punto de exhibición llamado pixel es la unidad descriptiva, es decir, cada punto de la imagen es considerado individualmente.

Al número de pixeles que puede visualizar el monitor tanto en horizontal como en vertical se le conoce como *resolución* gráfica. Las resoluciones gráficas más comunes son: 320 X 200, 640 X 350 o 640 X 480 (en la actualidad ya se empieza a trabajar con 1024 X 720). Estos valores indican el número de pixeles o puntos por pantalla. El primer valor hace referencia a puntos horizontales y el segundo a puntos verticales en que se divide el monitor.

Así, si se supone que un monitor tiene en modo gráfico una resolución de 640 X 480, tendremos entonces un total de 307,200 puntos en la pantalla.

Los paquetes de "pintura" almacenan las imágenes como una lista de pixeles (imagen mapeada en bits) como es el caso de PaintShow, PaintBrush, entre otros; los paquetes de dibujo almacenan la imagen como una lista de formas, tales como rectas, círculos y puntos (AutoCad). Si se quiere digitalizar fotografías u otras imágenes y después editarlos, se necesitará un paquete de "pintura". Un explorador (lector óptico) registra las imágenes en un formato con mapeo en bits porque las lee pixel por pixel.

Por otra parte, si va a trabajar con dibujos técnicos o diagramas con formas geométricas complicadas es más conveniente un paquete de dibujo. Las formas se convierten al formato del periférico de salida con la mayor resolución posible, de tal modo que si se tiene una impresora láser de 300 DPI, los dientes de las líneas diagonales estarán separados apenas 1/300 de pulgada. Una línea similar creada con un programa de pintura aparecería con las indentaciones más pronunciadas porque la definición depende del número de pixeles con que se creó originalmente la línea.

Las pantallas de computadora suelen tener una resolución de menos de 80 pixeles por pulgada lineal. Si el dibujo en pantalla se hace del tamaño que se desea en el producto final, los dientes serán cuatro veces más que la imagen correspondiente creada con un paquete de dibujo. Para resolver esto, los dibujos tendrán que hacerse cuatro veces más grandes en la pantalla que en el producto final. Sin embargo, la edición de las imágenes creadas con paquetes de pintura se dificulta conforme su tamaño aumenta.

TEXTO Y MATRIZ DE VIDEO

El crear imágenes, dibujarlas y combinarlas de manera interactiva, así como superponer imágenes y crear patrones, se encuentran disponibles mediante el uso de dos técnicas : Caracteres Gráficos y Mapas Binarios.

El caracter gráfico es el método común para exhibir texto (modo de texto o alfanumérico) y gráficos muy sencillos en tanto que el mapa binario es para gráficos complicados o imágenes.

El texto y las imágenes tienen representaciones internas diferentes en la memoria de las computadoras. Parecido a una página impresa, el tipo predefinido se usa para texto y un arreglo de puntos para imágenes. Para la impresión de texto, se arregla la memoria de la pantalla, de modo que las localizaciones de memoria consecutivas representan espacios seguidos donde se pueden colocar una parte del tipo.

Cada dirección de memoria recibe el nombre de *celda*. La memoria de pantalla compuesta por celdas se denomina *matriz de video*. El número que se almacena en cada celda le dice al sistema de graficación qué caracter exhibir en la pantalla de video en la posición correspondiente. Los caracteres disponibles incluyen letras, números, símbolos y simples formas de graficación del tamaño de un caracter.

DEFINICIONES DE CARACTERES GRÁFICOS

Las descripciones de los caracteres se almacenan en la memoria ROM. Generalmente se usan 8 bits consecutivos para definir un caracter, como se muestra

en la figura 8, en la que cada bit indica al sistema de gráficos si el pixel correspondiente se enciende o se apaga.

El sistema de gráficos usa un número caracter como índice para traer la definición correspondiente desde ROM y exhibirla en la pantalla.

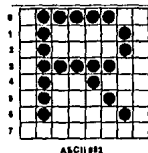


Figura 8

Las definiciones ASCII convencionales son de 0 a 127, los caracteres de 128 a 255 por lo general son gráficos o caracteres invertidos. Al respecto es interesante el caso del cursor que se muestra por medio del cambio del caracter donde se localiza la gráfica invertida y de nuevo al contrario. Esto se hace al asegurar el bit 7 del número del caracter y luego indicar un caracter en ROM que sea 128 veces mayor que el valor ASCII.

En el modo de caracteres gráficos, el circuito que lo genera tiene el conocimiento interno para organizar los ocho bits de una definición de caracter; en el área del tamaño del caracter de la pantalla correspondiente a la celda de la matriz que contiene el número del caracter. Así, un 69 en localización de memoria 1445 de la matriz, coloca una E en la tercera fila de la pantalla, en la columna 21 ($1024 + 5 \cdot 40 + 21 = 1445$).

ALMACENAMIENTO DE UNA IMAGEN EN LA MEMORIA DE LA COMPUTADORA

Por lo general un bloque de memoria de 1 a 12 K se distribuye para datos en pantalla, cada dirección de memoria corresponde a una localización particular en la pantalla, si se cambia el valor de alguna dirección de memoria específica, se modifica lo que aparece en pantalla en la ubicación correspondiente. La organización de la memoria para representar una pantalla de dos dimensiones se denomina *Mapa de Memoria* y su contenido, el cual representa una imagen en pantalla, recibe el nombre de Imagen de la Pantalla en Memoria.

La técnica usada para la memorización de los pixeles en la memoria de la computadora se conoce como bit-mapping (Figura 9). Cada uno de los puntos elementales o pixeles corresponde en la memoria a un bit cuyo estado puede ser 0 o

1. Dicho estado determina respectivamente si el pixel está encendido o apagado. Este concepto se aplica en las pantallas monocromáticas.

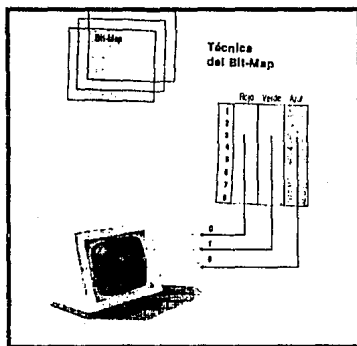


Figura 9: Bit Map

Si con un bit se puede determinar dos estados para cada pixel, con dos bits se definen cuatro posibles condiciones. Estas cuatro condiciones pueden corresponder a una escala de 4 tonos en una pantalla monocromática o, en su lugar, a cuatro colores diferentes en una pantalla de color.

Si se amplía el mapa de memoria o bit map y se asignan tres bits para cada pixel, se obtendrán ocho colores distintos o una escala de ocho tonos de gris en una pantalla monocromática.

En las combinaciones de bits memorizados no hay diferencia entre escala de gris o colores; es la pantalla la que da el color.

Supuesta una pantalla de resolución 640 X 400 (monocromática), se necesitan 32 Kbytes de memoria interna de la computadora para que cada pixel tenga

una posición en RAM. Esta capacidad de memoria se debe aumentar si, con la misma resolución, se desean obtener ocho colores o tonos de gris.

MAPEO BINARIO

Cuando se generan gráficas por medio del mapeo binario, cada pixel en la pantalla se exhibe en forma directa desde la RAM. En un mapa binario de un solo color, cada bit de memoria corresponde a un pixel de pantalla. Para una resolución de pantalla de 320 X 200 pixeles = 64 000 pixeles se requiere una RAM de cerca de 8 K ($8 \text{ K} \cdot 1024 \text{ Bytes} - \text{K} \cdot 8 \text{ bits/byte} = 65\,536 \text{ bits}$) con el objeto de mantener la imagen de memoria. El mapeo de la imagen de memoria a la de pantalla depende de la computadora.

Todo dibujo tiene como base el encendido o apagado de cualquier pixel que se desee; por lo tanto, los comandos de lenguajes de alto nivel deben tener como base algún código de máquina que traduzca las coordenadas pixel X-Y en un valor para almacenarlo en la RAM de mapa binario

Los programas de procesamiento de textos tienen la capacidad de administrar formatos de texto con organización de líneas, colocación de márgenes y adición de efectos especiales de impresión.

Después de este resumen nos adentraremos ahora a los formatos de archivo.

FORMATOS

- ASCII
 - Formato de Datos de Sistema (SDF)
 - Delimitado
- Formato de Intercambio de Documentación (DIF)
- Formato de Enlace Simbólico (SYLK)
- Formato de Hoja de Cálculo Lotus (WKS)
- Binario
- Otros

ASCII

El término "Formato ASCII" (American Standard Code for Information Interchange) se refiere a archivos que están compuestos por caracteres y espacios que no necesariamente están dispuestos en un orden determinado. Este formato es un método internacional de representación de información en las computadoras. Los archivos de texto creados por procesadores de palabras se guardan como ASCII.

SDF

Los archivos que usan el SDF guardan cada registro en una lista, estando separados unos de otros por retornos de carro. Además, los registros en este tipo de archivos mantienen una longitud fija para cada campo, no importando el tamaño real de la información de dicho campo, de tal manera que la información que exceda el tamaño será truncada. Los archivos en formato SDF a veces se denominan Archivos Planos.

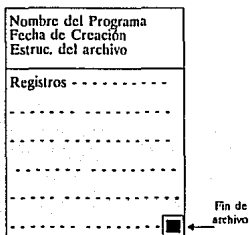
Este formato es el más usado por el Dbase III Plus (Standard Data Format, formato de datos normalizado)⁶. En resumen, un archivo SDF no marca los campos de ninguna forma, sino que los encadena juntos, terminando cada registro con un código de retorno de carro, dando como resultado que cada registro ocupe una línea de texto y por tanto el archivo contenga una línea por cada registro de la base de datos.

Estructura de un Archivo de Dbase

Un archivo de Dbase III Plus consta de tres partes principales: encabezado del archivo, área de datos y marca de fin de archivo.

En forma de diagrama se tiene:

6 Se puede crear un archivo en dbase tipo texto y de esta forma hacerlo SDF.



La primera parte contiene la información acerca de la estructura del mismo; es decir, aquella que se introdujo durante el proceso de creación del archivo (nombres de los campos, longitudes y tipos), pero no está almacenada en la forma en que se introdujo. En lugar de eso, Dbase III Plus cambia la información y crea el encabezado del archivo que sólo este programa y otros específicamente diseñados para trabajar con Dbase III Plus pueden entender.

Después del encabezado están los datos que son simplemente una tira de caracteres. Dbase III Plus determina los campos y registros usando la información almacenada en el área de encabezado (la estructura) para contar caracteres con objeto de determinar dónde terminan los campos y registros.

Finalmente, el último carácter del archivo es un carácter especial que indica el fin de archivo.

Físicamente se tendría:

```

03 5B 06 03 54 00 00 00 C1 00 5B 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4E 4F 4D 42 52 45 00 00 00 00 00 43 0B 00 9E 4C
1E 00 00 00 01. . .
    
```


donde:

Los bytes 2o., 3o. y 4o. indican la fecha de la creación o en su defecto la de la última actualización de la base de datos.

5o. byte: No. de registros de la base.

11o. byte: longitud del registro.

A partir del byte 33 se indican nombres, tipos y longitudes de los campos; la separación entre ellos es el byte 01 una vez que se terminan de definir los campos⁷ se encuentra la lista de los datos.

DELIMITADO

El formato DELIMITED (Delimitado) se llama así porque cada campo está marcado; es decir, están compuestos de registros con los campos encerrados entre comillas y separados por comas. Cada registro ocupa una línea, de tal forma que cada uno de ellos va seguido de un salto de carro.

Este formato es diferente del SDF (como puede observarse) en tres aspectos:

1. En el formato DELIMITED, todos los campos están separados por una coma.
2. Son eliminados los espacios en blanco al final de cada campo -si los hay-, teniendo como consecuencia un archivo más pequeño porque no transfiere los blancos innecesarios. Esto es importante en términos de procesamiento de textos porque el usuario no necesita insertar esos espacios en blanco.
3. Todos los campos de tipo carácter están encerrados entre comillas.

Ejem. "JUAN PEREZ";VOLCANES NO. 27,557-02-38

7 Se cuenta con 32 bytes como máximo para definir un campo.

La desventaja es que la lectura no es muy fácil de realizar y su valor se basa en el hecho de que puede usarse para rellenar cartas circulares creadas por varios programas de procesamiento de palabras, como WordStar o Word.

DIF, SYLK y WKS

Además de textos en ASCII, muchos programas pueden transferir datos usando uno de los tres formatos de archivos más usuales: SYLK, DIF y WKS. El formato SYLK (Enlace Simbólico) desarrollado por Microsoft Corporation, es usado a menudo por los productos de Microsoft para el intercambio de archivos. Los programas de Microsoft Chart (gráficos), Multiplan (Hoja de cálculo) y File (gestor de base de datos) pueden trabajar con archivos escritos en el formato SYLK. El formato DIF se usa en muchos programas, incluyendo Visicalc y el formato WKS utilizados por hojas de cálculo.

BINARIO

Cuando se almacena información en un archivo de computadora, puede ser almacenada de acuerdo con uno de dos sistemas de codificación. Uno de los sistemas de codificación es el ASCII y el otro es el sistema de codificación binario.

La diferencia primaria entre los dos sistemas es que la codificación ASCII está diseñada para permitir el intercambio de datos entre programas diferentes. La codificación binaria se usa para almacenar datos en un formato especial, usualmente exclusivo del programa con el que se trabaje. El almacenamiento binario es usualmente más eficiente que el almacenamiento ASCII. Un archivo de base de datos Dbase III Plus contienen información codificada en binario, así como en ASCII. El encabezado del archivo contiene códigos binarios específicos de Dbase III Plus. Los datos están almacenados en formato de código ASCII y pueden intercambiarse con otros programas que también reconozcan el sistema ASCII.

OTROS

De otro tipo de paquetería que genera sus propios archivos o archivos de datos.

Por ejemplo los archivos en formato TIFF (abreviación de Tagged Image Format File) que tienen la extensión TIF son comunmente usados. Este formato es llamado así porque cada archivo incluye una colección de información, llamadas etiquetas, las cuales describen el tipo de archivo; es decir, información acerca de la resolución, número de bits usados por pixel, entre otras.

CAPITULO III. Técnicas de Compresión

En la mayoría de las aplicaciones, sólo ciertas porciones del archivo contienen las imágenes que por ser significativas deben almacenarse, mientras que las demás pueden despreciarse. Por ejemplo, la tarjeta de identificación de un empleado de una compañía consta de una fotografía que ocupa unos 4 cm², una huella digital y una firma que podrían, ambas, ocupar unos 4 cm², respectivamente. Si la tarjeta en total ocupa 50 cm² el ahorro de almacenar sólo las imágenes significativas (fotografía, huella y firma) es del 76 por ciento.

VENTAJAS DE LAS TECNICAS DE COMPRESION

Un sistema de reducción puede ser la aplicación de técnicas de compresión de máscaras de bits. En la mayoría de imágenes hay zonas en que el color y la distribución de los puntos se repiten. Entonces, en lugar de almacenar todos los puntos de la imagen tal y como se presentan, con este nuevo sistema se introduce un factor de repetición que hace que una determinada porción de imagen se pueda representar con pocos puntos, aunque cabe aclarar que estas técnicas de compresión son poco efectivas con imágenes de muchos tonos.

Al darse un desarrollo en la transferencia de información a gran escala y el almacenamiento de información masiva; se da también un aumento en cuanto a problemas fundamentalmente de carácter económico, ya que al requerir mayor área de almacenamiento los gastos son mayores porque es más costoso un dispositivo con más capacidad y porque su tiempo de proceso es mayor.

Uno de esos problemas es el llamado "Run-Away Data Base". Es decir, las bases de datos que tienen información almacenada de una manera organizada, al ser muy grandes, el proceso de recuperación es muy largo, requiriendo de más capacidad de almacenamiento (diskettes, cintas); acompañado de esto, también se da un incremento en el número de usuarios y la duración de permanencia de los empleados.

Así, entonces, cuando se trabaja con base de datos, se trata de hacer un buen diseño para implantarla, de tal manera que exista la máxima reducción posible; esto es, evitar redundancia de campos. Por ejemplo, si se diseña una base de datos que tiene como campos: el nombre del alumno, carrera, edad y domicilio; lo que se haría a simple vista sería capturar los datos completos.

Sin embargo, lo recomendable aquí es que el campo carrera sea capturado con claves y en un archivo independiente pero a la vez relacionado, tener los nombres de carrera completos. Todo esto permite, por una parte, ahorro en el tiempo de captura ya que es más fácil y más rápido teclear una clave que el nombre completo de la carrera, y por otra, menor capacidad de almacenamiento, es decir, aunque el archivo de carreras ocupe determinado espacio de almacenamiento, será menor al espacio adicional en el archivo total si se colocaran los nombres completos; permitiendo obtener reportes completos relacionando los dos archivos.

Para que todo esto quede más claro y se observe la diferencia; en la figura 10 se tiene la longitud de los campos, suponiendo que el tamaño del archivo sólo incluyera los bytes que ocupan los datos (aunque realmente no es así⁸, para facilitar cálculos) y que se tienen 20 registros; se tendría:

Para el primer caso 14.72 MB (92 X 8 X 20). Para el segundo caso se utilizarían 14.08 MB ya que el archivo con claves de materia serían 10 MB y el archivo que contiene los nombres de carrera completo (12 carreras) 3.36 MB, teniendo un ahorro como puede verse de 640 bits de almacenamiento. Cabe señalar que conforme aumenta el número de registros el ahorro es mayor.

8 En un archivo además de grabar datos, se graban algunos caracteres útiles para el programa que los generó, así como marcas de inicio o fin de archivo; para el ejemplo no se toma en consideración todo esto ya que de alguna manera lo que interesa es notar la diferencia entre el tamaño de uno y otro archivo respecto a los datos.

Campo	Longitud
Nombre	30
Carrera	30
Edad	2
Domicilio	30

Total Longitud: 92

Campo	Longitud
Nombre	30
CveCarr	5
Edad	2
Domicilio	30

Total Longitud: 67

Figura 10: Como se ve en las tablas, para el primer archivo se tiene mayor longitud del registro, lo cual implica un ahorro de tiempo y almacenamiento si se trabaja con el segundo.

Por lo antes mencionado, el reducir la capacidad de almacenamiento es un punto muy importante en todo diseño ya que hay un ahorro en costo -como se dijo con anterioridad- que es trascendental en el buen funcionamiento de una empresa.

TIPOS DE COMPRESION

Para esta reducción se habla de dos tipos de compresión: *Compresión Lógica* y *Compresión Física*. Un ejemplo del primer tipo de compresión sería el antes mencionado; otro sería que un campo tipo fecha en lugar de almacenarse como comunmente se hace (010490:DD-MM-AA) se almacenara utilizando representación binaria; por lo que, si de la primera forma se utilizan 6 caracteres, para la segunda sólo serían dos. Esto es:

Representación Numérica	01	04	81
Representación Binaria	00001	0100	1010001

Se observa que se necesitan 5 bits para poder representar hasta el número 31 (días), 4 para tener hasta el mes 12 y 7 para tener hasta el año 2000; por lo que agrupando estos bits se forman sólo dos caracteres. Entonces se puede decir que la compresión lógica es un método útil en el diseño de base de datos, ya que minimiza su tamaño.

La *Compresión Física* se puede considerar como el proceso de Reducir la cantidad de datos, además aprovecha el hecho de que cuando un dato es codificado y separado en diversas entidades, las propiedades de ocurrencia de caracteres y grupos de caracteres difieren, en este tipo de compresión lo que se hace es codificar aquellos que se repiten utilizando pocos bits, en tanto que los caracteres que no son tan frecuentes son codificados con más bits.

El minimizar los requerimientos de almacenamiento utilizando la compresión de datos implica a su vez una disminución en los tiempos de ejecución de los programas, esto se debe a que los accesos a discos son menores.

Así también para transmisión, es una ventaja la reducción de datos, puesto que decrece el tiempo utilizado en dicho proceso.

TIPOS DE COMPRESION FISICA

En la actualidad la compresión se usa tanto para reducir el espacio utilizado en almacenamiento de datos como para la transmisión de los mismos, debido a que, con ello se reducen los períodos de tiempo empleados en ello.

Dentro de las técnicas de compresión física podemos mencionar las siguientes:

- Null Supression (Supresión de Nulos o Blancos)
- Bit Mapping (Mapeo Binario)
- Run Length (Supresión de Cadenas de Caracteres)

- Half-Byte Packing (Empacamiento de Medio's-Byte)
- Diatomic Encoding
- Statistical Encoding (Codificación Estadística)
- Otras

De acuerdo con las necesidades requeridas, alguna de las técnicas antes mencionadas será más eficiente que las demás.

Null Supression

Esta es una técnica simple; como lo indica su nombre, realiza una eliminación de los "blancos" (o nulos) contiguos que se repiten en una cadena de datos. Lo que hace el algoritmo es reemplazar los "blancos" o caracteres nulos por un par de caracteres. El primero es el indicador de la supresión de "blancos", en tanto que el segundo indica la cantidad de dichos caracteres encontrados en secuencia (Figura 11).

Datos iniciales	XYZbbbbbQRX
Compresión de datos	XYZSc5QRX

donde:

Sc = Indicador de Supresión

Como son dos caracteres los que se utilizan para representar cuando hay supresión, entonces no tiene caso suprimir dos blancos o uno; ya que si son dos no se estaría reduciendo nada y con uno se agregaría un caracter

Para reconstruir la cadena de datos original, se crea un algoritmo el cual debe reconocer el caracter especial que indica supresión y además reconozca que el siguiente caracter representa el número de blancos que fueron suprimidos.

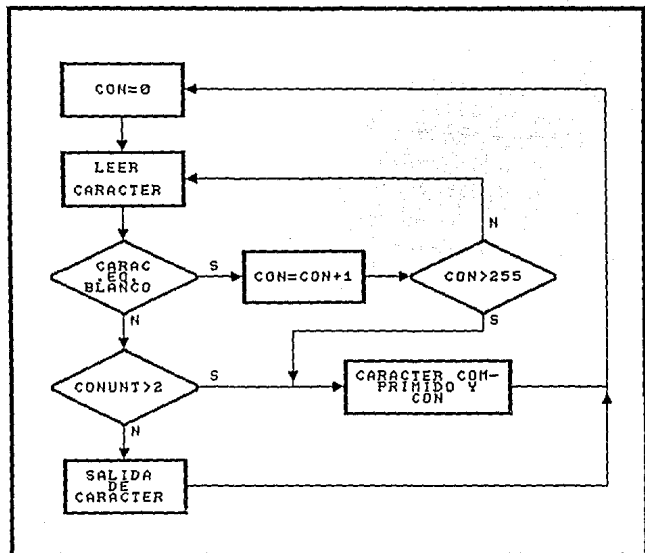


Figura 11: Diagrama de Supresión de Nulos o Blancos

Bit Mapping

La técnica Bit Mapping es útil para reducir un conjunto de datos constituidos de una proporción alta de un tipo específico de datos, los cuales pueden ser numéricos, cadenas de algún carácter especial o bien blancos. Como su nombre lo dice, un bit map se emplea para indicar la presencia o ausencia de datos.

Agregando un "bit map" al inicio de la cadena, se puede indicar la presencia o ausencia de nulos (si lo que se está reduciendo son blancos) de este modo se reduce el tamaño de la cadena de datos.

Su algoritmo se muestra en la siguiente página (Figura 12).

En la rutina de este proceso, primero se inicializan *bit*, *bit map* e *i* (el contador de caracteres); se toma el primer elemento y entonces se compara el contador de caracteres con 8, si es igual termina la rutina, si no, incrementa dicho contador. Después compara el caracter con un blanco; si es blanco, almacena el *bit* como cero si no almacena el *bit* como 1, en ambos casos se regresa a leer otro dato.

En este método, dentro del primer byte (*bit map*) un "cero" indica la presencia de un nulo, y con un "uno" su ausencia o bien presencia de otro caracter diferente de nulo, seguido de este byte, se presentan los caracteres que no son nulos. Un ejemplo de esto se muestra en la siguiente cadena:

Dato1	Nulo	Nulo	Dato2	Nulo	Nulo	Nulo	Dato3
-------	------	------	-------	------	------	------	-------

utilizando la técnica de Bit Mapping, se obtiene

Bit Map	Dato1	Dato2	Dato3
---------	-------	-------	-------

donde *bit map* es el número binario:

1 0 0 1 0 0 0 1

analizando de izquierda a derecha.

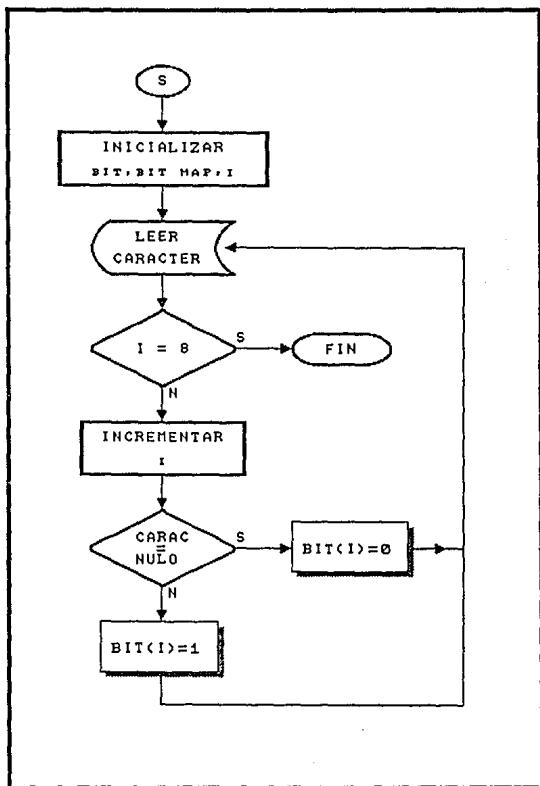


Figura 12: Diagrama de la técnica B. M.

Si se analiza de derecha a izquierda, ahora queda como sigue:

1 0 0 0 1 0 0 1

Por tanto, con el primer número 10010001 se observa que en la primer posición se tiene un caracter que no es blanco, luego los dos ceros siguientes indican que le siguen dos blancos, después otro caracter no nulo, posteriormente tres blancos y finalmente un dato no nulo representado por el número 1.

Este método sólo permite ir analizando cadenas de ocho datos ya que se tienen sólo ocho posiciones (byte) para representar un dato.

En el ejemplo anterior la reducción fue de un 50 por ciento ya que de ocho datos de la cadena se redujo a 4. Cuando dentro de la cadena no existe ningún blanco en lugar de haber reducción, existe un incremento en el número de datos debido al bit map, por tanto entre mayor número de nulos o blancos haya, la eficiencia de reducción aumentará. En la tabla siguiente se muestra la eficiencia de compresión para blancos.

Porcentaje Nulos	No. de Nulos Cad. Original	Tamaño de la Cadena Resultante	Razón de Compresión
0.0	0	9	0.888
12.5	1	8	1.000
25.0	2	7	1.143
37.5	3	6	1.334
50.0	4	5	1.600
62.5	5	4	2.000
75.0	6	3	2.667
87.5	7	2	4.000
100.0	8	1	8.000

Matemáticamente la eficiencia puede ser representada con un modelo. Si suponemos que p es la probabilidad de ocurrencia de nulos, con un largo de cadena S , entonces

$$S * (1 - p) + S / 8 = S_p$$

donde S_p es el número supuesto de nulos dentro de una cadena.

Por tanto la Razón de compresión será:

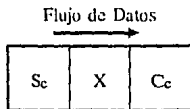
$$(1/S \{S(1+p) + S/8\})^{-1} = \{(1-p) + 1/8\}^{-1}$$

Run Length

Este método de compresión permite suprimir cualquier caracter que se repita secuencialmente. Se puede decir que es la técnica mejorada de Null Supression.

A diferencia del primer método, éste hace uso de un caracter adicional el cual indica que existe compresión, por lo que ahora en lugar de usar dos caracteres para representar la reducción, serán tres: el primero, como ya se mencionó, indicará que existe compresión, el segundo es el caracter suprimido y el tercero la cantidad de veces que aparece en forma consecutiva.

El proceso a seguir en Run Length, es convertir la existencia de caracteres repetidos (consecutivamente) en una cadena como la que a continuación se ilustra:



donde:

S_c = Caracter especial indicando compresión

X = Caracter repetido

C_c = Contador de caracter. Número de veces que se repite el caracter.

Como se requieren tres caracteres para indicar la compresión entonces el método solamente es efectivo para una secuencia de 4 o más caracteres repetidos.

Los pasos de este método se muestran en el siguiente diagrama (Figura 13).

Primeramente, se inicializan los contadores de caracter y repetición; se toma el primer dato o caracter; el contador de caracteres es incrementado en 1. Este es comparado con 1; en el primer ciclo siempre será igual a 1, entonces el caracter es almacenado temporalmente para después ser procesado si la cadena original contiene cuatro o más repetidos. Para los siguientes ciclos (del dos en adelante) el dato "leído" de la cadena original es comparado con el primer caracter almacenado. Si el presente caracter es igual al caracter en almacenamiento, la compresión puede ser posible sí, como anteriormente se mencionó, cuatro o más caracteres idénticos son encontrados en secuencia. Entonces el contador de caracter es incrementado en 1 y se lee otro dato. Si no son iguales el contador de repetición de caracter es comparado con cuatro; si es menor se inicializan nuevamente los contadores y se lee otro dato (no hay compresión), de lo contrario se pone el formato de compresión.

Para tener nuevamente la cadena original, se hace uso de una bandera (flag) la cual es puesta en Off, se lee el primer caracter. Si la bandera de "compresión" esta en Off -como ocurre en el primer ciclo- se compara dicho caracter con el caracter especial que indica compresión; si es igual, la bandera es puesta en On y se lee un nuevo dato, pero si es diferente únicamente se lee el dato siguiente. De la primera comparación, si la bandera está en On, entonces es obtenido el caracter repetido, así como el siguiente caracter que representa la cantidad de caracteres repetidos y entonces es aplicado el formato de descompresión.

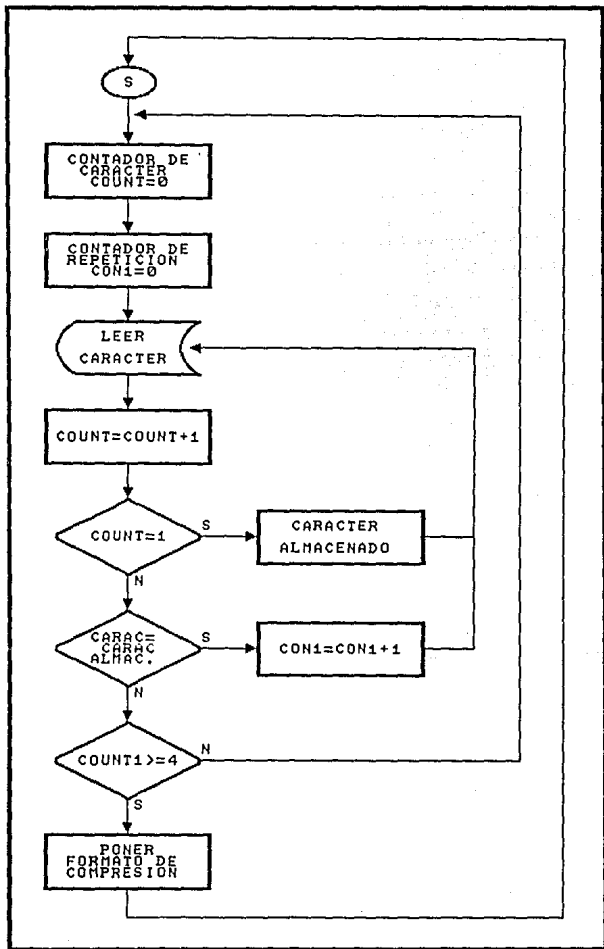


Figura 13: Diagrama para Run Lengh

En esta técnica la eficiencia depende del número de ocurrencias de caracteres repetidos en la cadena de datos.

No. de ocurrencias de caracteres repetidos	Promedio de caracteres repetidos	Razón de Compresión
10	4	1.010
10	5	1.020
10	6	1.031
10	7	1.042
10	8	1.053
10	9	1.064
10	10	1.075
20	4	1.020
20	5	1.042
20	6	1.064
20	7	1.087
20	8	1.111

No. de ocurrencias de caracteres repetidos	Promedio de caracteres repetidos	Razón de Compresión
20	9	1.136
20	10	1.163
30	4	1.031
30	5	1.064
30	6	1.099
30	7	1.136
30	8	1.176
30	9	1.220
30	10	1.266
40	4	1.042
40	5	1.087
40	6	1.136
40	7	1.190
40	8	1.250
40	9	1.316

Half-Byte Packing

Esta técnica de compresión de datos es vista como una derivación de la segunda técnica llamada Bit Mapping. Es utilizada fundamentalmente cuando una porción del patrón de bits es usada para representar esos caracteres. Por ejemplo,

analizando el código EBCDIC, se observa que para representar números los cuatro primeros bits (mitad del byte) son los mismos.

Estruc. Bit	Caracter Num.
1111 0000	0
1111 0001	1
1111 0010	2
1111 0011	3
1111 0100	4
1111 0101	5
1111 0110	6
1111 0111	7
1111 1000	8
1111 1001	9

Así entonces, si se tiene una secuencia continua de números, en otra técnica no se podría realizar compresión ya que no es el mismo caracter que se repite; sin embargo, con Half-Byte Packing si se puede realizar ya que los cuatro primeros bits se repiten. De la misma manera que Run-Length utiliza un caracter especial que indica compresión también esta técnica lo requiere.

También puede ser tomado el código ASCII ya que como se observa en la siguiente tabla, se utilizan los tres dígitos (ignorando el bit de paridad) para indicar que se trata de caracteres numéricos, alfabéticos o caracteres especiales.

Estruc. Bit	Caracter
011 0000	0
011 0001	1
011 0010	2
011 0011	3
011 0100	4
011 0101	5
011 0110	6
011 0111	7
011 1000	8
011 1001	9
010 0100	\$
010 1100	,
010 1110	.
010 1010	*

Utilizando este formato, se pueden comprimir cadenas hasta de 15 caracteres del mismo tipo repetidos secuencialmente, debido a que en cuatro bits el número máximo a representar es 15 (1111) que es medio byte. En tanto que si se utiliza un byte completo para el contador de caracteres se puede tener hasta 255 caracteres empacados. La representación de este método se muestra en la figura (14):

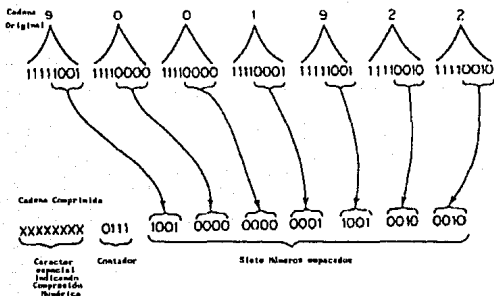


Figura 14: Half-Byte Packing

Al inicio de esta rutina primero se coloca en cero el contador (Contador=0), se toma el dato y se compara; si es numérico, el contador se incrementa en 1 y se toma otro dato, si no es así, entonces el contador es comparado con 4. Si es menor o igual el algoritmo regresa al primer bloque, pero si es mayor entonces se aplica el formato de compresión y regresa nuevamente al paso 1.

Si se está utilizando un medio byte para el contador, entonces se tiene que hacer otra comparación.

Diatomic Encoding

Diatomic Encoding es una técnica que consiste en tomar un par de caracteres y sustituirlos por uno especial por lo cual a simple vista se ve que la razón de compresión es de 2:1. Es decir



Sin embargo la razón de compresión va a depender del número de caracteres especiales disponibles para representar los pares de caracteres frecuentemente ocurrentes.

Su diagrama es el siguiente (Figura 15):

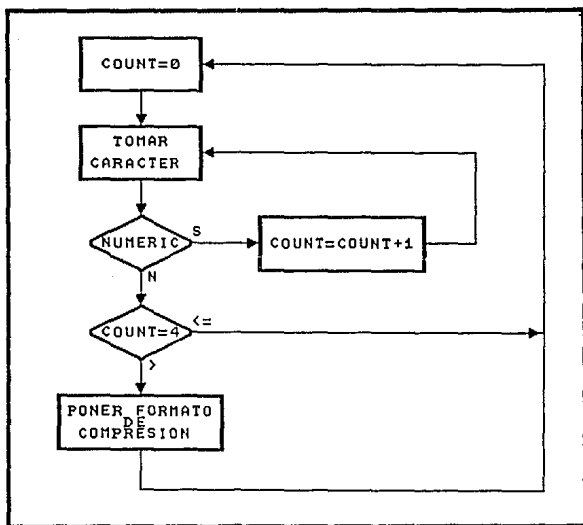


Figura 15: Diagrama de Diatomic Encoding

Statistical Encoding

Es un proceso que puede ser usado para obtener una minimización del promedio de la longitud del código de los datos, es una manera similar a la que Morse utilizó para sus claves, haciendo uso de puntos y guiones para representar caracteres, de tal manera que un punto representaba la letra "E" la cual es encontrada muy frecuentemente en el idioma inglés, mientras que una cadena larga de puntos y guiones eran usados para representar caracteres que aparecían con menor frecuencia. El desarrollo de este proceso es realizado utilizando la técnica de Huffman.

El método de Huffman es empleado para reducir el promedio de la longitud de los símbolos de un "alfabeto"; ya sea el alfabeto de un lenguaje, el código ASCII o el EBCDIC, etc. En otras palabras, es utilizar el menor número de bits para representar un carácter, de acuerdo a la probabilidad de ocurrencia.

Para entender el método es necesario tener claros algunos conceptos básicos, los cuales permitirán un entendimiento de cómo la redundancia puede ser reducida estadísticamente.

Teoría

Para un sistema eficiente de transmisión en n planos discretos en " λ " intervalos (segundos), el número de combinaciones de señales diferentes en T segundos es $n^{T/\lambda}$. La información es proporcional al tiempo de transmisión, el logaritmo de $n^{T/\lambda}$ es utilizado para obtener la información transmitida en T segundos proporcional a $(T/\lambda) \log n$.

El factor de proporcionalidad dependerá de la base del logaritmo empleado (lo más común es de base 2). Esto es:

$$H = T/\lambda \log_2 n$$

La unidad de información definida en forma anterior es conocida como bit o dígito binario. En la transmisión de datos para un período de 20 seg. usando 2

niveles discretos (0 y 1) en intervalos de 1 seg, la información contenida se convierte en:

$$H = 20/1 \log_2 2 = 20 \text{ bits}$$

La capacidad de un sistema es definida como, la cantidad máxima de información por segundo que un sistema puede transmitir y se expresa en bits por segundo. Para el ejemplo anterior:

$$C = H/T = 1/\lambda \log_2 n = 1/1 \log_2 2 = 1 \text{ bit/seg}$$

La frecuencia relativa de ocurrencia (alguna combinación o evento⁹) es definida como la probabilidad de ocurrencia, denotada por P, donde

$$P = \frac{\text{Número de veces ocurridas en un evento}}{\text{Total del número de posibilidades}}$$

Si n son los eventos posibles en n señales posibles, entonces $P = 1/n$ para ocurrencia de eventos semejantes. La información contenida por la aparición de algún evento en un intervalo de tiempo (H_1) es:

$$H_1 = \log_2 n = -\log_2 P \text{ bits/intervalo}$$

donde P representa $1/n$. Durante t períodos de tiempo, consistiendo de " λ " (en segundos) períodos, se obtiene la información total:

$$H = tH_1 = -t \log_2 P \text{ bits en } t \text{ períodos}$$

Como el número de períodos t es igual al tiempo total (T) dividido por el número de intervalos (λ), la información disponible en T segundos será:

$$H = -T/\lambda \log_2 P = T/\lambda \log_2 n \text{ bits en } T \text{ seg}$$

⁹ El conjunto de todos los resultados posibles de un experimento estadístico se llama espacio muestral y se representa por el símbolo S.

Con el procedimiento anterior como base, se puede considerar el caso donde diferentes señales o eventos no tienen la misma probabilidad de ocurrencia. En este caso se asumen dos niveles a tratar, 0 o 1, el primero con probabilidad P y el segundo con Q , de tal manera que la suma de ambas será la unidad ($P + Q = 1$)¹⁰. Entonces:

$$P = \frac{\text{Número de ocurrencias de 0}}{\text{Total del número de posibilidades}}$$

$$P = \frac{\text{Número de ocurrencias de 1}}{\text{Total del número de posibilidades}}$$

Por tanto del desarrollo anterior y tomando en cuenta que $P = P_1 + P_2 + \dots + P_n$, se obtiene:

$$H = -t \cdot \sum_{i=1}^n P_i \log_2 P_i \text{ bits en } t \text{ períodos}$$

Para intervalos de tiempo T :

$$H = -T/\lambda \cdot \sum_{i=1}^n P_i \log_2 P_i \text{ bits en } t \text{ segundos}$$

Y finalmente para un mensaje con n símbolos (o niveles) posibles con probabilidad de ocurrencia P_i a P_n el promedio de información por intervalos simples "λ" es:

$$H_{\text{avg}} = \sum_{i=1}^n P_i \log_2 P_i \text{ bits/símbolos}$$

Por ejemplo:

Si se lanza una moneda al aire, sabiendo que ésta tiene cara (H) y cruz (T), entonces se tienen dos símbolos X_1 y X_2 que pertenecen a un conjunto X . De la misma manera, si ahora se lanzan dos monedas, cuyos resultados se representan por

10 Ya que el valor límite de una probabilidad es 1.

$T=0$ y $H=1$, la probabilidad del lanzamiento sería:

Resultado	Símbolo	Probabilidad	Código
TT	X ₁	0.25	00
TH	X ₂	0.25	01
HT	X ₃	0.25	10
HH	X ₄	0.25	11

Por tanto, la entropía a promedio de bits requeridos para representar cada resultado es:

$$H_{\text{avg}} = - \sum_{i=1}^4 P_i \log_2 P_i = - 4(0.25 \cdot \log_2 0.25) = 2$$

Lo que significa que se requiere dos símbolos binarios para la representación de cada símbolo (X₁,...,X₄) o resultado como se tiene en la tabla anterior.

Si por alguna razón, el lanzamiento de la moneda fue fijo, de tal forma que sólo existieron ocurrencias de cruz (T) entonces sólo un símbolo es requerido (X₁). Por lo que utilizando la fórmula para obtener la entropía, se tiene:

$$H_{\text{avg}} = - \sum_{i=1}^1 \log_2 1 = 0$$

Entonces se dice que el símbolo no provee información; esto es, la entropía es cero.

Si ahora, para el ejemplo anterior, se fija la probabilidad de las ocurrencias de cruz (T) es incrementada a 0.75 por lo que la probabilidad de ocurrencias de caras será de 0.25, quedando la tabla de probabilidad como sigue:

Resultado	Símbolo	Probabilidad	Código
TT	X ₁	0.5625	00
TH	X ₂	0.1875	01
HT	X ₃	0.1875	10
HH	X ₄	0.0625	11

Obteniendo nuevamente la entropía, utilizando las nuevas probabilidades:

$$\begin{aligned}
 H_{avh} &= -0.5625 \log_2 0.5625 + 0.1875 \log_2 0.1875 \\
 &\quad + 0.1875 \log_2 0.1875 + 0.0625 \log_2 0.0625 \\
 &= 1.62 \text{ bits por símbolo}
 \end{aligned}$$

Comparando este resultado con el del primer ejemplo en donde se suponía una misma probabilidad (0.25 para todos) el promedio se redujo 0.38; es decir, se obtuvo una reducción de cadenas grandes de datos, asignando códigos pequeños a los datos cuya ocurrencia sea más frecuente y códigos mayores para datos con menor grado de ocurrencia. Para obtener estos códigos se emplea la codificación de Huffman.

Código de Huffman

El código de Huffman es un código óptimo cuyo resultado es el promedio más pequeño de los métodos estadísticos. Existen dos reglas para este método:

1. El código de cada carácter debe ser único.
2. No habrá inicio o fin de código una vez que el flujo de datos inicia.

El ejemplo anterior empleando el método de Huffman, es graficado a continuación:

Caracter	Probabilidad		Código
X1	.5625		0
X2	.1875		10
X3	.1875		110
X4	.0625		111

Para obtener el diagrama anterior se realizan los siguientes pasos:

- Los símbolos son listados en orden descendente.
- Se agrupan las dos frecuencias más pequeñas (X3, X2) para obtener una frecuencia compuesta llamada Nodo.
- La frecuencia compuesta (0.25) es agrupada con X2.
- La frecuencia obtenida en el paso 3 es agrupada con la primera frecuencia de la lista (X1) para obtener la unidad que es la probabilidad de ocurrencia de los cuatro caracteres (Nodo Maestro).
- Asignar 0's y 1's.
- Construir código para cada caracter iniciando de derecha a izquierda, es decir, a partir del nodo maestro.

A partir de los códigos de Huffman se puede obtener la entropía o promedio:

$$H_{avg} = 1 \cdot 0.5625 + 2 \cdot 0.1875 + 3 \cdot 0.1875 + 3 \cdot 0.0625 = 1.63 \text{ bits}$$

el cual es muy aproximado al obtenido a través de la fórmula utilizada anteriormente, es decir, que a partir de esta técnica se está obteniendo el promedio más pequeño de bits por caracter.

De una manera general, el método se describe de la siguiente forma:

- Listar los símbolos en forma descendente de ocurrencia.
- Agrupar los dos últimos símbolos (frecuencias de ocurrencia menores) para construir el primer nodo, que es la suma de las dos frecuencias más pequeñas.
- Comparar este resultado con la antepenúltima frecuencia; si esta tercer frecuencia es mayor que la frecuencia compuesta entonces se suma para obtener otro nodo, de lo contrario esa tercer frecuencia (de abajo hacia arriba) se tiene que sumar a la cuarta frecuencia dando como resultado el segundo nodo, que será comparado con el primero. Si es mayor (el segundo) entonces se suman, pero si no, se suma el segundo nodo con una quinta frecuencia; cuyo resultado será un tercer nodo que se comparará con el primero, y así sucesivamente hasta agrupar todas las probabilidades y llegar a un sólo nodo que es el Nodo Maestro.
- Asignar 0's y 1's en cada uno de los nodos.
- Crear códigos partiendo del Nodo maestro a la izquierda; el bit encontrado en cada nodo formará parte del código.

Un ejemplo más completo de este método de Huffman es el se muestra en la siguiente figura:

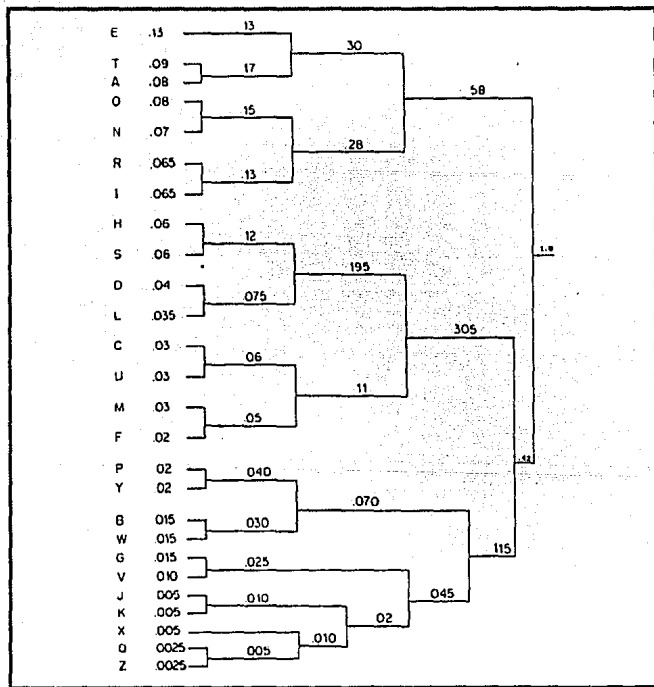


Figura 16: Ejemplo referente a la probabilidad de ocurrencia del alfabeto del idioma inglés.

Una vez que las probabilidades han sido agrupadas hasta llegar a la unidad se procede a asignar los 0's y 1's en cada nodo, quedando de la siguiente manera:

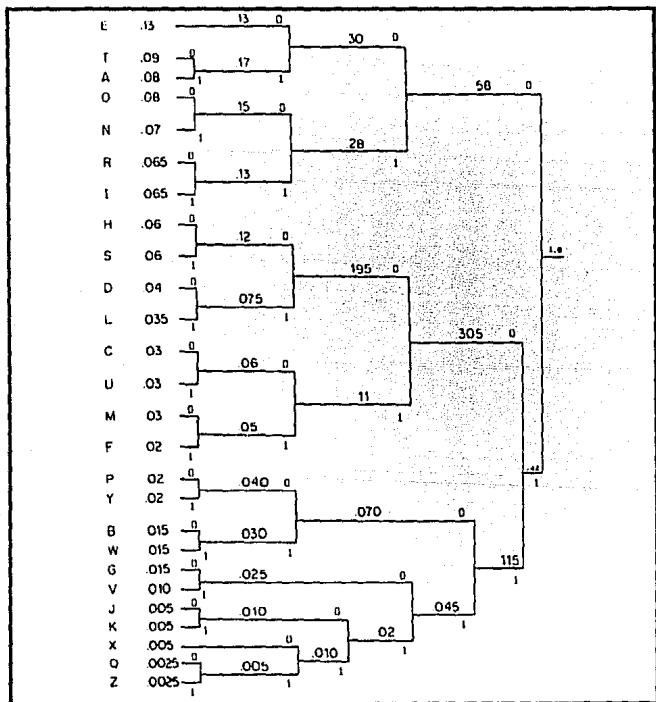


Figura 17: El mismo diagrama de la figura anterior con la asignación de 0's y 1's.

Finalmente el cálculo completo del número de bits requeridos para la representación de caracteres es sustituido por la siguiente fórmula:

$$b = f(-\log_2 P)$$

donde

P = probabilidad de ocurrencia de la letra

f(x) = el entero mayor o igual que x

Para el ejemplo anterior:

$$b = 2.94 \text{ bits/caracter}$$

Lo cual es aproximadamente 3 bits por caracter.

Otros

Pattern Substitution

Esta técnica es fundamentalmente una forma sofisticada de Diatomic Encoding, la diferencia radica en que en lugar del caracter especial se hace uso de caracteres "patrón".

Para el empleo de este método, se requiere una tabla de patrones. La tabla contiene una lista de argumentos y valores de función. Cada valor es un caracter especial indicador de compresión, el cual representa el valor de compresión de un argumento en particular.

Si se tiene una cadena como la que sigue:

NOW IS THE TIME FOR ALL GOOD MEN

Tabla de Patrones

Lista de Argumentos

Funciones Valor

THE
FOR
ALL

Sc1
Sc2
Sc3

La compresión de la cadena sería:

NOW IS S_{c1} TIME S_{c2} S_{c3} GOOD MEN

Para obtener la compresión, los datos originales son buscados dentro de la lista de argumentos, inicialmente igual al argumento más pequeño en la tabla de patrones. El argumento examinado es comparado con esa lista de argumentos de igual magnitud (número de caracteres). Si un par es obtenido, el valor asociado a la lista de argumentos reemplaza aquella porción de la cadena original de datos y entonces resulta una compresión de datos. Si el par no se obtiene, entonces se incrementa el ancho del argumento buscado para compararlo nuevamente con el argumento (de la lista) que tenga el mismo ancho y repetir nuevamente el proceso.

Un segundo método para mejorar esta técnica resulta del uso de blancos como delimitadores. El valor binario u octal de los caracteres entre blancos puede ser generado y comparado con el valor binario u octal en la porción de la lista de argumentos de la tabla de patrones. Este proceso simplifica la búsqueda de un argumento largo así como disminuye el tiempo requerido para la codificación de patrones.

Relative Encoding

La técnica de Relative Encoding es empleada de manera efectiva cuando hay una secuencia de "tipos" (géneros) en la cadena de datos original, no para archivos convencionales. Un ejemplo son los datos de telemetría, que a continuación se explica.

Para la generación de datos en telemetría, se usa un dispositivo sensor para registrar lecturas en intervalos predefinidos. Estas son transmitidas a un sitio central para procesos adicionales. Ejemplos de señales de telemetría son: el espacio de números, los cuales representan lecturas de temperatura, análisis de espectros de colores y otros datos, en intervalos predefinidos. Normalmente las señales de telemetría contienen una secuencia de campos numéricos que se componen de subsecuencias de géneros o números, como se observa.

Lectura de telemetría original:

46 46 46.1 46.1 46.1 46 46 46 46.1 46.1 46.1 46.2

Lo que realiza el método es tomar el primer dato como referencia, de tal manera que el segundo dato va ser representado con un valor relativo del primero; posteriormente, el tercero sería un valor relativo respecto al segundo y así sucesivamente. Estos valores relativos son incrementos o decrementos (indicado con un signo menos) del número inmediato anterior.

De tal manera que las lecturas anteriores, utilizando Relative Encoding quedan como sigue:

46 0 .1 0 0 -.1 0 0 .1 0 0 .1

De lo anterior se ve que el primer valor es pasado tal cual, como la segunda lectura es la misma (cadena original), es decir, no se incrementa ni decremента, entonces se coloca un cero. La tercera lectura es 46.1, por tanto existe un incremento de .1 respecto a la segunda por lo cual se coloca un .1, para las restantes lecturas se trabaja de la misma manera hasta obtener todos los datos. Haciendo una comparación de las lecturas originales y las lecturas "comprimidas"; se visualiza que las primeras son 38 caracteres (tomando en cuenta los números y puntos decimales), los cuales fueron reducidos a solo 18; existiendo por tanto una reducción del 52 por ciento.

Forms Mode Operation

Este método es empleado cuando los datos son transmitidos de y hacia una pantalla en una serie de formas predefinidas. Cuando se opera en modo de formas, la pantalla puede ser utilizada para llenar los tipos de blancos de operación. En este modo de operación dos tipos de datos son desplegados: información protegida o fija y variables de información. La información fija corresponde a la información de los nombres de campo (nombre, dirección, etc.). En el modo de formas, el campo fijo puede verse como la pregunta mientras que la información introducida en la variable correspondiente es considerada como respuesta.

Usando esta técnica, el operador denota la forma de él o ella y esa forma es transmitida de la computadora a la terminal o es generada localmente dentro de la memoria de la terminal o desde un dispositivo periférico fijado a la terminal. Los campos fijos son precedidos por una "FS" (inicia campo fijo) en tanto que un campo variable es precedido por una "GS" (inicia campo variable) y un caracter parámetro; el cual es usado para definir las operaciones permitidas dentro de los campos "variables" tal como numéricos solamente, alfabéticos, alfanuméricos, transmisión inhabilitada y así sucesivamente.

CAPITULO IV. Aplicación de los Sistemas Reales

Este sistema permite realizar consultas de datos de un individuo, mapas, dibujos, etc. anexando su imagen digital. Un sistema de este tipo puede ser de gran ayuda e interés para aquellos usuarios que deseen contar por ejemplo con un directorio de su personal; que pueda mostrarle, además de sus datos personales, su imagen. También puede ser de gran utilidad en la obtención de información para un catálogo de equipo, archivos de dibujo o proyectos realizados.

Este sistema se desarrolló a través del manejo de archivos tradicionales y se anexa su imagen digital (Figura 18).

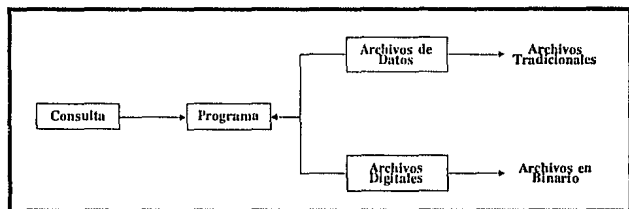


Figura 18: Diagrama General del Sistema

Uno de los problemas que se presentó fue que la imagen digital ocupaba un rango de 38000 bytes por lo que se tuvo que realizar una compactación de dicha imagen, reduciendo éste hasta un 90 por ciento¹¹ (Sistema mejorado).

11 El archivo de formato TIFF (The Tagged Image File Format) generado por el paquete PaintShow es de 38,740 bytes es reducido entre un rango de 3.8 y 20 KB. Se habla de rangos porque dependiendo de la imagen es la reducción.

El procedimiento de la elaboración es el siguiente:

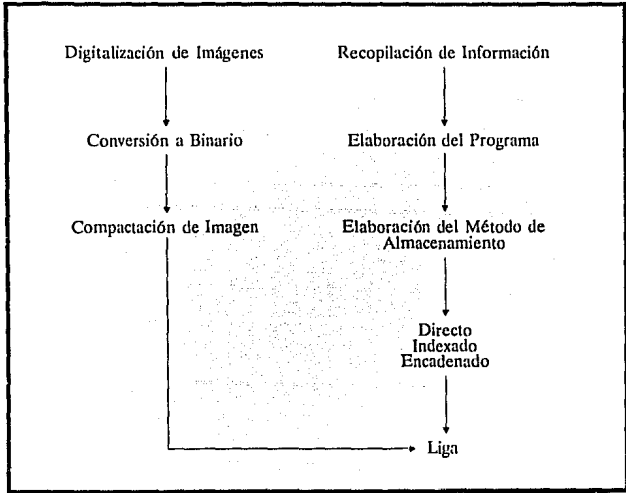


Figura 19: Diagrama del Procedimiento de la Elaboración del Sistema

SISTEMA EN BASIC (Primer Sistema)

Este sistema al igual que cualquier otro proyecto a desarrollar ha ido pasando por diversas etapas. Primeramente el sistema fue realizado en Basic, debido a que en este lenguaje se cuenta con una instrucción "BLOAD" (palabra reservada) la cual permite mostrar un archivo binario directamente en la pantalla.

La creación de este sistema es utilizando fotografías de personas incluyendo sus datos tales como: nombre, dirección y teléfono.

Uno de los pasos para este sistema fue digitalizar las fotografías utilizando un Scanner llamado ScanMan de Logitech (cuyas características fueron descritas en el primer capítulo), así como el software propio de dicho scanner [PaintShow].

Una vez que la fotografía se encuentra en la pantalla del PaintShow ésta es capturada para que el archivo obtenido sea convertido a Binario, de tal manera que pueda ser presentado en la pantalla utilizando el programa en basic.

El programa permite la consulta de los datos, es decir, el archivo contiene los datos de cada persona como: Nombre, Dirección y Teléfono así como el nombre correspondiente al archivo de la fotografía; este último dato es utilizado para que cuando se realice la consulta se llame ese archivo con BLOAD. Su algoritmo es presentado en la figura 20.

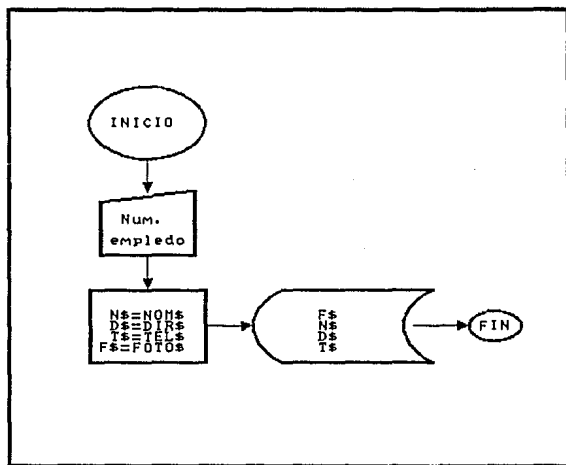


Figura 20 Diagrama del primer sistema

SISTEMA UTILIZANDO BASES DE DATOS (Segundo Sistema)

Esta idea surge porque generalmente los datos correspondientes al personal de una empresa (de un catálogo de equipo, material, etc.) se tienen en una base de datos; entonces basta con agregar un campo más, el cual corresponde al identificador o nombre de archivo de la imagen.

En este caso el sistema se desarrolla utilizando una base, creada en Dbase III Plus, que contiene los datos de equipo neumático, aunque claro, también puede ser de datos de personas como en el primer sistema o bien de otro tipo.

Se realiza un programa en Dbase que permita el uso de los datos de la base. Lo que hace este programa es pedir el nombre o clave de la figura a consultar para que el registro sea localizado; los datos de dicho registro son grabados en un nuevo archivo los cuales serán tomados por un programa en Basic para mostrarlos en la pantalla.

El nombre de la figura es también guardado en un archivo para ser utilizado por otro programa que muestra la imagen en pantalla.

Como se puede observar, este segundo sistema es un conjunto de varios programas, compuesto primeramente por el programa principal que permite acceder a la base de datos; el que muestra la imagen; y, el que coloca los datos en pantalla, así como, un cuarto programa, que pone la pantalla en alta resolución.

En el primer sistema no se puso atención en la resolución o calidad de las fotografías; sin embargo en este segundo sistema es importante ya que por tratarse de equipo se requería de una mayor calidad al desplegar la imagen, de tal manera que surgieron dos problemas: en primer lugar, el de convertir un archivo TIFF¹² en binario, mediante programa, ya que el paquete Story Board sólo hace esta conversión

12 Archivos (TIFF) que genera el paquete PaintShow, es decir, el software del Scanner utilizado.

en resolución menor (CGA), el segundo problema es desplegarlo con una mayor calidad; esto es, no basta con tener el archivo preparado para alta resolución si la pantalla no lo está.

Para el algoritmo de conversión se toman en cuenta dos puntos:

1. Los primeros 174 bytes del archivo¹³, generado al "scannear" la imagen son de uso exclusivo para el software del Scanner, por tanto no son útiles para el archivo final (Figura 21).

** P1.TIF	
0000(0000)	49 49 2A 00 08 00 00 00 0C 00 FF 00 03 00 01 00
0016(0010)	00 00 01 00 00 00 00 01 03 00 01 00 00 00 80 02
0032(0020)	00 00 01 01 03 00 01 00 00 00 E0 01 00 00 02 01
0048(0030)	03 00 01 00 00 00 01 00 00 00 03 01 03 00 01 00
0064(0040)	00 00 01 00 00 00 06 01 03 00 01 00 00 00 01 00
0080(0050)	00 00 07 01 03 00 01 00 00 00 01 00 00 00 11 01
0096(0060)	04 00 01 00 00 00 AE 00 00 00 15 01 03 00 01 00
0112(0070)	00 00 01 00 00 00 1A 01 05 00 01 00 00 00 9E 00
0128(0080)	00 00 18 01 05 00 01 00 00 00 A6 00 00 00 1C 01
0144(0090)	03 00 01 00 00 00 01 00 00 00 00 00 00 00 41 00
0160(00A0)	00 00 01 00 00 00 41 00 00 00 01 00 00 00 FF FF
0176(00B0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0192(00C0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0208(00D0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0224(00E0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0240(00F0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Figura 21: En esta figura se observa un recuadro en el se marcan los primeros 174 bytes que serán borrados.

- 13 Un archivo TIFF consiste de una cabecera, seguido de un número de etiquetas o campos. Es decir, estos datos proporcionan información acerca de la estructura de la imagen como: el tipo de compresión (si se está utilizando), el largo y ancho, el scanner utilizado, nombre de la persona que creó la imagen.

2. El nuevo archivo debe contener 7 bytes al inicio que son los que identifican a un archivo como binario, es decir:

FD 00 B8 00 00 00 96

de tal manera que el archivo quedaría como se muestra en la figura siguiente:

P2.BIN	
0000(0000)	FD 00 B8 00 00 00 96 FF FF FF FF FF FF FF FF
0016(0010)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0032(0020)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0048(0030)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0064(0040)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0080(0050)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0096(0060)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0112(0070)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0128(0080)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0144(0090)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0160(00A0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0176(00B0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0192(00C0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0208(00D0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0224(00E0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0240(00F0)	FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Figura 22: Aquí se muestra como queda el archivo una vez que ha sido transformado.

El algoritmo resultante es el mostrado en la figura 23.

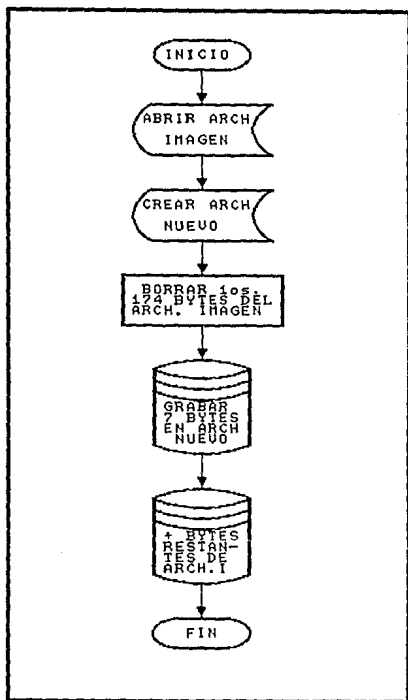


Figura 23: Algoritmo que permite la conversión de un archivo TIFF a Binario.

Para que este nuevo archivo pueda ser desplegado en la pantalla con una resolución mayor, se realiza un pequeño programa que utiliza la interrupción 10

(Hexadecimal) del BIOS que se encuentra en ROM para manipular los registros de tal manera que la pantalla o el video es preparado para tal efecto¹⁴.

Es decir, los registros que serán alterados son AL y AH, que son la parte baja y alta del registro A respectivamente.

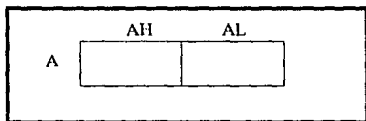


Figura 24: Los registros tienen una parte alta y una parte baja.

Para activar la pantalla en alta resolución se coloca un 11 en AL y 00 en AH (números hexadecimales).

- 14 Una interrupción es una señal al procesador procedente de un programa o de un dispositivo material. Las interrupciones suspenden momentáneamente las actividades del procesador con objeto de que este pueda llevar a cabo una cierta tarea.

Una interrupción tiene una sección de código (gestionador de interrupciones), es decir, un cierto valor residente en memoria. Cuando una interrupción es utilizada lo que hace el procesador es ejecutar ese código; al finalizar el procesador la tarea solicitada continua con la actividad que estaba desarrollando antes de la interrupción. El procesador encuentra estos gestionadores utilizando vectores de interrupción, los cuales son como una tabla con tres columnas y N filas, donde N es el número de interrupciones que permite el sistema. Antes de pasar el control al gestionador de interrupciones, el sistema almacena información sobre el programa actual en la pila, lo que permite al sistema volver a tomar el control después de que el gestionador termine.

Con la interrupción 10 se está indicando que se utilice el Servicio de Video (Figura 25), como este servicio tiene varias funciones se pone 00 en AH para utilizar el Modo de Video; el poner 11 en AL indica una resolución de 640 X 480 en dos colores y finalmente se utiliza la interrupción 20 para regresar a sistema operativo.

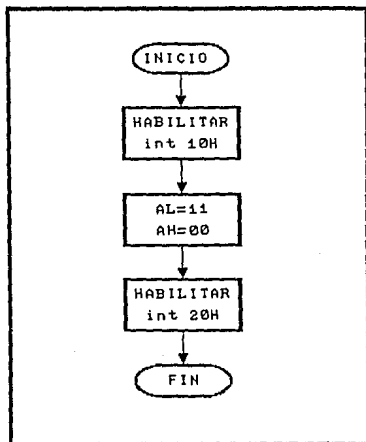
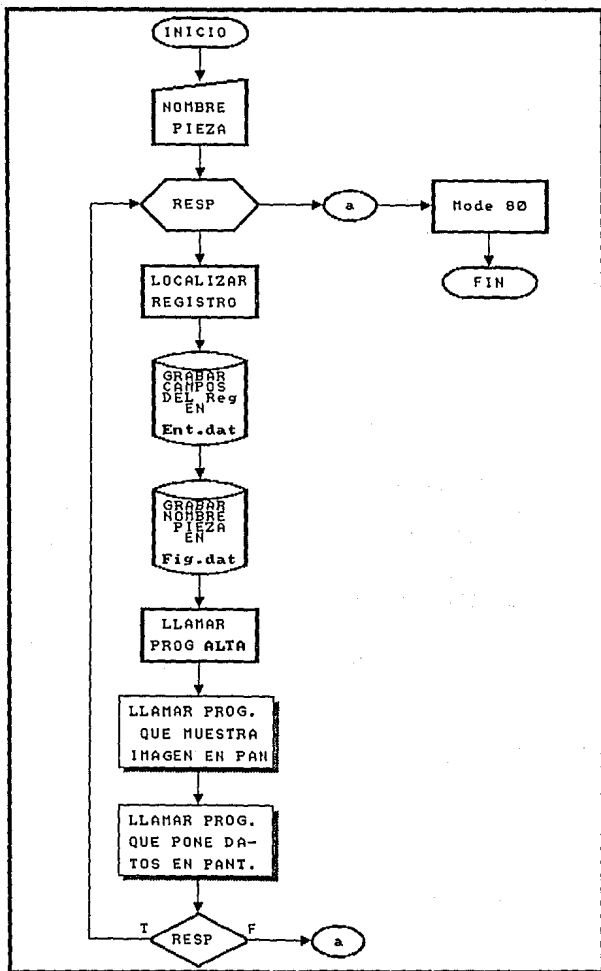


Figura 25: Algoritmo para poner la pantalla en alta resolución.

El diagrama de explicación del sistema completo se muestra en la Figura

26.



SISTEMA EMPLEANDO COMPACTACION (Tercer Sistema)

PROCESO DE COMPACTACION

El paso referente a la compactación dentro de este sistema se refiere a minimizar el espacio de almacenamiento en disco de los archivos de imágenes.

En el primer sistema los archivos de imágenes ocupaban 16392 bytes (archivos binarios) ya que eran presentados con menor resolución. Los archivos para el segundo sistema ocupan 38407 bytes debido a que de 38574 bytes del archivo en formato TIFF se le quitan 174 y se le agregan 7, esto con la finalidad de presentar las imágenes con mejor calidad.

Por tanto uno de los inconvenientes que se veía era que los archivos de imágenes ocupaban mucho espacio de almacenamiento en disco, es decir, si se contaba por ejemplo con 20 imágenes se debía disponer de casi 800 KB; esto es, más de la capacidad de un disco de 3 1/2.

Por lo cual un punto importante a resolver era el disminuir esa cantidad de almacenamiento. Además de realizar un programa que compactara los archivos de imagen, se debía hacer un programa que a través del archivo compactado se pudiera desplegar la imagen ya que de otra manera de nada serviría el que los archivos estuvieran reducidos.

Una vez revisados los diferentes procesos de compactación (cuyos algoritmos fueron presentados en el capítulo anterior) se eligió trabajar con el de Eliminación de Nulos o Blancos considerado el más óptimo para esta aplicación.

El algoritmo de compactación se explica de la siguiente manera (Figura 27): Se prepara un arreglo cuya capacidad es de 38574 Bytes (tamaño del archivo TIFF), se inicializan dos contadores: k que es la posición en la que se almacena el byte de salida y j que es el contador de caracteres (en este caso el carácter FF) que serán compactados.

Los datos del archivo de imagen (TIFF) son leídos y guardados en el arreglo $M[i]$ para que el proceso sea más rápido, se toma un carácter (byte) y se

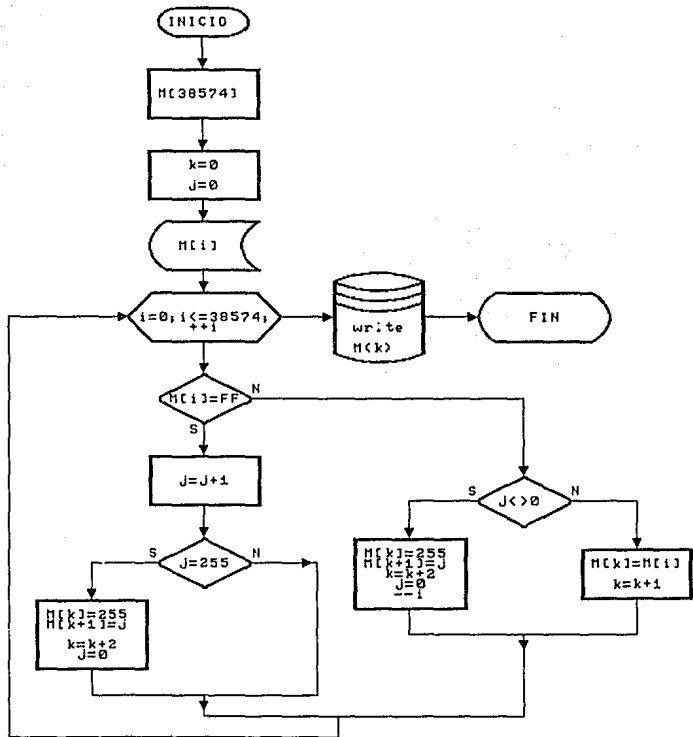


Figura 27: Algoritmo para la reducción de un archivo TIF, formato de PaintShow

compara con FF (hexadecimal) que para este caso es el caracter que más se repite en este tipo de archivos. Si $M[i]$ es FF se incrementa el contador j , el cual es comparado con 255, si es mayor entonces se almacena FF (que es el caracter compactado) y en el siguiente byte también FF (j) que es el número de caracteres compactados; k se incrementa en 2 porque se almacenaron 2 bytes y regresa a leer otro caracter. Si no es igual a 255 sólo regresa a leer el siguiente dato.

Cuando el caracter en análisis no es igual a FF se hace la comparación con j que debe ser diferente de cero, ya que si esto es cierto se deben almacenar dos bytes, porque significa que se encontraron "nulos" y que además de almacenar FF también se almacena j . Después el contador j se hace igual a cero porque empieza una nueva comparación, i se decrementa en 1 porque el último valor leído aun no ha sido almacenado o tomado en cuenta. Por el contrario si no es diferente de cero (j) solo se almacena el caracter leído y entonces k solo se incrementa en 1. En ambos casos se regresa a leer otro caracter.

Una vez que se han leído o analizado cada uno de los caracteres se termina el ciclo y el nuevo arreglo $M[k]$ es grabado en un archivo nuevo; resultando este más pequeño (entre un rango de 3 y 20 Kbytes)

Con respecto al algoritmo original se pueden notar dos cambios importantes:

1. En la comparación de j con 255, el "mayor que" ($>$) se cambia por "igual" ($=$) porque solo se puede representar el 255 (FF) y no el 256 que resultaría si se dejara el primer símbolo ($>$).
2. En la comparación de j con 0 se puso "diferente" ($<>$) para simplificar el segundo programa (visualización de la imagen utilizando este archivo comprimido y que a continuación se enuncia).

PROCESO PARA MOSTRAR LA IMAGEN

Para este algoritmo (Figura 28) se da un tamaño de arreglo de 20 KB. Se inicializa el contador c que es la posición relativa de la memoria de video en la cual se va a mostrar el caracter leído.

El primer ciclo es realizado para conocer la longitud del archivo (ya que éste va a variar) almacenada en la variable tam que es utilizada para el segundo ciclo. Se toma un byte ($M[i]$) y se compara con FF; si este es igual, j toma el valor de $M[i + 1]$ porque es la cantidad de caracteres compactados y por lo tanto es el número de veces que FF será desplegado en pantalla, una vez que termina el despliegue, c toma el valor de k y regresa a tomar otro dato.

Si $M[i]$ no es igual a FF, solo es desplegado en pantalla, se incrementa c y se toma otro valor. Al comparar todos los $M[i]$ el proceso termina.¹⁵

IMPLEMENTACION DEL SISTEMA

Después de las diversas fases a través de los que se pasó se llega a un sistema en el cual se utilizan las mejoras que se enuncian en el apartado anterior. Los pasos a seguir son:

- » Digitalizar Imagen¹⁶
- » Altas
- » Bajas
- » Cambios
- » Compactar Imagen
- » Consulta de Información

15 Todos los programas realizados se encuentran en el anexo.

16 La digitalización y compactación de la imagen son dos pasos que pueden realizarse antes o después de la captura o corrección de los datos, debido a que son procesos independientes pero cuyo identificador de la imagen debe corresponder.

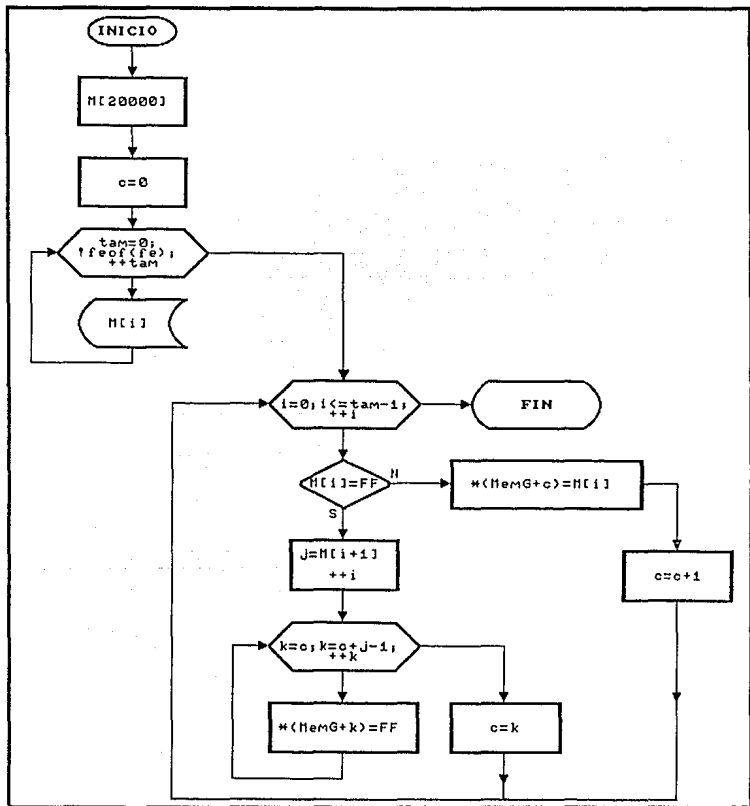
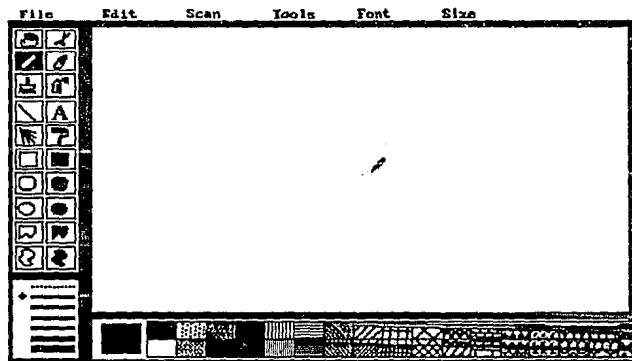


Figura 28: Algoritmo para mostrar la imagen a partir del archivo compactado.

La digitalización de las imágenes que utiliza el sistema es un paso, que por comodidad, debe ser realizado antes de entrar al menú del programa principal¹⁷, ya que si no existen estos archivos no se obtienen resultados al usar la opción Compactar Imagen. De otra manera, quizás se realice primero la captura, después salir del programa para digitalizar y posterior a esto entrar de nuevo.

Para digitalizar la imagen, como ya se apuntó en capítulos anteriores, se utiliza el paquete PaintShow en conjunto con el scanner ScanMan de Logitech. Para hacer uso de ello debe estar cargado en memoria el manejador (driver) del mouse y tener conectado y activado el scanner, estando en el subdirectorio que se encuentra el paquete (generalmente PSPLUS) se tecléa PAINT que es un archivo (BAT) que ejecuta el programa PSPLUS con sus parámetros necesarios. Apareciendo la pantalla de la siguiente manera:



17 En este programa principal sólo se incluyen las últimas cinco opciones, debido a que la primera es un proceso completamente a parte.

Se lleva el cursor a la opción SCAN y reteniendo el botón se elige Setup Scan, esto permite fijar las dimensiones de la imagen a digitalizar (ancho y largo) así como su resolución (DPI). Después se utiliza el subcomando Scan Picture del mismo comando, al hacerlo el scanner se pone en modo de lectura (se enciende la luz roja), indicando que está listo para digitalizar. Para iniciar el proceso, la ventana de lectura es colocada al inicio de la imagen, se presiona el botón del costado izquierdo (Scan Button) y reteniéndolo se desliza hasta finalizar la imagen. Conforme el scanner va leyendo, la imagen se va mostrando en la pantalla.

Para grabarla, se usa la opción Save Picture del comando FILE. Se escribe el nombre y se elige sin compresión (none) con lo que se finaliza la digitalización. Una vez que los archivos de imagen ya están listos, se entra al menú principal (MENU).

Para la captura de los datos se utiliza un programa ejecutable llamado CAPTURA, el cual muestra la siguiente pantalla:

C A P T U R A	
IMAGEN	: ____
DESCRIPCION:	_____

En él se pide el nombre del archivo de imagen (sin extensión)¹⁸ si este existe sale un mensaje que lo indica, sino se continua con la captura en cuatro renglones disponibles para la descripción del equipo, si los datos no están correctos

¹⁸ Es conveniente aclarar que dicho nombre deberá ser el mismo que tendrá al ser digitalizada la imagen.

se pueden corregir antes de grabar, o bien evitar que sean grabados aunque estén correctos.

El registro que es grabado en la captura puede en un momento dado ser eliminado, basta con escribir el identificador del archivo que contiene la imagen, desplegándose automáticamente la descripción correspondiente o bien un mensaje si no existe, teniendo la opción de evitar dicha baja si se responde con 'N' a la pregunta ¿desea borrarlo?

Finalmente, también puede ser corregida o modificada la información capturada. Del mismo modo que la opción anterior, el registro es localizado a través del identificador de la imagen, el cual también puede ser corregido al igual que cualquier renglón de descripción o también indicar que el registro no existe si ese es el caso.

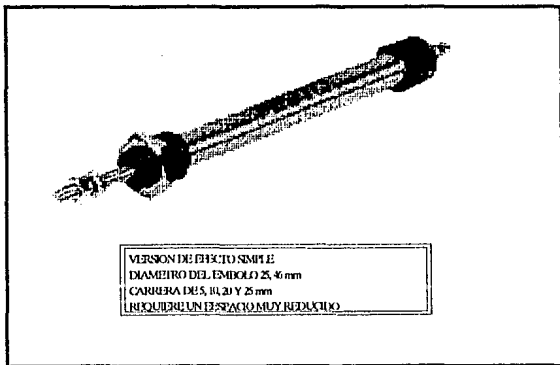
Antes de poder realizar la consulta, una vez que se tienen capturados los datos, se deben compactar los archivos de imagen para que puedan ser incluidos correctamente.

El programa pide el nombre del archivo a compactar, sin extensión, ya que automáticamente se asume que se trata de un archivo con formato TIFF; si dicho archivo no existe, la pantalla despliega el mensaje para indicarlo al usuario; de lo contrario se genera un archivo más pequeño que el original, cuyo nombre será el mismo pero con extensión PAC. Aquí se debe recordar que se está utilizando el método Eliminación de Nulos o Blancos para compactar dicho archivo; es decir, la supresión de blancos para reducirlo. No importa que un archivo sea compactado más de una vez (archivo TIFF) ya que el archivo final será el mismo porque se sobrescribe.

Finalmente, se puede realizar una consulta de la información, en la que se pide al igual que en otras opciones el nombre del identificador -o la clave, si así se prefiere- correspondiente a la pieza que se quiere visualizar.

El programa localiza el registro indicado para que se generen dos archivos, uno que contiene el identificador y otro los datos; estos archivos son

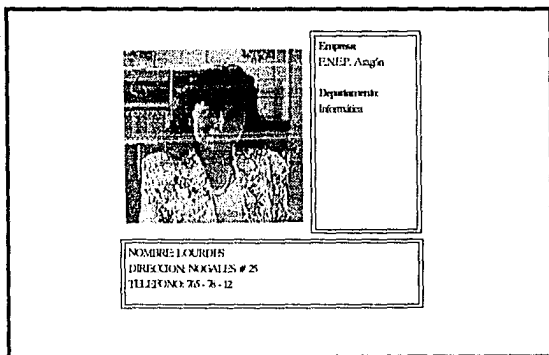
utilizados por el programa que muestra la pieza¹⁹. Es decir, se lee el archivo del identificador asumiendo por default la extensión PAC, si el archivo no existe es indicado a través de la pantalla, de la misma manera sucede con el archivo de datos (txt); pero si todo es correcto, la información (imagen y datos) es desplegada de una manera rápida, como se muestra en la siguiente figura.



Cabe señalar que en el programa de despliegue, se va almacenando la imagen, según se va descomprimiendo, a partir del segmento A000H que es la que utiliza la tarjeta VGA, y así la imagen mostrarse con mejor calidad.

¹⁹ Se habla de la pieza porque se está trabajando con el sistema que muestra información de equipo neumático.

Si el sistema es un banco de datos de personal, la pantalla mostraría:



The screenshot displays a personnel record interface. On the left is a black and white photograph of a man. To the right of the photo is a box containing the following text: "Empresa: ENEP. Aragón" and "Departamento: Informática". Below the photo and the company information is another box containing the text: "NOMBRE: COURTES", "DIRECCION NOGALES # 25", and "TELEFONO: 765-76-12".

COSTO DEL SISTEMA

El realizar un proyecto implica el cuestionar su costo y utilidad, si éste ofrece la solución a un problema reduciendo costos, el proyecto resulta rentable. Para este caso se puede hablar por ejemplo de dos utilidades: una es el tener un banco de información de los empleados de una empresa. Otra, es que en un comercio donde se puede tener un catálogo de herramientas, equipo o bien mercancía con la que se cuenta; esto permitiría en cierto momento el "ganar" clientes ya que si por ejemplo un cliente llega y necesita comprar X pieza pero no recuerda el nombre y solo la conoce físicamente a través de la pantalla la puede reconocer y entonces adquirirla.

Al hablar del costo del sistema se hace referencia a la inversión que se tendría que hacer para utilizarlo, considerando como requerimientos básicos:

CPU 386 1M de RAM	
Monitor VGA	
Disco Duro	1400
Scanner Logitech	300
Mouse	100
Total	1800 Dolares

que no es realmente una inversión costosa si tomamos en cuenta que en la mayoría de los establecimientos se cuenta por lo menos con una PC de este tipo que es lo más costoso de estas herramientas.

Además el sistema tiene una característica muy importante que es la de compactar el archivo de imagen y tomar éste último para mostrarlo obteniendo así una reducción en:

- Espacio en disco
- Tiempo de Acceso

esto es, se lee más rápido un archivo pequeño que uno más grande. Esto dos puntos a su vez reducen costos.

Finalmente, este sistema puede ser adaptado a sistemas multiusuarios, al contarse ya con software de comunicación entre estos sistemas y PC se puede hacer las transmisión tanto de los archivos de imágenes como de los programas haciendo algunas adecuaciones; ahorrándose aquí además de espacio en disco, tiempo en proceso.

CAPITULO V. Evolución de los Sistemas

No siempre los sistemas de información han sido como se conocen en la actualidad, sino que a través de los años el hombre ha propiciado cambios para resolver sus necesidades de la mejor manera posible y con mayor eficiencia.

Desde épocas remotas el hombre ha tenido la necesidad de procesar datos, y en un principio lo hizo de manera muy rudimentaria. Haciendo uso de su ingenio ha producido y utilizado información de manera eficaz, organizándola y teniéndola lista cuando la requiere. Un punto fundamental para que existan progresos en los sistemas es el avance en los procesos de datos y en la tecnología.

Es así, que de los procesos manuales se pasa al procesamiento de datos electromecánico y posteriormente, al uso de las computadoras, para la solución de estos problemas.

Actualmente se emplean sistemas que utilizan archivos tradicionales (secuenciales, directos, indexados) para el manejo de grandes cantidades de datos, es decir, sistemas que hace algunos años fueron realizados en lenguajes (cobol es un ejemplo palpable) que en ese tiempo eran lo más potente para determinadas tareas. Estos procesos aún continúan vigentes porque fueron diseñados para un equipo en especial, el cambiarlos implicaría una nueva inversión económica que algunas empresas o asociaciones no pueden solventar.

Estos sistemas han sido creados también en lenguajes que al tener un diseño estructurado son de más fácil manejo y por esto implica la utilización de menos tiempo porque los programas son más pequeños.

Posterior a esto, vienen los manejadores de bases de datos tanto para PC como para sistemas multiusuarios; algunos de ellos permiten generar reportes sencillos utilizando directamente sus comandos; sin embargo también permiten la programación. Actualmente se tienen generadores llamados de "cuarta generación" los cuales proporcionan al usuario herramientas para producir trabajos que requieren mayor grado de dificultad.

Muchos de los programas que se realizan hoy en día para grandes volúmenes de información utilizan manejadores de bases de datos, debido a que facilitan en gran medida el acceso a la información. El generar un sistema con esta herramienta es más sencillo que si se genera con un lenguaje, sin embargo si este sistema es muy sofisticado y sus necesidades no son resueltas por el manejador, se tiene que recurrir a ellos.

Al ir surgiendo necesidades, se va implementando cada vez más software, de tal manera que los datos proporcionen al usuario la mejor información posible. Por ejemplo, con los primeros sistemas de bases de datos se podía obtener información a manera de texto, actualmente haciendo uso de esa misma base se pueden obtener gráficas que posiblemente (en algunos casos) representan mejor la realidad.

Se han creado bancos de información en varias áreas, por ejemplo:

- De información hemerográfica en donde se tienen asuntos difundidos por seis periódicos que se distribuyen en la Ciudad de México (Excélsior, El Universal, El Nacional, El Financiero, UnomasUno y La Jornada), en el que se incluyen referencias de notas informativas y de opinión de dichos diarios. Sistematización de los contenidos de manera cronológica, temática y por autor. Se consideran y ordenan los temas diversos como política, economía, derecho, educación, ciencia, cultura, sociedad, comunicación, desastres y deportes.
- También se tienen sistemas con disposiciones legales publicadas en el diario oficial en orden cronológico, geográfico y temático, por secretaría o por dependencia de origen, o por norma jurídica.
- Sistemas de información bibliográfica. Hoy en día la computación ofrece grandes ventajas en el manejo de la información documental, las bibliotecas cada vez más utilizan esta tecnología para realizar sus tareas diarias.

La implementación de estos sistemas surge con el fin de agilizar el registro, control y recuperación de las colecciones con las que cuenta determinada biblioteca, así como la ampliación de los servicios que prestan a los usuarios para permitir la consulta a los bancos de información.

SISTEMAS RECIENTES

Actualmente existen sistemas en los cuales se utiliza el lector óptico de código de barras, con esto se evita el cometer errores al ser ingresado el código. Por ejemplo, en algunos eventos se entregan gafetes a los participantes en donde además de estar registrados sus datos personales es anexado un código de barras, el cual se utiliza para ser leído y automáticamente registrar la asistencia de dicha persona, agilizando como consecuencia esta tarea. También pueden ser utilizados en almacenes para llevar una mejor contabilidad de las entradas y salidas de productos.

Se hace uso de sistemas con bancos de imágenes, las cuales pueden ser integradas a los datos, textos y gráficas existentes. Teniendo, de esta manera, la información respaldada que estando archivada puede sufrir deterioros.

La tendencia es el manejo de multimedia, de tal manera que para presentar información ésta es tomada a través de una cámara de video y al combinarse con algún paquete en especial y eligiendo una opción, la información es presentada como un video o bien combinando texto e imágenes. Por ejemplo, sistemas de información de censos, relaciona la ubicación geográfica de un lugar específico con los datos estadísticos inherentes al mismo.

Otro ejemplo de sistema que podemos mencionar en la actualidad, es el que se utiliza en la obtención de las licencias, la fotografía es capturada a través de una cámara de video conectada a la computadora y posteriormente impresa a partir de ella.

Todos estos sistemas hacen referencia a equipos PC, sin embargo cada vez son más utilizados los equipos multiusuarios apoyados en el sistema UNIX, siendo una de sus principales características el tomar a los periféricos como cualquier

archivo o bien el tener todo en memoria y así realizar los procesos más rápidamente, creándose cada vez más software para este tipo de equipos.

Se observa entonces que los cambios y progresos se van dando con mayor frecuencia, teniendo como finalidad hacer que los sistemas tengan mayor confiabilidad y además, presentándolos al usuario final con la mayor información posible.

CONCLUSIONES

Como se mencionó en el último capítulo, el hombre siempre ha requerido una alternativa para organizar y clasificar información, de tal manera que pueda utilizarla en el momento que la requiera.

Así entonces, los avances tecnológicos han sido tomados para tener mejores y más eficientes sistemas de información. Se puede observar que la tendencia es cambiar de: información almacenada en archiveros a información almacenada en cintas, discos, discos compactos, cartuchos, etc.; reduciendo con este cambio espacio físico de almacenamiento y por ende reduciendo costos, que en la actualidad es un punto fundamental en la economía de toda organización.

El desarrollo de este proyecto es interesante debido a que es algo que actualmente se utiliza. Algunos años atrás, los sistemas de información eran realizados utilizando (sólo) archivos tradicionales como son los secuenciales, directos e indexados; seguido a esto se utilizaron y se siguen utilizando los manejadores de bases de datos que permiten una mayor flexibilidad al manejar los datos.

Tomando en cuenta todo esto se pensó en un sistema que tuviera algo más que solo datos tradicionales como lo es una imagen, siendo esto de utilidad para empresas; entre otras cosas para el registro de su personal, o para la presentación de catálogos.

Sin embargo, esto último no significa que no se siga haciendo uso de archivos tradicionales porque estos siguen vigentes y será así durante mucho tiempo. El haber utilizado un manejador de datos como el Dbase es importante porque es un sistema comercial, permite una manera fácil de manejar datos y posibilita hacer llamadas a programas externos, de tal manera que estas rutinas se pueden seguir usando en sistemas posteriores.

Un punto muy importante o quizás el más importante, la inclusión y el desarrollo de un programa de comprensión porque siempre en toda empresa, asociación o cualquier organismo la economía -como ya se mencionó anteriormente-

es un punto fundamental, por lo cual el almacenamiento resulta algo trascendental, ya que siempre se tratará de utilizar el menor espacio posible.

Seguramente surgirá la pregunta del por qué considerarlo tan importante, existiendo actualmente paquetes comerciales que realizan este trabajo; la respuesta es que se necesitaba un archivo compactado pero que se conociera cómo estaba conformado para ser utilizado por el programa, de tal manera que el sistema hiciera uso de este archivo para poderlo mostrar. Haciendo una comparación con un paquete de compactación la diferencia fue mínima; teniendo que para una imagen el paquete de compactación lo dejaba sólo 300 bytes más pequeño (aproximadamente).

Este sistema da la posibilidad de poder ser modificado tomando esta idea, puesto que va surgiendo software y hardware cada vez más avanzado, como son los manejadores de bases de datos, lenguajes, etc, en lo que respecta al primero; y scanners refiriéndome a lo segundo. Pero es importante proporcionar las bases de como realizar software de investigación que permita entrelazarlo con el software comercial para formar uno propio que promueva desarrollo tecnológico específico.

En términos generales los objetivos planteados al inicio, han sido alcanzados de una manera satisfactoria porque se logró obtener el sistema que se pretendía y porque se cubrieron los puntos para llegar a él como fueron:

Utilización de la digitalización para la obtención de imágenes; manipulación de los archivos digitalizados para su uso adecuado; compactación de estos archivos y finalmente la implementación del sistema.

BIBLIOGRAFIA

Libros:

1. "THE HAND SCANNER. Handbook", David D. Bruschi, PC Edition, 1993.
2. "DISEÑO DE SISTEMAS DE INFORMACION", John G.Burch y Gary Grudnitski, Grupo Noriega Editores. Megabyte
3. "DATA COMPRESSION. Techniques and Applications Hardware and Software Considerations", Gilbert Held, John Wiley & Sons
4. "TURBO PASCAL. Biblioteca de Programas", Kris Jamsa y Steven Nameroff, McGraw-Hill
5. "APLIQUE EL DBASE III PLUS", Edward Jones, Ed. McGraw-Hill
6. "IMAGE PROCESSING SYSTEM ARCHITECTURES", Josef Kittler y Michael J. B. Duff, Research Studies Press
7. "SCANMAN HAND-HELD SCANNER USER'S MANUAL", Logitech
8. "CONCEPTOS DE LOS SISTEMAS DE INFORMACION PARA LA ADMINISTRACION". Henry C. Lucas, Jr. McGraw-Hill
9. "PRINCIPIOS DE SISTEMAS DE INFORMACION", George M. Scott, McGraw-Hill

Enciclopedias:

10. "ENCICLOPEDIA Mc GRAW - HILL de la PROGRAMACION DE MICROCOMPUTADORAS. Lenguajes y Sistemas Operativos", William J. Birnes, Tomo I
11. "EL MUNDO DE LA COMPUTACION", Océano, Tomo II

Revistas:

12. "BYTE", Diciembre 1989. Volumen 14, Número 13, 243-256 pp.
13. "RADIO-GRAFICA" Electrónica Popular.No. 211, Julio-Agosto de 1991, 60-61 pp.

ANEXO

PROGRAMA DE CONSULTA EN BASIC

```
10 CLS
20 PRINT "Número de Empleado: ";LOF(1)/128;
30 INPUT "",N
40 NR = LOF(1)/128
50 IF N > NR OR N <= 0 THEN GO TO 10
60 GET #1,N
70 N$ = NOM$
80 D$ = DIR$
90 T$ = TEL$
100 F$ = FOTOS
110 SCREEN 2
120 BLOAD F$ + ".PIC"
130 LOCATE 16,30 : PRINT "NOMBRE : ";N$
140 LOCATE 18,30 : PRINT "DIRECCION : ";D$
150 LOCATE 20,30 : PRINT "TELEFONO : ";T$
160 IF INKEY$="" THEN 160
170 SCREEN 0
180 END
```

PROGRAMA PONDATOS

```
10 OPEN "ENT.DAT" FOR INPUT AS #1
20 INPUT #1,P$
30 INPUT #1,D1$
40 INPUT #1,D2$
50 INPUT #1,D3$
60 INPUT #1,D4$
120 LOCATE 21,1 : PRINT D1$
130 LOCATE 22,1 : PRINT D2$
140 LOCATE 23,1 : PRINT D3$
170 END
```

PROGRAMA PONIMAG

```
5 DEF SEG = &HA000
6 SHELL "ALTA"
10 OPEN "FIG.DAT" FOR INPUT AS #1
15 READ F$
70 BLOAD F$ + ".BIN",0
100 SHELL "MODE 80"
120 END
```

PROGRAMA DE CONSULTA (Utilizando Bases de Datos)

```

SET TALK OFF
SET STATUS OFF
SET SAFETY OFF
!VIDPOP
CLEAR
@ 0,0
CALL CLIPPPOP WITH "PANTALLA"
SET COLOR TO W/N,N/W
I=1
WAIT "OPRIMA CUALQUIER TECLA PARA CONTINUAR"
FIG=' '
R=' '
USE PIEZAS
GO TOP
RESP=.T.
DO WHILE RESP
  CLEAR
  CALL CLIPPPOP WITH "SUBP2"
  SET COLOR TO W/N,W/N
  @ 6,37 GET FIG PICTURE "A#"
  READ
  SET COLOR TO N/W,W/N
  LOCATE FOR PIEZA = FIG
  COPY TO ENT.DAT NEXT 1 DELIMITED WITH
  LOCATE FOR PIEZA = FIG
  COPY TO ENT.DAT NEXT 1 DELIMITED WITH "
  LOCATE FOR PIEZA = FIG
  COPY TO FIG.DAT FIELDS PIEZA FOR PIEZA = FIG DELIMITED WITH BLANK
  LOCATE FOR PIEZA = FIG
  !ALTA
  RUN PONDATOS
  RUN PONIMAG
  SET COLOR TO W/N,W/N
  I=1
  WAIT " "
  !MODE 80
  CALL CLIPPPOP WITH "SUBP1/"
  SET CLOR TO W+/B,W+/B
  @ 20,75 GET R PICTURE "A"
  READ
  SET COLOR TO W/N,N/W
  IF UPPER(R)='N'
    RESP=.F.
  ENDF
ENDDO
!MODE 80
SET SAFETY ON
SET TALK ON
SET STATUS ON
CLOSE DATABASE

```

PROGRAMA DE REDUCCION DE UN ARCHIVO TIFF

```
# include <stdio.h>
# include <dos.h>

# define BUF 38574

FILE *fc;
FILE *fs;
unsigned char M[BUF];
unsigned int i,j,k;

main(int argc, char *argv[])
{
    if (argc == 3)
    {
        printf("\n Uso: compac Arch.Fic Arch.Des");
        exit(1);
    }
    fc = fopen(argv[1], "rb");
    if (fc == NULL)
    {
        printf("\n No se puede abrir el archivo %s,argv[1];", argv[1]);
        exit(1);
    }
    fs = fopen(argv[2], "wb");
    if (fs == NULL)
    {
        printf("\n No se puede abrir el archivo %s,argv[2];", argv[2]);
        exit(1);
    }
    fread (M, sizeof(M), 1, fc);
    k = 0;
    j = 0;
    for (i = 174; i <= BUF - 1; i++)
    {
        if (M[i] == (unsigned char) 255)
        {
            ++j;
            if (j == 255)
            {
                M[k] = 255;
                M[k + 1] = j;
                k = k + 2;
                j = 0;
            }
        }
    }
}
```



```

else
{
    if (j!=0)
    {
        M[k]=255;
        M[k+1]=j;
        k=k+2;
        -i;
        j=0;
    }
    else
    {
        M[k]=M[i];
        ++k;
    }
}
fwrite(M,k,1,fs);
fclose(fe);
fclose(fs);
}

```

PROGRAMA PARA MOSTRAR IMAGEN EN PANTALLA A PARTIR DEL ARCHIVO REDUCIDO

```

# include <stdio.h>
# include <io.h>
# include <dos.h>

# define BUFF 20000

FILE *fc;
unsigned char M[BUFF];
unsigned char huge *MemG;
unsigned int c,i,j,k,tam;

main(int argc, char *argv[])
{
    if (argc!=2)
    {
        printf ("\nUso fsize Nombre_de_Archivo.");
        exit(1);
    }
    fc = fopen (argv[1],'rb');
    if (fc == NULL)
    {
        printf ("\nNo se puede abrir %s.",argv[1]);
        exit (1);
    }
    for (tam=0;!feof(fc); ++tam) fread (&M[tam],1,1,fc);
    -tam;
    _AH = 0x00;
}

```

```

_AL = 0X11;
geninterrupt (0x10);
MemG = (unsigned char huge *) 0xA0000000;
c=0;
for (i=0;i<=tam-1;+ +i)
{
    if (M[i] == (unsigned char) 255)
    {
        j=M[i+1];
        + +i;
        for (k=c;k<=c+j-1;+ +k)
            *(MemG+k) = 255;
        c=k;
    }
    else
    {
        *(MemG+c)=M[i];
        c=c+1;
    }
}
_AH = 0x00;
_AL = 0x02;
getch ();
geninterrupt (0x10);
fclose (fc);
}

```

Los programas siguientes son muy semejantes a los tres anteriores, excepto que han sido adecuados para ser incorporados al último sistema. Esto es para que se puedan trabajar en conjunto.

PROGRAMA MODIFICADO DE COMPACTACION PARA ADECUARSE AL SISTEMA FINAL

```

#include <stdio.h>
#include <dos.h>
#include <conio.h>
#define BUF 38574

FILE *fc;
FILE *fs;
char arche[6],archs[6];
char exte[6],exts[6];
unsigned char M[BUF];
unsigned int i,j,k;
char tecla;

main()
{
    clrscr();
    textmode(C80);
    textbackground(1);
    textcolor(15);

```

```

gotoxy(7,7);
cprintf("-----| if("");
gotoxy(7,8);
cprintf("|| NOMBRE DE ARCH A COMPACTAR: ");
gets(arche);
gotoxy(44,8);
cprintf("||");
gotoxy(7,9);
cprintf("|| NOMBRE DE ARCH SALIDA: ");
strcpy(archs,arche);
gotoxy(44,9);
cprintf("||");
gotoxy(7,10);
cprintf("-----| if("");
strcpy(exte, ".tif");
strcpy(exts, ".pac");
strcpy(archs,exts);
gotoxy(31,9);
cprintf("%s",archs);
fs = fopen(archs,"wb");
if (fs == NULL)
{
printf("\n No se puede abrir el archivo %s",archs);
exit(1);
}
fread (M,sizeof(M),1,fe);
k = 0;
j = 0;
for (i = 174; i + +)
{
if (M[i] == (unsigned char) 255)
{
+ + j;
if (j == 255)
{
M[k] = 255;
M[k + 1] = j;
k = k + 2;
j = 0;
}
}
else
{
if (j != 0)

```

```

    {
        M[k] = 255;
        M[k + 1] = j;
        k = k + 2;
        --i;
        j = 0;
    }
    else
    {
        M[k] = M[i];
        + +k;
    }
}
}
write(M,k,1,fs);
fclose(fe);
fclose(fs);
gotoxy(7,12);
cprintf("\n Proceso Terminado");
tecla = getche();
}

```

PROGRAMA PARA DESPLEGAR IMAGEN ADECUADO AL SISTEMA FINAL

```

# include <stdio.h>
# include <io.h>
# include <dos.h>
# include <conio.h>

# define BUFF 20000

FILE *fa;
FILE *fc;
FILE *fd;
char nom[3];
char exte[7];
char dato[52];
int ren;
unsigned char M[BUFF];
unsigned char huge *MemG;
unsigned int c,i,j,k,tam;

main()
{
    fa = fopen ("fig.dat","r");
    fgets (nom, 3, fa);
    fclose (fa);
    strcpy(exte, ".pac");
    strcat(nom,exte);
    fe = fopen (nom,"rb");
    if (fe == NULL)

```

```

{
    printf("\ No se puede abrir el archivo %s",nom);
    exit(1);
}
for (tam=0;!feof(fc);
    ++tam;
fread (&M[tam],1,1,fc);
--tam;
__AH = 0x00;
__AL = 0x11;
geninterrupt (0x10);
MemG = (unsigned char huge *) 0xA0000000;
c=0;
for (j=0;j      am-1;+ + i)
{
    if (M[i] == (unsigned char) 255)
    {
        j = M[i + 1];
        + + i;
        for (k = c;k ;+ + k) *(MemG + k) = 255;
        c = k;
    }
    else
    {
        *(MemG + c) = M[i];
        c = c + 1;
    }
}
fclose (fc);
fd = fopen("ent.dat","r");
textcolor(15);
gotoxy(5,20);
textbackground(1);
printf("-----");
ren = 21;
while(!feof(fd))
{
    fgets(dato, 52, fd);
    gotoxy(5,ren);
    printf(" | | %s",dato);
    gotoxy(60,ren);
    printf(" | | ");
    ren + +;
}
ren--;
gotoxy(5,ren);
printf("-----");
fclose (fd);
scanf("%c",&stdin);
}

```

```

PROGRAMA DE ALTAS
SET TALK OFF
SET STATUS OFF
SET SAFETY OFF
USE PIEZAS1
STORE 'S' TO RESP,RESP1,RESP2
! VIDPOP
DO WHILE RESP1 = 'S'
  RESP3 = 'N'
  pieza1 = ' '
  STORE SPACE(51) TO D1,D2,D3
  STORE SPACE(50) TO D4
  @ 0,0
  CALL CLIPPPOP WITH 'CAP?'
  SET COLOR TO W/N,W/B
  @ 10,19 GET PIEZA1
  READ
  LOCATE FOR PIEZA = PIEZA1
  IF PIEZA = PIEZA1
    @ 20,19 SAY 'LA PIEZA YA EXISTE'
  ELSE
    @ 12,19 GET D1
    READ
    @ 13,19 GET D2
    READ
    @ 14,19 GET D3
    READ
    @ 15,19 GET D4
    READ
    @ 20,19 SAY ' ESTA CORRECTO ? ' GET RESP3
    READ
    DO WHILE RESP3 = 'N'
      @ 0,0
      CALL CLIPPPOP WITH 'CAP?'
      SET COLOR TO W/N,W/B
      @ 10,19 GET PIEZA1
      @ 12,19 GET D1
      @ 13,19 GET D2
      @ 14,19 GET D3
      @ 15,19 GET D4
      READ
      @ 20,19 SAY ' DESEA GRABAR ? ' GET RESP2
      READ
      IF RESP2 = 'S'
        APPEND BLANK
        REPLACE PIEZA WITH PIEZA1
        REPLACE DATO1 WITH D1
        REPLACE DATO2 WITH D2
        REPLACE DATO3 WITH D3
        REPLACE DATO4 WITH D4
      ENDIF
    ENDWHILE
  ENDIF
ENDWHILE

```

```
@ 20,19 SAY " ESTA CORRECTO ? " GET RESP3
READ
ENDDO
ENDIF
@ 22,19 SAY " CONTINUA CAPTURA ? " GET RESP1
READ
ENDDO
CLOSE DATA
SET COLOR TO W/N,N/W
SET SAFETY ON
SET STATUS ON
SET TALK ON

PROGRAMA DE BAJAS
SET TALK OFF
SET STATUS OFF
SET SAFETY OFF
USE PIEZAS1
GO TOP
STORE 'S' TO RESP,RESP1
! VIDPOP
DO WHILE RESP1='S'
    PIEZA1=' '
    @ 0,0
    CALL CLIPOP WITH "CAP"
    @ 10,19 GET PIEZA1
    READ
    SET COLOR TO W/B,N/W
    LOCATE FOR PIEZA=PIEZA1
    IF PIEZA=PIEZA1
        @ 12,19 SAY DATO1
        @ 13,19 SAY DATO2
        @ 14,19 SAY DATO3
        @ 15,19 SAY DATO4
        @ 20,19 SAY " DESEA BORRARLO ? " GET RESP
        READ
        IF RESP='S'
            DELETE FOR PIEZA=PIEZA1
            PACK
        ENDIF
    ELSE
        @ 20,19 SAY " EL REGISTRO NO EXISTE "
    ENDIF
    @ 22,19 SAY " CONTINUA CON EL PROCESO ? " GET RESP1
    READ
ENDDO
CLOSE DATA SET COLOR TO W/N,N/W
SET SAFETYON
SET STATUSON
SET TALK ON
```

PROGRAMA DE CAMBIOS

```
SET TALK
OFF SET
STATUS OFF
SET SAFETY OFF
USE PIEZA
S1
GO TOP
STORE SPACE(51) TO D1,D2,D3
STORE SPACE(50) TO D4
STORE 'S' TO RESP,RESP1
! VIDPOP
DO WHILE RESP1='S'
PIEZA1=' '
@ 0,0
?? CHR(255)+CHR(255)+'CAP/'
@ 10,19 GET PIEZA1
READ
SET COLOR TO W/B,N/W
LOCATE FOR PIEZA = PIEZA1
IF PIEZA = PIEZA1
@ 12,19 SAY DATO1
@ 13,19 SAY DATO2
@ 14,19 SAY DATO3
@ 15,19 SAY DATO4
@ 20,19 SAY "( DESEA MODIFICARLO ? " GET RESP
READ
IF RESP='S'
?? CHR(255)+CHR(255)+'CAP/'
@ 10,19 GET PIEZA1
@ 12,19 GET D1
@ 13,19 GET D2
@ 14,19 GET D3
@ 15,19 GET D4
READ
@ 20,19 SAY " DESEA GRABARLO ? " GET RESP1
READ
IF RESP1='S'
REPLACE PIEZA WITH PIEZA1
REPLACE DATO1 WITH D1
REPLACE DATO2 WITH D2
REPLACE DATO3 WITH D3
REPLACE DATO4 WITH D4
ENDIF
ENDIF
ELSE
@ 20,19 SAY " EL REGISTRO NO EXISTE "
ENDIF
@ 22,19 SAY "( CONTINUA CON EL PROCESO ? " GET RESP1
READ
ENDDO
```



```

CLOSE DATA
SET COLOR TO W/N,N/W
SET SAFETY ON
SET STATUS ON
SET TALK ON

```

PROGRAMA DE CONSULTA UTILIZANDO EL ARCHIVO DE IMAGEN COMPACTADO

```

SET TALK OFF
SET STATUS OFF
SET SAFETYOFF
IVIDPOP
CLEAR
@ 0,0
CALL CLPPPOP WITH "PANTALLA/"
SET COLOR TO W/N,N/W
WAIT "OPRIMA CUALQUIER TECLA PARA CONTINUAR"
FIG=' '
R=' '
USE PIEZAS1
GO TOP RESP =.T.
DO WHILE RESP
  CLEAR
  CALL CLIPPOP WITH "SUBP2/"
  SET COLOR TO W/N,W/N
  @ 6,37 GET FIG PICTURE "A#"
  READ
  SET COLOR TO N/W,W/N
  LOCATE FOR PIEZA = FIG
  COPY TO ENT.DAT NEXT 4 FIELDS DATO1,DATO2,DATO3,DATO4 SDF
  COPY TO FIG.DAT FIELDS PIEZA FOR PIEZA = FIG DELIMITED WITH BLANK
  LOCATE FOR PIEZA = FIG
  RUN VERIMG
  set color to w/n,w/n
  IMODE 80
  CALL CLIPPOP WITH "SUBP1/"
  SET COLOR TO W +/B,W +/B
  @ 20,75 GET R PICTURE "A"
  READ
  SET COLOR TO W/N,N/W
  IF UPPER(R) = 'N'
    RESP = .F.
  ENDIF
ENDDO
IMODE 80
SET SAFETY ON
SET TALK ON
SET STATUS ON
CLOSE DATA

```