

## REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

**Diseño de salidas de sistemas**

**Autor: Martha Lizzetthe León Rodríguez**

**Tesina presentada para obtener el título de:  
Lic. en Sistemas Computarizados [sic]**

**Nombre del asesor:  
Sergio Barraza Ibarra**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





**UNIVERSIDAD VASCO DE QUIROGA**

**LICENCIATURA EN SISTEMAS COMPUTARIZADOS**

# DISEÑO DE SALIDA DEL SISTEMA

**TESINA**

**QUE PARA OBTENER EL TÍTULO DE  
LICENCIADO EN  
SISTEMAS COMPUTARIZADOS**

**PRESENTA:**

**MARTHA LIZZETTE LEÓN RODRÍGUEZ**



PSU0014Q  
ERDO 952006

**MORELIA, MICH.  
1996**

1996  
IO ZAVALA

T7



UNIVERSIDAD VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

# DISEÑO DE SALIDA DEL SISTEMA

TESINA  
QUE PARA OBTENER EL TÍTULO DE  
LICENCIADO EN  
SISTEMAS COMPUTARIZADOS

PRESENTA:  
MARTHA LIZZETTE LEÓN RODRÍGUEZ

CLAVE 16PSU0014Q  
No. de Acuerdo 952006

MORELIA, MICH.  
1996



En memoria a mi **Abuelto**, el Sr. Jesús Rodríguez,  
que gracias a su apoyo, motivación y consejos hoy  
veo realizado uno de mis grandes sueños.

Al M.A. Ing. Sergio Barraza Ibarra por su  
ayuda brindada en forma desinteresada  
para la elaboración de este trabajo.

A Dios, por haberme prestado vida para permitirme  
concluir mi carrera; por acompañarme en ~~el~~ camino  
de la vida, mostrándome su amor y bondad. Gracias  
pues sin tí nada de esto sería posible.

A mis ~~c~~ **A mis padres:** amigos.

Ing. Eduardo León Rivas  
y  
Q.F.B. Martha E. Rodríguez

Con amor y agradecimiento por su confianza y apoyo.

**A mis hermanos:**

Sandra  
Eduardo  
Marissa

Con cariño

# DISEÑO DE SALIDAS DEL SISTEMA

Al M.A. Ing. Sergio Barraza Ibarra por su  
ayuda brindada en forma desinteresada  
para la elaboración de este trabajo.

PAGINA

INTRODUCCION .....	4
CAPITULO 1 OBJETIVO .....	6
CAPITULO 2 COMO IDENTIFICAR LAS NECESIDADES DE SALIDA DEL SISTEMA .....	7
2.1 Objetivos de la salida .....	8
2.2 Tipos de salidas .....	8
2.3 Aspectos importantes de la salida .....	9
<b>A mis compañeros y amigos.</b>	
CAPITULO 3 COMO PRESENTAR LA INFORMACION .....	11
3.1 Formato tabular .....	11
3.2 Formato gráfico .....	12
3.2.1 Tipos de gráficos .....	12
3.2.2 Cuando utilizar gráficos .....	13
3.2.3 Uso de iconos .....	15
3.3 Presentaciones en color .....	16
CAPITULO 4 DISEÑO DE SALIDA .....	18
4.1 Reportes impresos .....	18
4.1.1 Métodos para la salida impresa .....	19
4.1.2 Formas especiales .....	20
4.1.3 Copias múltiples de la salida .....	21
4.1.4 Documentación de retorno .....	21

# 4.2 DISEÑO DE SALIDAS DEL SISTEMA ..... 22

4.3	.....	23
4.3.1	Encabezados	23
4.3.2	Datos y detalles	24
4.3.3	Resúmenes	24
4.3.4	Lineamientos	24

## INDICE

	PAGINA
INTRODUCCION .....	4
CAPITULO 5 DISEÑO DE SALIDA EN PANTALLA .....	23
CAPITULO 1 OBJETIVO .....	6
CAPITULO 2 COMO IDENTIFICAR LAS NECESIDADES DE SALIDA DEL SISTEMA .....	7
2.1 Objetivos de la salida .....	8
2.2 Tipos de salidas .....	8
2.3 Aspectos importantes de la salida .....	9
BIBLIOGRAFIA .....	34
CAPITULO 3 COMO PRESENTAR LA INFORMACION .....	11
3.1 Formato tabular .....	11
3.2 Formato gráfico .....	12
3.2.1 Tipos de gráficas .....	12
3.2.2 Cuándo utilizar gráficas .....	13
3.2.3 Uso de Iconos .....	15
3.3 Presentaciones en color .....	16
CAPITULO 4 DISEÑO DE SALIDA .....	18
4.1 Reportes impresos .....	18
4.1.1 Métodos para la salida impresa .....	19
4.1.2 Formas especiales .....	20
4.1.3 Copias múltiples de la salida .....	21
4.1.4 Documentos de retorno .....	21

4.2 Desarrollo de una plantilla para la salida impresa .....	22
4.3 Diseño de la salida impresa .....	23
4.3.1 Encabezados .....	23
4.3.2 Datos y detalles .....	24
4.3.3 Resúmenes .....	24
4.3.4 Lineamientos .....	24
<b>CAPITULO 5</b>	
<b>DISEÑO DE SALIDA EN PANTALLA</b> .....	26
5.1 Plantillas para pantallas .....	26
5.2 Diseño de pantallas .....	27
5.2.1 Secciones de una pantalla .....	27
5.2.2 Diseño de pantallas múltiples .....	28
5.2.3 Diseño de ventanas .....	29
5.2.4 Criterios de diseño para ventanas .....	30
<b>CONCLUSIONES</b> .....	32
<b>BIBLIOGRAFIA</b> .....	34
<b>CASO PRACTICO</b> .....	37

El capítulo cuarto presenta el diseño de la salida impresa, los métodos existentes para la salida impresa, las formas especiales, así como la forma preimpresa, las copias múltiples de salida y los documentos de retorno; en este capítulo, también se hace mención al desarrollo de una plantilla para la salida impresa, posteriormente se menciona el diseño en sí de la salida, el cual puede contener encabezados, datos y detalles, además de resúmenes y lineamientos.

## INTRODUCCION

En la quinta unidad se hace mención del desarrollo de las plantillas para la salida en monitor, siguiendo, con el diseño de los pantallas; dentro de este tema se hace referencia a las secciones de la misma.

Se considera que en la elaboración de la tesina necesaria para obtener el preciado título de **Licenciado en Sistemas Computarizados**, intervienen factores muy importantes; una labor de estudio e investigación y nuevas aportaciones, no se pretende indicar que constituyan el descubrimiento precisamente de nuevos conocimientos ni de nuevos métodos, técnicas o sistemas que pudieran engrosar los elementos con que cuenta nuestra profesión para su mejor desempeño, ya que el alumno que elabora este tipo de trabajo, por su calidad de recién egresado, carece de experiencia profesional; pero con la capacidad de análisis y síntesis para integrar un trabajo sobre un tema en especial y que sirva de consulta o influencia para la comunidad universitaria interesada en el área.

En virtud de lo anterior, en el capítulo primero menciono los objetivos que deseo lograr con el presente trabajo, en el segundo capítulo describo la manera de identificar las necesidades de salida de los sistemas de información, así como los objetivos, los tipos de salida y los aspectos a considerar como más importantes de la salida, producto principal para el usuario final; en el capítulo tercero menciono la manera como debe ser presentada la información, considerando para esto los formatos tabular y gráfico, así como las presentaciones en color; dentro de el formato gráfico, se mencionan los tipos de gráficas, cuándo se deben utilizar estas, además el uso de iconos para una mayor apreciación visual.

El capítulo cuarto presenta el diseño de la salida impresa, los métodos existentes para la salida impresa, las formas especiales, así como la forma preimpresa, las copias múltiples de salida y los documentos de retorno; en este capítulo, también se hace mención al desarrollo de una plantilla para la salida impresa, posteriormente se menciona el diseño en sí de la salida, el cual puede contener encabezados, datos y detalles, además de resúmenes y lineamientos.

En la quinta unidad presento el diseño de la salida en pantalla; aquí hago mención del desarrollo de las plantillas para la salida en monitor, siguiendo, con el diseño de las pantallas; dentro de este tema se hace referencia a las secciones de la misma, además presento, el diseño de las pantallas múltiples que facilitan el uso del sistema de información al usuario final, así como el diseño de ventanas, y los criterios bajo los cuales se debe estructurar una ventana.

Al final este trabajo, presento las conclusiones que considero más importantes para el diseño de la salida en los sistemas de información.

## OBJETIVO

Presentar una metodología sistemática y formal a seguir por parte de profesionales en diseño de sistemas de Información, que le permita desarrollar la distribución para presentar información en formatos impresos y en pantalla; definir las especificaciones de la salida de la computadora, así mismo para seleccionar el mejor dispositivo de salida, con el propósito de presentar la información a los usuarios de esta, además de diseños que utilicen colores o gráficas que permitan lograr un mayor impacto visual en el usuario final, así como generar los reportes óptimos para una eficiente toma de decisiones en el nivel directivo de las organizaciones.

El término salida se utiliza para denotar cualquier información producida por un sistema de información, ya sea impresa o en una pantalla. Cuando se diseña la salida se:

- \* Identifica la salida específica que es necesaria para satisfacer los requerimientos de información.
- \* Seleccionar los métodos para presentar la información.
- \* Crear los documentos, reportes u otros formatos que contengan la información producida por el sistema.

Los métodos de la salida varían a través de los sistemas.

## CAPITULO 2

### 2.1 Objetivos de la salida.

La salida de un sistema de información debe alcanzar uno o más de los siguientes objetivos:

### COMO IDENTIFICAR LAS NECESIDADES DE SALIDA DEL SISTEMA.

1. Expresar información (relaciones, personas, estado actual o proyecciones para el futuro).
2. Señalar eventos importantes, oportunidades, advertencias o tendencias.
3. Iniciar una acción.
4. Comparar.

El diseño de la salida de la computadora debe avanzar en una forma organizada y bien pensada: tiene que desarrollarse correctamente mientras que al mismo tiempo se garantice que cada elemento de la salida está diseñado para que las personas encuentren que el sistema es fácil de emplear. No se puede clasificar como buena una salida estéticamente atractiva o que haga uso de una nueva tecnología a menos que satisfaga las necesidades de la organización y de sus usuarios.

El término salida se utiliza para denotar cualquier información producida por un sistema de información, ya sea impresa o en una pantalla. Cuando se diseña la salida se:

\* Identifica la salida específica que es necesaria para satisfacer los requerimientos de información.

\* Seleccionan los métodos para presentar la información.

\* Crean los documentos, reportes u otros formatos que contienen la información producida por el sistema.

1. Un reporte

2. Un documento

3. Un formato

Los métodos de la salida varían a través de los sistemas.

## 2.1 Objetivos de la salida.

La salida de un sistema de información debe alcanzar uno o más de los siguientes objetivos:

1. Expresar información relacionada con actividades pasadas, estado actual o proyecciones para el futuro.
2. Señalar eventos importantes, oportunidades, problemas o advertencias.
3. Iniciar una acción.
4. Confirmar una acción.

## 2.3 Aspectos importantes de la salida.

El buen diseño de la salida de los sistemas, no puede ser desarrollado en forma independiente del uso que se dará a la salida. No se puede clasificar como buena una salida estéticamente atractiva o que haga uso de una nueva tecnología a menos que satisfaga las necesidades de la organización y de sus usuarios.

1. ¿Quiénes recibirán la salida?

El objetivo principal durante el diseño de la salida de la computadora es la información que será presentada a las personas.

2. ¿Qué contenido, formato y medio de presentación?

## 2.2 Tipos de salidas.

La salida, ¿presenta información, solicita una respuesta, o inicia una acción? El uso del Sin importar si la salida es un reporte o un listado del contenido de un archivo, éste siempre es resultado de un proceso por computadora. La salida del sistema puede ser:

La cantidad de datos también sugiere si deben emplear métodos de impresión o

1. Un reporte en una pantalla.
2. Un documento
3. Un mensaje con qué frecuencia es necesaria la salida?

El calendario junto con la oportunidad de la salida son guías específicas de

De acuerdo con las circunstancias y los contenidos, la salida puede ser impresa o presentada en una pantalla.

El contenido de la salida tiene su origen en las siguientes fuentes:

1. Recuperación de un dispositivo de almacenamiento
2. Transmisión desde un proceso o actividad del sistema
3. Directamente desde una fuente de entrada

### 2.3 Aspectos importantes de la salida.

Se debe dar respuesta a cinco preguntas en forma completa y apropiada, que ayudarán a comprender mejor lo que debe ser la salida de un nuevo sistema:

1. ¿Quiénes recibirán la salida?

Quizá los usuarios externos tengan requerimientos específicos que no se puedan cambiar y que dictan los requerimientos de contenido, formato y medio de presentación.

2. ¿Cuál es el uso que se le pretende dar?

La salida, ¿presenta información, solicita una respuesta, o inicia una acción?. El uso determina el contenido, la forma y el medio a utilizarse para su generación.

3. ¿Cuántos detalles son necesarios?

La cantidad de datos también sugiere si deben emplear métodos de impresión o de presentación en una pantalla.

4. ¿Cuándo y con qué frecuencia es necesaria la salida?

El calendario junto con la oportunidad de la salida son guías específicas de

diseño. Algunas salidas se producen con poca frecuencia y sólo cuando aparecen ciertas condiciones.

### 5. ¿Qué método utilizar?

La salida impresa se emplea con mayor frecuencia, sin embargo, si un sistema da respuesta del tipo si o no a las consultas, entonces, a menudo es apropiado presentar la respuesta en una pantalla.

La forma en que se presenta la información determinará si la salida es clara y comprensible, si los detalles son convincentes y si la toma de decisiones se efectúa con mayor rapidez y exactitud.

#### 3.1 Formato tabular.

El formato tabular debe diseñarse bajo las siguientes condiciones:

- a) Cuando los detalles dominan y son necesarios pocos comentarios o explicaciones
- b) Cuando los detalles son presentados en categorías concretas
- c) Cuando cada categoría debe tener una etiqueta
- d) Cuando se deben obtener totales o realizar comparaciones entre diversos componentes

En un formato tabular cierta información es más importante y, por consiguiente, debe resaltar sobre la demás, en general, no debe estar seguro de que los siguientes aspectos sean los que sobresalgan:

- a) Excepciones a las expectativas normales
- b) Categorías más importantes de actividades o entidades
- c) Resúmenes de las categorías o entidades más importantes
- d) Identificación única de la información
- e) Entidades que dependan del tiempo

## CAPITULO 3

Se debe presentar la información que presenta los detalles en un orden significativo, aquel donde éstos sean más sencillos de analizar.

Ordinar respecto que se debe de considerar el aborde el diseño de la salida es estar los **COMO PRESENTAR LA INFORMACION.** seleccionar las características que mejorarán la claridad.

La forma en que se presenta la información determinará si la salida es clara y comprensible, si los detalles son convincentes y si la toma de decisiones se efectúa con mayor rapidez y exactitud.

Las gráficas deben hacer fácil su objetivo con respecto al lector o de lo contrario. Deben contener pocas palabras, pueden mostrarse en pantallas de video, elaborarse con varios colores en impresoras de bajo costo, dibujar en gráficos. El formato tabular debe utilizarse bajo las siguientes condiciones: áreas especiales que pueden conectarse a la computadora. Las gráficas pueden mejorar la presentación

### 3.1 Formato tabular.

- El formato tabular debe utilizarse bajo las siguientes condiciones:
- Cuando los detalles dominan y son necesarios pocos comentarios o explicaciones
  - Cuando los detalles son presentados en categorías discretas
  - Cuando cada categoría debe tener una etiqueta para lo que serán empleadas, así
  - Cuando se deben obtener totales o realizar comparaciones entre diversos componentes.

En un formato tabular cierta información es más importante y, por consiguiente, debe resaltar sobre la demás, en general, se debe estar seguro de que los siguientes aspectos sean los que sobresalgan:

- Gráficas de sectores. Describen partes de un todo que guardan relación con un
  - Excepciones a las expectativas normales
  - Categorías más importantes de actividades o entidades
  - Resúmenes de las categorías o actividades más importantes largo de varios periodos
  - Identificación única de la información objetos sobre la misma gráfica, permite al lector
  - Entidades que dependen del tiempo los demás

**3. Gráficos** Se desea presentar la información en un formato que presente los detalles en un orden significativo, áquel donde éstos sean más sencillos de localizar.

El primer aspecto que se debe de considerar al abordar el diseño de la salida es evitar los datos que no son necesarios. El siguiente paso es seleccionar las características que mejorarán la claridad.

### 3.2 Formato gráfico.

Las gráficas deben hacer notar su objetivo con rapidez al lector o de lo contrario éste perderá el interés. Deben contener pocas palabras, pueden mostrarse en pantallas de video, elaborarse con varios colores en impresoras de bajo costo, dibujar en graficadores o producirse en transparencias de color por medio de cámaras especiales que pueden conectarse a la computadora. Las gráficas pueden mejorar la presentación de la información.

#### 3.2.2 Cuando utilizar gráficas.

El éxito de las gráficas depende de la tarea para la que serán empleadas, así como de la naturaleza de la información presentada.

a) Para mejorar la efectividad de los reportes que como salda se envían a los usuarios que del **3.2.1 Tipos de gráficas.**

b) Para manejar el volumen de información.

c) Para ajustarse a preferencias personales.

1. **Gráficas de sectores.** Describen partes de un todo que guardan relación con un desarrollo o actividad en particular.

Las gráficas son más adecuadas para detectar tendencias en el desempeño de

2. **Gráficas de áreas.** Muestran cambios en en desempeño a lo largo de varios periodos de tiempo. La superposición de varios objetos sobre la misma gráfica, permite al lector ver el cambio de uno en relación con los demás.

**3. Gráficas de barras y escalones.** Muestran cambios en categorías. Sin embargo, más que unir cada punto, miden los puntos dato desde la escala horizontal hasta el nivel apropiado de la escala vertical.

Las gráficas complementan otra información, no la reemplazan; deben hacer notar su objetivo con rapidez al lector o de lo contrario éste perderá el interés. Deben contener pocas palabras.

*Estándares para el diseño de gráficas.*

Las gráficas quizá no ahorren tiempo en la toma de decisiones ni tampoco disminuyan el volumen de información producido por el sistema. Alcanzar estos objetivos depende de los usuarios y de otras características de diseño del sistema de información, como puede ser la forma en que se prepara la información que no es de carácter geográfico. Sin embargo, si se diseñan adecuadamente, las gráficas son complementos excelentes para tablas y reportes escritos.

### 3.2.2 Cuando utilizar gráficas.

Las tablas se emplean por varias razones:

- a) Para mejorar la efectividad de los reportes que como salida se envían a los usuarios que deben recibirlos,
- b) Para manejar el volumen de información,
- c) Para ajustarse a preferencias personales.

*Facilidad para la presentación efectiva de datos.*

Las gráficas son más adecuadas para detectar tendencias en el desempeño de la empresa que los reportes escritos o tabulares. Las comparaciones también son más fáciles de hacer con gráficas que con datos en forma tabular; también facilitan el recordar grandes cantidades de datos asentados en una serie de reportes.

***Manejo del volumen de información.***

La comprensión de grandes cantidades de datos en una gráfica, no disminuye la cantidad de información. El beneficio real de la comprensión es que separa la información en grupos más pequeños, lo que permite recordarlos y comprenderlos con mayor facilidad. La información fragmentada aísla los elementos y facilita su comparación.

***Estándares para el diseño de gráficas.***

Los estándares apoyan el trabajo del diseñador en la preparación de presentaciones gráficas, pero también son de gran ayuda para que el usuario lea los datos.

Toda la salida gráfica contiene texto, cuyo diseño es tan importante como el de los demás aspectos de salida. Cada reporte gráfico, ya sea impreso o presentado en una pantalla, debe incluir un título, así como la fecha en que se preparó. Para toda una serie, es importante añadir números de página.

Dado que el texto toma más tiempo en leerse que las gráficas, es importante el sitio donde se coloque y su legibilidad. Las etiquetas para datos verticales deben colocarse en forma horizontal, teniendo cuidado de evitar detalles en exceso. Un espaciamiento consistente entre todas las etiquetas y el uso del tipo de letra ayudará a aumentar la claridad.

Las etiquetas pueden aumentar la exactitud con la que las personas leen los datos de una gráfica de barras. Sin embargo, la ubicación de las etiquetas afecta la claridad. El valor de los datos debe colocarse en el extremo de la barra.

Todos los ejes horizontales y verticales deben ser proporcionales, al igual que sus etiquetas. Se pueden utilizar tipos en negritas, itálicas o subrayados para recalcar las frases o palabras importantes. El uso exclusivo de mayúsculas en palabras, títulos o

pies de figura extensos, disminuye la claridad y, por tanto, debe evitarse. Tampoco deben utilizarse abreviaturas.

El empleo de la misma tipografía asegura la consistencia de las gráficas y pantallas diseñadas por diferentes analistas. El uso de la tipografía correcta también influye en la legibilidad del texto.

### 3.2.3 Uso de iconos.

Los iconos son la representación gráfica de las entidades descritas por los datos. Hoy los iconos se emplean comúnmente en las interfaces de computadora para representar documentos, cestos, archivos e impresoras.

Los iconos comunican información en forma inmediata cuando son seleccionados en forma apropiada, ya que duplican imágenes con las que los usuarios están familiarizados. Los iconos eliminan la necesidad de que los usuarios aprendan abreviaturas, notaciones o nomenclatura especial. Por otra parte, las mismas imágenes no añaden complejidad a las demás tareas. En contraste, el tiempo necesario para leer las etiquetas o pies de figura por lo común contribuyen a la complejidad. Sin embargo, los iconos apropiados aseguran que las palabras y frases correctas tengan un significado preciso.

Los siguientes lineamientos sugieren cuándo y cómo utilizar los iconos en el diseño de sistemas:

- \* Seleccionar los iconos que serán reconocidos y comprendidos en forma inmediata por los usuarios.
- \* Si no existe ningún icono familiar para determinada situación, emplee etiquetas que eviten la necesidad de que los usuarios aprendan y recuerden símbolos o imágenes

poco familiares.

- \* Utilice el mismo ícono para representar los mismos conceptos a través de varios medios de salida.
- \* Evite el empleo de etiquetas en los íconos; la imagen misma debe comunicar su significado con claridad, sin necesitar de una leyenda o etiqueta.
- \* Utilice una distribución que mantenga el espacio y evite la aglomeración entre íconos.
- \* Mantenga el mismo tamaño entre los diferentes tipos de símbolos a menos que se muestren diferencias en el rendimiento.

### 3.3 Presentaciones en color.

Cada vez son mayores las facilidades de color en sistemas de cómputo grandes y pequeños. Sin embargo, al igual que con las gráficas, el uso inapropiado del color puede ser más un obstáculo que una ayuda para la productividad del usuario y la gerencia. El color debe mejorar, no reemplazar, el buen diseño de la salida.

En general, se recomienda el uso de cuatro o menos colores en un reporte o pantalla. Entre más colores se utilicen, mayor deberá ser la información proporcionada por los datos y figuras. Si la información se presenta en forma visual, entonces menor será su claridad si se emplean cuatro colores que, sólo dos.

El analista debe tener cuidado de mantener la consistencia en el uso del color a través de toda la salida en reportes de un sistema.

Los colores intensos sobre una pantalla recalcan la información más importante. Los colores brillantes incluyen al blanco, el turquesa y el rosa. Los colores oscuros son el magenta, el rojo, el verde y el azul.

Por otra parte, ni el color ni las gráficas mejoran o compensan un diseño pobre.

**Sin embargo, el uso efectivo de las gráficas puede mejorar los resultados que alcanzan los gerentes y usuarios cuando trabajan con la salida del sistema.**

**Las gráficas, los iconos y el empleo del color, pueden aparecer en cualquier tipo de salida de computadora, ya sea impresa o en la pantalla de una terminal.**

Los analistas de sistemas especifican la salida impresa cuando necesitan enviar por correo un documento ya sea para un cliente o proveedor, imprimir un registro de datos o notificar cierta información, o para hacer llegar al mismo tiempo un gran volumen de información a varias personas.

El analista debe buscar el empleo de sólo aquellas salidas impresas que son absolutamente necesarias. Hasta donde sea posible, el desarrollo de un sistema de información basado en computadora debe reducir, no aumentar, el número de reportes impresos que circulan por toda la organización. El proporcionar detalles innecesarios no ayuda a nadie, por lo que los analistas siempre deben estar alertas para evitar la producción de datos extraños.

#### 4.1 Reportes impresos.

Los reportes impresos varían en tamaño aunque los analistas a menudo emplean los siguientes tamaños estándares:

9 1/2 por 11 pulgadas

11 por 14 1/2 pulgadas

8 1/2 por 14 1/2 pulgadas

## CAPITULO 4

### DISEÑO DE SALIDA IMPRESA.

Los analistas de sistemas especifican la salida impresa cuando necesitan enviar por correo un documento ya sea para un cliente o proveedor, imprimir un registro de datos o notificar cierta información, o para hacer llegar al mismo tiempo un gran volumen de información a varias personas.

El analista debe buscar el empleo de sólo aquellas salidas impresas que son absolutamente necesarias. Hasta donde sea posible, el desarrollo de un sistema de información basado en computadora debe reducir, no aumentar, el número de reportes impresos que circulan por toda la organización. El proporcionar detalles innecesarios no ayuda a nadie, por lo que los analistas siempre deben estar alertas para evitar la producción de datos extraños.

#### 4.1 Reportes Impresos.

Los reportes impresos varían en tamaño, aunque los analistas a menudo emplean los siguientes tamaños estándares:

9½ por 11 pulgadas

11 por 14<sup>7/8</sup> pulgadas

8 por 14<sup>7/8</sup> pulgadas

Todos estos tamaños son para formas continuas, que son hojas de papel unidas entre sí y que son alimentadas a la impresora una detrás de otra.

#### **4.1.1 Métodos para la salida impresa.**

Todos los métodos de impresión, ya sean o no de impacto, utilizan una impresora de computadora para colocar la información de salida sobre el papel donde se imprime el reporte. Los analistas de sistemas están empleando cada vez más la impresión por láser para generar la salida impresa. Este método es más eficaz para integrar gráficas y texto.

Uno de los objetivos de diseño de un sistema es el manejo del volumen de información generada. También debe considerarse la salida en película cuando es deseable la información impresa pero es objetable la cantidad de papel necesaria para hacerlo. Todas las características de la salida impresa también se encuentran disponibles en microfilmes o microfichas, que son los dos métodos para producir salida en película.

El tiempo que toma localizar el rollo de película o tarjeta que contiene la información deseada aunado con el necesario para colocar la película en la lectora de microfichas o microfilmes, es la verdadera desventaja de este método.

Entre los procedimientos del sistema que el analista prepara, deben incluirse los necesarios para almacenar y recuperar la salida en películas.

#### 4.1.2 Formas especiales.

Existen dos tipos de formas múltiples:

Los sistemas de información basados en computadora también producen formas especiales. Una de las más comunes es la *forma preimpresa* que está diseñada para incluir símbolos especiales y marcas registradas de la organización, y que se imprime con varios colores, los que dependen de los requerimientos establecidos por el analista o usuario. Las formas se imprimen sobre papel blanco.

Entre las situaciones más representativas donde debe considerarse el uso de formas preimpresas, se encuentran las siguientes:

- \* Reglamentos o requerimientos legales que obligan al uso de formas preimpresas
- \* Destinatarios que esperan un formato estándar
- \* La inclusión del logotipo de la organización, una marca registrada o símbolo que debe estar incluido en la forma
- \* Trabajo artístico o gráficas que tendrán mejor apariencia si se preimprimen.

Las impresoras láser se emplean cada vez más para imprimir tanto los datos y la forma, incluyendo las gráficas, símbolos y líneas asociadas con las formas preimpresas. En el futuro, el color será un aspecto de diseño más común para la salida impresa por computadora.

Cuando los analistas desarrollan el diseño de la salida, tienen la responsabilidad de especificar las partes preimpresas y las generadas por la computadora. Al mismo tiempo, tienen la responsabilidad de conocer o averiguar dónde se deben utilizar formas preimpresas.

### 4.1.3 Copias múltiples de la salida.

Existen dos tipos de forma múltiple:

1. Copias sin papel carbón. Este tipo de copias que no emplean papel carbón son posibles gracias a un recubrimiento químico especial que se encuentra en la parte trasera de cada copia, con excepción de la última. La escritura o impresión de la original se traslada a las copias. El recubrimiento causa que la imagen aparezca en las copias que se encuentran debajo del original.

Las formas que no emplean papel carbón, aunque son más caras que las sí lo utilizan, tienen la ventaja de no requerir de tiempo o equipo extra para retirar el papel carbón de las formas continuas.

2. Copias con papel carbón. Entre cada hoja de papel se coloca papel carbón que sirve sólo una vez.

Dado que estos métodos de copiado dependen de la impresión para producir las copias, resultan adecuados para impresora de impacto. Sin embargo, no son apropiados para impresoras que no son de impacto, como las de inyección de tinta, donde los datos son rociados sobre la forma electrostática.

### 4.1.4 Documentos de retorno.

Si se emplean rastreadores ópticos capaces de leer formas impresas o escritas a mano, entonces el sistema puede preparar una proposición en un documento u otra forma para utilizarla como *documento de retorno*, salida que más adelante regresará como documento de entrada.

## 4.2 Desarrollo de una plantilla para la salida impresa.

Tener la información y los detalles correctos en un reporte, así como la selección del medio de salida apropiado, es esencial. Sin embargo, el diseño de la salida impresa determinará la utilidad que ésta tiene para quien la recibe.

La plantilla de salida es la distribución de objetos en el medio de salida. Cuando los analistas diseñan esta plantilla, construyen un modelo del reporte o documento tal como aparecerá después de que el sistema entre en operación. La plantilla debe mostrar la localización y posición de lo siguiente:

### \* Toda la información variable

Detalles

Resúmenes y totales

Marcas de control

Separadores

### \* Todos los detalles preimpresos

Encabezados

Títulos y nombres del documento

Nombre de la compañía y dirección

Instrucciones

Notas y comentarios

La plantilla es el prototipo que conducirá más adelante, a la elaboración de los programas durante el proceso de desarrollo. Cada variable contenida en la plantilla debe reflejarse en una instrucción de programa si se emplea un lenguaje de tercera generación.

4.3.2 Se pueden utilizar formas especiales que muestran el espaciamiento y las posiciones disponibles para impresión.

### 4.3 Diseño de la salida impresa.

Para comenzar el diseño de una plantilla para reporte, primero es necesario determinar qué datos estarán incluidos en el reporte. El análisis de requerimientos proporciona esta información, y el diccionario de datos contiene la información descriptiva necesaria: el tipo de dato y su longitud.

#### 4.3.1 Encabezados.

Toda salida producida por un sistema de información debe tener un título. Éste es el procedimiento para el diseño de encabezados: colóquese el título del reporte y los encabezados sobre la plantilla empleando para ello las columnas específicas donde se desea que aparezca la información, y céntrese el título. Si se emplea una herramienta automatizada para crear la plantilla, entonces sólo será necesario colocar el cursor en la columna deseada y comenzar a escribir la información apropiada. El número de página proporciona una rápida referencia para los usuarios que trabajan con datos que se encuentran en distintas partes del reporte.

Después de escribir los encabezados, la atención se centra en el contenido del reporte. Antes de marcar los campos, se proporcionan los encabezados de las columnas que se desea que aparezcan impresos sobre cada página. Es buena práctica utilizar un subrayado, guión o cualquier otro símbolo para separar los encabezados de las columnas del comienzo de los datos. Toda columna debe tener un encabezado que describa su contenido. Escriba nombres y palabras y evite las abreviaturas.

### 4.3.2 Datos y detalles.

El procedimiento para diseñar los datos y detalles es el siguiente: proporcionar la descripción de los datos que aparecerán por debajo de los encabezados de columna. Si se emplean puntos decimales, símbolos para moneda u otros especiales, éstos deberán marcarse. Para describir los artículos, algunas organizaciones marcan los caracteres primero y último del campo con un símbolo apropiado y los unen con una línea para mostrar todo el campo. Cualquier forma es aceptable.

Aunque los reportes pueden continuar en varias páginas, basta con definir sólo una vez los detalles de cada línea. La línea ondulada que desciende después de cada dato sobre la hoja que contiene la plantilla de la forma, señala que éste se repite las veces que sea necesario sobre cada página del reporte.

### 4.3.3 Resúmenes.

Algunos diseños de reportes especifican resúmenes de información, columnas de totales o subtotales. El principio para mostrarlos es el siguiente: etiquetar los títulos y encabezados tal como se desea que aparezcan; denotar los datos variables con especificaciones X o 9, de acuerdo con el tipo de campo, y señalar la máxima longitud del campo.

### 4.3.4 Lineamientos.

Existen muchos lineamientos que hacen el trabajo del analista más fácil y, lo más importante, aseguran que los usuarios reciban un reporte claro y comprensible. Ya se han utilizado varios y a continuación se presenta un resumen de ellos:

1. Los reportes y documentos deben estar diseñados para leerse de izquierda a derecha y de arriba hacia abajo.
2. Los datos más importantes deben ser los más fáciles de encontrar.
3. Todas las páginas deben tener un título y un número de página además de mostrar la fecha en que fue preparada la salida.
4. Todas las columnas deben estar etiquetadas.
5. Se deben evitar las abreviaturas.

Cada caso de diseño es propiamente un trabajo menor que trabajar con la mayor parte de la salida impresa. Normalmente, requiere que el usuario escriba instrucciones sobre cómo debe ser el diseño. Con la ayuda de un diseñador, puede suponer que los programadores sabrán cómo hacer todo en un momento, de la vuelta a la siguiente página y que nunca se olvidará el nombre de una columna de un reporte. No es lo mismo hacer las mismas suposiciones cuando se trabaja en la pantalla.

### 3.1 Plantillas para pantallas

Cada página de presentación visual recibe el nombre de pantalla o panel. En este caso, la plantilla muestra el cuadrado vacío. El primer de la pantalla comienza con la verificación de las características de la pantalla de presentación visual. Entre estas se incluyen las siguientes:

- » Dimensiones físicas de la pantalla
- » Número de regiones y columnas de datos que pueden ser mostrados al mismo tiempo
- » Grado de resolución (alta, mediana o baja)
- » Número de colores disponibles (por ejemplo, monocromático, tres colores, ocho colores, etc.)

## CAPITULO 5

### DISEÑO DE SALIDA EN PANTALLA.

Esta clase de dispositivos proporcionan un espacio menor para trabajar que la mayor parte de la salida impresa. Asimismo, requieren que el usuario reciba instrucciones sobre cómo utilizar la pantalla. Con la salida impresa se puede suponer que las personas saben cómo buscar algo en el reporte, dar la vuelta a la siguiente página y qué pasos seguir cuando han terminado con el reporte. No es lo mismo hacer las mismas suposiciones cuando se diseñan las pantallas.

#### 5.1 Plantillas para pantallas.

Cada página de presentación visual recibe el nombre de *pantalla* o *panel*. En este caso, la plantilla facilitará o impedirá su uso. El diseño de la plantilla comienza con la verificación de las características de la pantalla de presentación visual. Entre éstas se incluyen las siguientes:

- » Dimensiones físicas de la pantalla
- » Número de renglones y columnas de datos que pueden ser mostrados al mismo tiempo
- » Grado de resolución (alta, mediana o baja)
- » Número de colores disponibles (por ejemplo, monocromático, tres colores, ocho colores, etc)

- » Métodos de realce (subrayado, negritas, parpadeo, diferentes intensidades)
- » Métodos para el control de la intensidad (alta/baja; normal/inversa)

Las pantallas para presentación visual por lo general tienen 80 columnas con 24 o 25 líneas.

La información debe leerse de izquierda a derecha y de arriba hacia abajo, además de que debe ser fácil encontrar los detalles más importantes. Por otra parte cierta información, como los títulos y encabezados, deben colocarse siempre en forma consistente.

## 5.2 Diseño de pantallas.

El diseño de una pantalla comienza al reconocer que la pantalla está compuesta por diferentes áreas. Las herramientas para elaborar plantillas ayudan al analista a especificar el contenido de los formatos de diseño, ya sean únicos o múltiples.

### 5.2.1 Secciones de una pantalla

Resulta útil dividir la pantalla en secciones que se emplean de manera consistente para presentar información, identificaciones y mensajes para el usuario. La experiencia ha demostrado que la esquina superior izquierda es la parte más visible de la pantalla. El buen diseño de una pantalla hace el mejor uso posible de estas áreas.

Al diseñar pantallas de salida se necesitan áreas para:

- 1) Encabezados y títulos

- 2) El contenido de la pantalla
- 3) Mensajes e instrucciones
- 4) Algunas explicaciones sobre la información que se presenta

Los encabezados y títulos se colocan en la parte superior de la pantalla, los mensajes e instrucciones en la parte inferior, y las explicaciones, si éstas son necesarias, en el cuadrante superior izquierdo.

Las herramientas para la elaboración de plantillas, ya sea de papel o asistidas por computadora, sirven para conducir el diseño de las pantallas. El diseño adecuado de éstas incluye un título y encabezados para las columnas. Los datos en cada columna se indican de la misma manera que para la salida impresa.

La información que indica a los usuarios cómo proseguir en general se muestra en la parte inferior de la pantalla. Se debe señalar a los usuarios cómo recuperar la siguiente pantalla de información y cómo abandonar el sistema. El diseño también puede instruir al programador para que éste escriba el software en una forma que, si el usuario presiona otra tecla y esto puede ser causa de error, el software intercepte la entrada y evite con ello el error.

### 5.2.2 Diseño de pantallas múltiples.

Con frecuencia los diseñadores emplean varias pantallas para dar a los usuarios la información que ellos necesitan. La primera pantalla proporciona información general. Al presionar una tecla específica, el usuario recupera una segunda pantalla que contiene los detalles.

La técnica anterior es muy común en el diseño de pantallas. Permite que los usuarios recorran con rapidez todos los detalles e identifiquen cada artículo para el que

requieren de más información. Al mismo tiempo, la explosión en detalles para un artículo en particular sobre una segunda (incluso una tercera) pantalla, mantiene la facilidad de lectura de la primera al requerir sólo los detalles suficientes para identificar el artículo deseado.

### 5.2.3 Diseño de ventanas.

En la salida de una computadora, las ventanas son subdivisiones de la pantalla que hacen posible presentar al mismo tiempo diferentes conjuntos de salida. Los usuarios pueden ejecutar varios programas al mismo tiempo (característica denominada multitarea) y presentar la salida de cada programa en una ventana. También un programa puede presentar más de una salida al mismo tiempo.

*Capacidades de las ventanas.* De acuerdo con los detalles del sistema, puede ser posible reposicionar las ventanas sobre la pantalla. Los usuarios también desean tener la capacidad de cambiar el tamaño de una ventana, ya sea aumentarlo para incluir en ella más información, o de ocultar las ventanas que en un determinado momento no son necesarias. El *traslape* permite que los usuarios muevan la información a primer plano cuando sea necesario y que la reemplacen de nuevo con otra información al traslaparla, una ventaja que permite utilizar una gran parte de la pantalla sin alterar la flexibilidad ofrecida por las ventanas. Todas estas características deben ser proporcionadas por el sistema operativo de la computadora. De esta forma, el analista de sistemas no diseña la lógica de procesamiento, pero debe discutir la característica, si es que ésta existe, con el usuario tal como se utilizará en la aplicación bajo desarrollo.

El uso de ventanas debe considerarse cuando la aplicación lo requiera o deba ser mejorada con facilidades para:

- » Presentar datos diferentes o conjuntos de reportes al mismo tiempo

- » Para cambiar entre varios programas, mostrando la salida de cada uno de ellos
- » Mover información de una ventana a otra (dentro del mismo programa o entre diferentes programas)
- » Permitir que los usuarios reposicionen la información sobre la pantalla para adecuarla a sus necesidades particulares

Estas situaciones son comunes en uno o más de los siguientes ejemplos:

\* Consultas sobre pedidos.

\* Verificación de cuentas por pagar

\* Paquete de auditoría para el pago de impuestos

#### 5.2.4 Criterios de diseño para ventanas.

Las ventanas no cambian la necesidad de utilizar buenas prácticas de diseño. La necesidad de títulos, encabezados, números de página y otros aspectos son igualmente importantes en las ventanas. Cada documento o reporte que aparezca sobre una ventana debe estar diseñado como un documento único; en consecuencia, la plantilla es importante.

Tal como ya se indicó, a menudo se requiere de la capacidad para cambiar la ejecución del programa entre varias ventanas. Si se permite la entrada de datos, el sistema debe indicar, cuál es la ventana que se encuentra activa. Si el documento contenido en una ventana está formado por varias páginas, los usuarios deben ser capaces de ver cualquier página y entonces recorrerla en cualquier forma que ellos deseen. Cuando surge la necesidad de imprimir una copia de la página, debe ser

posible hacerlo en una forma sencilla. Cada ventana debe tener la capacidad de recorrer un documento, ya sea hacia arriba o hacia abajo, y de imprimirlo en forma independiente de las demás ventanas.

En algunos casos los analistas deciden utilizar *ventanas de aparición repentina (pop-up windows)*. Este tipo de ventanas aparecen y sobreescriben temporalmente información sobre la pantalla. Las pantallas de aparición repentina son efectivas para presentar a los usuarios mensajes y pedir información.

Las ventanas de cualquier tipo deben utilizarse con cuidado y de acuerdo con los lineamientos presentados. Cuando se emplean apropiadamente, las ventanas pueden incrementar la facilidad de uso de un sistema. En contraste, las ventanas mal diseñadas aparecerán como truculentas y pueden dañar seriamente la utilidad del sistema.

Para muchos usuarios finales, los reportes y la salida de una aplicación son el sistema. La habilidad de los usuarios finales para utilizar la información proporcionada y emplearla en la forma en que se presenta determina en gran medida la utilidad y aceptación del sistema.

Los iconos, que son representaciones gráficas de los datos, también pueden sustituir los informes tabulares y resultan especialmente útiles para mostrar proporciones o comparaciones.

La salida se puede generar por medio de una gran variedad de dispositivos entre los que se incluyen impresoras, cámaras y graficadores.

## CONCLUSIONES

Durante el diseño lógico, los analistas desarrollan las especificaciones para la salida del nuevo sistema; las cuales serán utilizadas después por los programadores para desarrollar el software que producirá el reporte. Para el usuario, la información que se obtiene del sistema, es una de las características más importantes, para financiar el proyecto, por lo tanto el analista debe seleccionar los métodos de salida que presenten la información con mayor efectividad.

La información tabular, presentada en un formato de renglón y columna, se recomienda para presentar detalles cuando se necesitan pocos comentarios o explicaciones.

Las formas gráficas para la presentación de información incluyen las gráficas de sectores, curvas de escalones y barras así como los mapas. Las gráficas de este tipo son superiores a los reportes tabulares para señalar tendencias y efectuar comparaciones al mismo tiempo que facilitan el recuerdo de grandes cantidades de datos.

Los íconos, que son representaciones gráficas de los datos, también pueden sustituir los informes tabulares y resultan especialmente útiles para mostrar proporciones o comparaciones.

La salida se puede generar por medio de una gran variedad de dispositivos entre los que se incluyen impresoras, cámaras y graficadores.

Para especificar el diseño lógico, los analistas crean una plantilla de la salida, que es un modelo que muestra la localización de la información. Para diseñar y presentar la plantilla se pueden utilizar cualquier tipo de forma, ya sea manual o directamente en computadora. En estas plantillas se marcan los detalles que son constantes y los que son variables; además se indican los resúmenes, los totales y las marcas de control para recalcar puntos específicos de información. Cuando se diseñan pantallas o paneles, los analistas posiblemente encuentren que es necesario utilizar varias pantallas o capacidades especiales de ventana, como las de aparición repentina, que sean fáciles de leer. La facilidad de lectura debe ser uno de los objetivos que sirvan de guía en el diseño de la salida de los sistemas.

Por lo tanto, considero que la metodología propuesta en este trabajo se apega a los estándares de productividad y eficiencia para el diseño de las salidas en los sistemas de información.

#### INTRODUCCION DE LOS COMPUTADORES EN LOS NEGOCIOS

Elias M. Awad

Edit. Prentice Hall

México, 1991.

#### INFORMATICA, PRESENTE Y FUTURO

Sanders, Donald H.

Edit. Mc Graw Hill

México, 1990.

#### SISTEMAS DE INFORMACIÓN PARA LA TOMA DE DECISIONES

Cohen, Daniel

Edit. Mc Graw Hill

México, 1994.

## **BIBLIOGRAFIA**

Davis, Gordon B.; Olson, Margrethe H.

Segunda Edición

Edt. Mc Graw Hill

### **ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACION.**

James A. Senn

Segunda Edición

Editorial Mc Graw Hill

INFORMATIZACIÓN DE EMPRESAS

García-Bedel, José Javier

Edt. Mc Graw Hill

### **INTRODUCCION AL DISEÑO DE SISTEMAS**

Erid Squire

Representaciones y Servicios de Ingeniería S.A

México, 1990.

CONCEPTOS DE LOS SISTEMAS DE INFORMACION PARA LA ADMINISTRACION

Lucas, Henry C. Jr.

Segunda Edición

### **INTRODUCCION DE LOS COMPUTADORES EN LOS NEGOCIOS**

Elias M.Awad

Edit. Prentice Hall

México, 1991.

TECNICAS DE ORGANIZACION Y ANALISIS DE SISTEMAS

Rodríguez Alfrado, Márquez, Antonio

### **INFORMATICA, PRESENTE Y FUTURO**

Sanders, Donald H.

Edit. Mc Graw Hill

México, 1990.

ORGANIZACION DE LOS SERVICIOS INFORMATICOS

Sales Panfilo, Jesús

### **SISTEMAS DE INFORMACIÓN PARA LA TOMA DE DECISIONES**

Cohen, Daniel.

Edit. Mc Graw Hill

México, 1994.

**SISTEMAS DE INFORMACION GERENCIAL**

Davis, Gordon B.; Olson, Margrethe H.

Segunda Edición

Edit. Mc Graw Hill

México 1989.

**GRAFICAS POR COMPUTADORA**

**INFORMATIZACION DE EMPRESAS**

García-Badell, José Javier

Edit. Mc Graw Hill

México, 1994.

**CONCEPTOS DE LOS SISTEMAS DE INFORMACION PARA LA ADMINISTRACION**

Lucas, Henry C. Jr.

Segunda Edición

Edit. Mc Graw Hill

México, 1988.

**TECNICAS DE ORGANIZACION Y ANALISIS DE SISTEMAS**

Rodríguez, Alfredo; Márquez, Antonio

Edit. Mc Graw Hill

México.

**ORGANIZACION DE LOS SERVICIOS INFORMATICOS**

Salas Parrilla Jesús

Edit. Mc Graw Hill

México, 1989.

**PRINCIPIOS DE SISTEMAS DE INFORMACION**

Scott, George M.

Edit. Mc Graw Hill

México, 1988.

**GRAFICAS POR COMPUTADORA**

Demel, John T.; Miller, Michael J.

Edit. Mc Graw Hill

México, 1988.

**CARACTERISTICAS DEL SISTEMA.**

Proporciona un ambiente agradable al usuario, así como facilidad en el manejo de este administrador, mediante menús y peticiones.

**MENU PRINCIPAL.**

Al iniciar el sistema de administrador de Compact Disk, aparecerá la pantalla del menú principal, la cual contiene las siguientes opciones:

Catálogos

Consultas

Reportes

Salir

**CATALOGOS.**

Esta opción lleva a la pantalla de capture de datos para el administrador, esta opción contiene a su vez las opciones de

Compact Disk se muestra la lista de CD que ya se encuentran dados de alta, si el CD no se encuentra en esa lista, el usuario presionará <ins> con la finalidad de añadir un nuevo CD. Si entre la lista se encontró el deseado, se presionará <ESC> para que se capturen los datos que se deben capturar, los cuales son:

Clave: Esta, se lleva automáticamente al elegir el título del CD.

## DESCRIPCION DEL SISTEMA.

Nombre: Aparece automáticamente al elegir el CD.

El administrador de Compact Disk, es un entorno desarrollado para proporcionar al usuario la consulta rápida y precisa de grabaciones musicales de los artistas nacionales e internacionales.

Se presiona <ESC> para que se introduzca su clave, si se desconoce, se presiona <ESC>, aparece un mensaje que indica si se desea consultar los intérpretes existentes en la lista para elegir el deseado y se presiona <ESC>.

## CARACTERISTICAS DEL SISTEMA.

Proporciona un ambiente agradable al usuario, así como facilidad en el manejo de este administrador, mediante menús y peticiones.

## MENU PRINCIPAL

Intérprete: Se muestra una lista con los intérpretes que se encuentran dados de alta; si el intérprete que se desea dar de alta ya se encuentra en esa lista, únicamente se presionará <ESC>, si no se desea presionar <ins> para añadir un nuevo intérprete.

Al iniciar el sistema de administrador de Compact Disk, aparecerá la pantalla del menú principal, la cual contiene las siguientes opciones:

**Catálogos**

**Consultas**

**Reportes**

**Salir**

Autor: En la pantalla que aparece se proporciona al usuario la lista con los autores y su clave ya dados de alta con anterioridad, si en esa lista se encuentra el nombre del autor deseado únicamente se presionará <ESC> y si no se presionará <ins> para insertar una nueva compañía.

## CATALOGOS.

Esta opción lleva a la pantalla de captura de datos para el administrador, esta opción contiene a su vez las opciones de:

**Compact Disk:** Al elegir esta opción se muestra en la pantalla la lista de CD que ya se encuentran dados de alta, si el CD no se encuentra en esa lista, el usuario presionará <Ins> con la finalidad de añadir un nuevo CD. Si entre la lista se encontró el deseado, se presionará <ENTER>. Después aparecen los datos que se deben capturar, los cuales son: menú principal presiona <ESC>.

**Clave:** Esta, se lleva automáticamente al elegir el título del CD.

**Nombre:** Aparece automáticamente al elegir el CD.

### CONSULTA.

**Intérprete:** Se pide que se introduzca su clave, si se desconoce, se presiona <ENTER>, aparece un mensaje que indica si se desea consultar los intérpretes existentes en la lista para elegir el deseado y se presiona <ESC>. musicales que se han registrado en el administrador, nos lleva a un nuevo menú con sus opciones

**Compañía Disquera:** Es el mismo mecanismo que con intérprete.

Cuando ya se capturaron los datos que se piden, aparece una pantalla en donde se deberán capturar las canciones correspondientes al CD. Para que se capture la clave, de no conocerse se presiona <ENTER> para que se muestre el mensaje que cuestiona acerca de si se desea consultar la lista de los intérpretes ya existentes, para que

**Intérprete:** Se muestra una lista con los intérpretes que se encuentran dados de alta; si el intérprete que se desea dar de alta ya se encuentra en dicha lista únicamente se presionará <ESC>; si no, se deberá presionar <Ins> para añadir un nuevo intérprete.

Para mostrar las canciones correspondientes a cada CD, se deberá elegir el CD deseado.

**Autor:** En la pantalla que aparece se proporciona al usuario la lista con los autores y su clave ya dados de alta con anterioridad, si en esa lista se encuentra el nombre del autor deseado únicamente se selecciona y presiona <ESC>; de lo contrario se presionará <Ins> y se capturan los datos de nuevo autor.

**Compañía Disquera:** En la lista que aparece se debe de elegir la compañía deseada <ESC>, en caso de no encontrarse en esa lista, se deberá presionar <Ins> para insertar una la nueva compañía.

**Género Musical:** Se pide el género musical al que pertenecen las canciones dadas de alta al igual que en la opción anterior.

**NOTA:** Para regresar al menú principal presione <ESC>.

## CONSULTA.

Esta opción nos ayuda a visualizar las grabaciones musicales que se han registrado en el administrador, nos lleva a un nuevo menú con sus opciones correspondientes, las cuales son:

**Por intérprete:** Su objetivo es consultar las grabaciones que están registradas en el administrador por el intérprete del Compact Disk. Se solicita que se capture la clave, de no conocerse se presiona <ENTER> para que se muestre el mensaje que cuestiona acerca de si se desea consultar la lista de los intérpretes ya existentes, para que el usuario elija el deseado presionando <ESC>. Regresa al usuario a la pantalla anterior, en donde se debe presionar <ENTER> para que aparezca la pantalla que contiene la clave y el título del CD del intérprete solicitado.

Para mostrar las canciones correspondientes a cada CD, se deberá elegir el CD deseado y presionar <ENTER>.

**Por Compañía Disquera:** Se deben seguir los mismos pasos que se siguieron para consultar por intérprete.

**Por Género Musical:** Igualmente se seguirán los pasos anteriormente descritos en la consulta por intérprete.

```
#include "inkey.ch"
#include "setcurs.ch"
```

## REPORTES.

Esta opción nos proporciona la facilidad de llevar un fácil manejo de los CD existentes en el mercado. las opciones que contiene son:

Por Compact Disk: Se manda a impresión, presionando <ENTER>. En el listado aparecen los títulos de los CD con su clave e intérprete.

Por intérprete: Al igual que por CD, se manda a impresión, presionando <ENTER>. En el listado impreso, se proporcionan los intérpretes con sus claves existentes en el Administrador de Compact Disk.

```
** nTop      && rengion superior
** nLeft     && Columna izquierda
** Por intérprete: Al igual que por CD, se manda a impresión, presionando
<ENTER>. En el listado impreso, se proporcionan los intérpretes con sus claves
existentes en el Administrador de Compact Disk.
** Validation && Arreglo conteniendo objetos para Validación
** Pictures   && Arreglo conteniendo los pictures de edición de columna
** Near      && Uso de Búsqueda al elemento mas parecido
** Edit      && Si se permite la edición del Browse
** PGuia     && Pantalla Guia
** Val_Baja  && Procedimiento para bajas
** Proc_Edit && Procedimiento para edición de campos
** Listado   && Procedimiento objeto para listados
** Refresh   && Arreglo logico indicando con .T. o .F. los campos claves
** Valid     && Arreglo de validación
```

**SALIR.**

Termina el Administrador de Compact Disk.

```
local b, column, cType, n
local cColorSave, nCursSave
local More, nKey, lAppend
local sn, bpart
local w_guia, CanDelete, lcf
local Getlist:={}
```

```
if Edit = Nil
    Edit = .t.
end
```

```
W_Guia := savescreen(23,1,24,79)
```

```
Go top
```

```
if PGuia # Nil
```

```
    if PGuia
```

```
        if iscolor()
```

```
            if Near .and. ValType(Indextoy(0)) = 'C'
```

```
                @ 23,0 Say "
```

```
                @ 23,Col()+1 say "<ln> Aездr."
```

```
**** 0000000000
```

```
*
```

```
* browse.prg
```

```
*
```

```
#include "inkey.ch"
```

```
#include "setcurs.ch"
```

```
#define MY_HSEP "INI"
```

```
#define MY_CSEP "  "
```

```
func Browse(AControl, nTop, nLeft, nBottom, nRight, Fields, Headings, Validation, ;  
Pictures, Near, Edit, PGuia, Val_Baja, Proc_Edit, Listado, Refresh, Vinitial)
```

```
End  
** nTop      && renglon superior  
** nLeft     && Columna izquierda  
** nBottom   && Renglon inferior  
** nRight    && Columna derecha  
** Fields    && Arreglo conteniendo nombres de los campos en Browse  
** Headings  && Arreglo conteniendo Encabezados de Columnas  
** Validation && Arreglo conteniendo objetos para Validacion  
** Pictures   && Arreglo conteniendo los pictures de edicion de columna  
** Near      && Uso de Busqueda al elemento mas parecido  
** Edit      && Si se permite la edicion del Browse  
** PGuia     && Pantalla Guia  
** Val_Baja  && Procedimiento para bajas  
** Proc_Edit && Procedimiento para la edicion de campos  
** Listado   && Procedimiento objeto para listados  
** Refresh   && Arreglo logico indicando con .T. o .F. los campos llaves  
** Vinitial  && Rutina de iniciacion
```

```
local b, column, cType, n  
local cColorSave, nCursSave  
local lMore, nKey, lAppend  
local sn, bpart  
local w_guia, CanDelete, tof  
local Getlist:={}
```

```
If Edit = Nil
```

```
    Edit:=.t.
```

```
end
```

```
W_Guia := savescreen(23,1,24,79)
```

```
Go top
```

```
If PGuia # Nil
```

```
    If PGuia
```

```
        if iscolor()
```

```
            if Near .and. ValType(Indexkey(0)) = 'C'
```

```
                @ 23,0 Say "
```

```
                @ 23,Col()+1 say "<Ins> Aadir, "
```

```

end
end
else
    @ 23, Col() say "<Del> Borrar,"
    @ 24,1 say "<Esc> Salir, <F3 > Busqueda, <F2 > Listado"
end
end
end
End
End
/* Creacion del objeto Browse */
b := TBrowseDB(nTop, nLeft, nBottom, nRight)

/* Asignacion a las variables de Instancia del Browse */
b:headSep := MY_HSEP    && Separador de Encabezados
b:colSep := MY_CSEP    && Separador de

/* Asigna un procedimiento 'skipper' (para handle append mode) */
b:skipBlock := {|x| Skipper(x, lAppend)}
nCursSave := SetCursor(SC_NORMAL)
b:colorSpec := "W+/BG,W+/R"

for n = 1 to Len(Fields)

/* agrega las columnas nuevas */
column := TBColumnNew( Headings[n], FieldWBlock(Fields[n], Select()))

/* evaluate the block once to get the field's data type */
b:addColumn(column)

next

/* Congela las dos primeras columnas */

lAppend := .f.

lMore := .t.
while (lMore)
    /* stabilize the display */
    while ( !b:stabilize() )
        nKey := lnKey()
        if ( nKey = 27 )
            If PGUIA == NIL
                If PGUIA
                    Borraguia()
            end
        end
        end
        restscreen(23,1,24,79,w_guia)
        setcursor(NCursSave)
        return (.T.)    /* (abort if a key is waiting) */

```

```

case ( nKey == K_CTRL_PGDN )
end
end
if ( b:stable )
/* display is stable */
case ( if ( b:hitBottom .and. !Append )
/* banged against EOF; go into append mode */
Append := .t.
nKey := K_DOWN
else
if ( b:hitTop .or. b:hitBottom )
Tone(125, 0)
end
case ( nKey == K_CTRL_PGDN )
/* everything's done; just wait for a key */
nKey := InKey(0)

case ( nKey == K_CTRL_LEFT )
end
end

case ( nKey == K_CTRL_RIGHT )
/* process key */
do case
Case ( NKey == K_INS )
b:goBottom()
do while !(B:Stabilize()); end
nkey:=K_DOWN
b:down()
case ( nKey == K_DOWN )
b:down()

case ( nKey == K_UP )
b:up()

if ( !Append )
Append := .f.
b:refreshAll()
end

case ( nKey == K_PGDN )
b:pageDown()

case ( nKey == K_PGUP )
b:pageUp()
if ( !Append )
Append := .f.
b:refreshAll()
end

case ( nKey == K_CTRL_PGUP )
b:goTop()
Append := .f.

```

```

case ( nKey == K_CTRL_PGDN )
    b:goBottom()
    lAppend := .f.

case ( nKey == K_RIGHT )
    b:right()

case ( nKey == K_LEFT )
    b:left()

case ( nKey == K_HOME )
    b:home()

case ( nKey == K_END )
    b:end()

case ( nKey == K_CTRL_LEFT )
    b:panLeft()

case ( nKey == K_CTRL_RIGHT )
    b:panRight()

case ( nKey == K_CTRL_HOME )
    b:panHome()

case ( nKey == K_CTRL_END )
    b:panEnd()

case ( nKey == K_ESC )
    lMore := .f.

case ( nKey == K_RETURN )
    If Edit
        DoGet(b, lAppend, Validation[b.Colpos],
Pictures[b.ColPos],Proc_EDIT, Refresh, Vinitial, AControl)
    ELSE
        lMORE := .f.
    Endif
Case (NKey == K_DEL)
    If Val_Baja # NIL
        CanDelete := EVAL(Val_Baja)
        DBCommit()
    else
        Candelete:= .T.
    Endif
    If CanDelete
        sn = 'S'
        bpart:=DoBox(21,0,23,28)
        @ 22,1 say 'Teclera <S>i para Borrar:' get Sn valid Sino(SN)
        read
        If sn == 'S' .or. Sn == 's'

```

```

Delete
DBCommit()
b:refreshAll()
while ( !b:stabilize() ); end

elseif ( n > 0 .and. Recno() == LastRec() + 1 )
while ( i < n )
RestScreen(21,0,23,28,bpant)
Endif
If ( Eof() )
If ( !Append )
i++
else
SKIP -1
end

Case nKey == K_F2
If Listado # NIL
tof:= EVAL(Listado)
else
tof:=Espera('No tiene opcion de listado',10)
end

Case nKey == K_F3 .and. near .and. ValType(Indexkey(0)) = 'C'
If !DoSeek()
elseif ( n < 0 ) SetCursor( if(ReadInsert(), SC_INSERT, SC_NORMAL) )
while ( i < n )
b:goBottom()
!Append := .f.
else
SetCursor( if(ReadInsert(), SC_INSERT, SC_NORMAL) )
!Append := .F.
b:refreshAll()
while ( !b:stabilize() ); end

end
Otherwise
If Edit
SetCursor(SC_NORMAL)
KEYBOARD( Chr(nKey) )
DoGet(b, !Append, Validation[b:Colpos],
Pictures[b:ColPos],Proc_Edit, Refresh, Vinitial, AControl)
end

end

****
end
restscreen(23,1,24,79,w_guia)
SetCursor(nCursSave)
Commit
return (.t.)
local column, get, nKey, c
Local GetList:=(), CExpr, xEval, LFrash
****
* Skipper()
* while ( !b:stabilize() ); end

func Skipper(n, !Append)
local i
/* If confirming new record, append blank */
If ( !Append .and. Recno() == LastRec() + 1 )
i := 0
If Vinitial # NIL
If ( LastRec() != 0 ) (Vinitial)

```

```

if ( n == 0 )
    SKIP 0
    APPEND BLANK
elseif ( n > 0 .and. Recno() != LastRec() + 1 )
    while ( i < n )
        SKIP 1
        if ( Eof() )
            if ( !Append )
                i++
            else
                SKIP -1
            end
        end
    end
    exit
end
elseif ( n < 0 )
    while ( i > n )
        SKIP -1
        if ( Bof() )
            exit
        end
    end
end
end
SetCursor(SC_NORMAL)
return (i)

```

\*\*\*\*

```

* DoGet()
*

```

```

func DoGet(b, !Append, Valid, Pic, Proc_Edic, Refresh, VInitial, AControl)
local blnsSave, IScoreSave, IExitSave
local column, get, nKey, c
Local GetList:=(), CExpr, xEval, LFresh

```

```

/* make sure browse is stable */
while ( !b:stabilize() ); end

```

```

/* if confirming new record, append blank */
if ( !Append .and. Recno() == LastRec() + 1 )
    if VInitial # NIL
        Eval (VInitial)
    end
end

```

```

else
    APPEND BLANK
end
DBCommit()
end
if Proc_Edic # NIL
    c := EVAL(Proc_Edic, AControl)
    DBCommit()
    B:RefreshAll()
    while ( !b:stabilize() ); end
    Return (NIL)
endif
/* save state */
IScoreSave := Set(_SET_SCOREBOARD, .f.)
IExitSave := Set(_SET_EXIT, .t.)
blnsSave := SetKey(K_INS)

/* set insert key to toggle insert mode and cursor */
SetKey( K_INS, ;
    (|| SetCursor( if(ReadInsert(!ReadInsert()), SC_NORMAL, SC_INSERT)));
)

/* initial cursor setting */
SetCursor( if(ReadInsert(), SC_INSERT, SC_NORMAL) )

/* get column object from browse */
cExpr := IndexKey(0)
if ( !Empty(cExpr) )
    /* expand key expression for later comparison */
    xEval := &cExpr
end
column := b:getColumn(b:colPos)

/* create a corresponding GET */
get := GetNew(Row(), Col(), column:block, column:heading,, b:colorSpec)
Get:Picture := Pic
Get:PostBlock := Valid
/* read it */
ReadModal( {get} )
DBCommit()
/* restore state */
SetCursor(SC_NORMAL)
Set(_SET_SCOREBOARD, IScoreSave)
Set(_SET_EXIT, IExitSave)
SetKey(K_INS, blnsSave)
Ifresh := (xEval != &cExpr)
if ( !Ifresh )

```

```

    if ( nRec < FreshOrder(b) )
    else /* record pointer may move if bol is on screen */
        B:refreshCurrent( nRec )
    end
    nKey := LastKey()
    if ( nKey == K_UP .or. nKey == K_DOWN .or. ;
        nKey == K_PGUP .or. nKey == K_PGDN )
    end
    return (NIL)
end

```

```
return (NIL)
```

```
Function DoSeek()
```

```
Local vent, Val, GetList:=[], ret, tof
```

```
SetCursor( if(ReadInsert(), SC_INSERT, SC_NORMAL) )
```

```
Val:=Space(58)
```

```
Vent:=DoBox(10,10,12,70,' Buscar?')
```

```
@ 11,12 get Val
```

```
read
```

```
if lastkey() # 27
```

```
Go top
```

```
tof:=DBSeek(Val,.t.)
```

```
end
```

```
Restscreen(10,10,12,70,Vent)
```

```
Return !(Eof())
```

```
Static function BorraGuia(PGuia)
```

```
If PGuia # NIL
```

```
  If PGuia
```

```
    if iscolor()
```

```
      @ 23,1 say "
```

```
      @ 24,1 say "
```

```
    else
```

```
      @ 23,1 say "
```

```
      @ 24,1 say "
```

```
    end
```

```
  End
```

```
End
```

```
return (NIL)
```

```
static func FreshOrder(oB)
```

```
local nRec
```

```
  nRec := Recno()
```

```
  oB:refreshAll()
```

```
  /* stabilize to see if TBrowse moves the record pointer */
  while ( !oB:stabilize() ); end
```

```

Procedure
  If ( nRec != LastRec() + 1 )
    /* record pointer may move if bof is on screen */
    Local Op, Vert
    while ( Recno() != nRec )
      /* falls through unless record is closer to bof than before */
      oB:up()
      while ( !oB:stabilize() ); end
    end
  end
return (NIL)

```

```

Menu to Op
Do Case
  Case Op = 1
    Cat_CD(AControl, 1)
  Case Op = 2
    Cat_Interp(AControl, 1)
  Case Op = 3
    Cat_Autor(AControl, 1)
  Case Op = 4
    Cat_CDisquera(AControl, 1)
  Case Op = 5
    Cat_Genero(AControl, 1)
End

```

```

End
RestScreen(03,05,09,25, Vert)
Return

```

```

Procedure Cat_CD(AControl, Cierra, Filtro, AbreInt, AbreComp)

```

```

Local Vert
Local F:={Clave', 'Titulo'}
Local H:={Clave', 'T|t|u|o'}
Local V:={| | (1)}, (| (1))}
Local P:={XXXXXXXXXXXX, 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'}

```

```

AbreInt:=If(AbreInt == NIL, 1, AbreInt)
AbreComp:=If(AbreComp == NIL, 1, AbreComp)
If AbreInt
  Select 2
    AbreB(AControl, "Interp", 2)
  End
If AbreComp
  Select 4
    AbreB(AControl, "CompD", 2)
  End
Select 7
  AbreB(AControl, "Cancion", 1)
Select 8
  AbreB(AControl, "CD", 2)
Set Order to 2

```

Procedure Catalogo(AControl, Cat\_Logo de Compact Disk)

Local Op, Vent

Vent:=DoBox(03,05,09,25)

While (Op # 0)

- @ 04,07 Prompt 'Compact Disk'
  - @ 05,07 Prompt 'Interpretes'
  - @ 06,07 Prompt 'Autor'
  - @ 07,07 Prompt 'Comp. Disquera'
  - @ 08,07 Prompt 'G,nero Musical'
- Menu to Op

Do Case

- Case Op = 1  
Cat\_CD(AControl, .t.)
- Case Op = 2  
Cat\_Interp(AControl, .t.)
- Case Op = 3  
Cat\_Autor(AControl, .t.)
- Case Op = 4  
Cat\_CDisquera(AControl, .t.)
- Case Op = 5  
Cat\_Genero(AControl, .t.)

End

RestScreen(03,05,09,25, Vent)

Return

Procedure Cat\_CD(AControl, Cierra, Filtro, AbreInt, AbreComp)

Local Vent

Local F:={'Clave', 'Titulo'}

Local H:={'Clave', 'T i t u l o'}

Local V:={|(t.), |(t.)}

Local P:={'XXXXXXXXXX', 'XXX'}

AbreInt:=If(AbreInt == NIL, .t., AbreInt)

AbreComp:=If(AbreComp == NIL, .t., AbreComp)

If AbreInt

Select 2

AbreB(AControl, "Interp", 2)

End

If AbreComp

Select 4

AbreB(AControl, "CompD", 2)

End

Select 7

AbreB(AControl, "Cancion", 1)

Select 6

AbreB(AControl, "CD", 2)

Set Order to 2

```

Vent:=DoBox(06,08,18,69, "Cat logo de Compact Disk")
If Filtro # NIL
  Set Filter to &Filtro
End
Browse(AControl, 07,09,17,68, F, H, V, P, ., ., ., ., {{AControl|Edit_CD(AControl)})
Set Order to 1
If Filtro # NIL
  Set Filter to
End
RestScreen(06,08,18,69, Vent)
If Cierra
  Close All
End
Return

Procedure Edit_CD(AControl)

Local GetList:={}, Vent
Local Cve, Tit, Int, Comp

  Cve:=CD->Clave
  Tit:=CD->Titulo
  Int:=CD->Interprete
  Comp:=CD->CompD
  Vent:=DoBox(04,02,23,78)
  @ 05,10 Say 'Clave: ' Get Cve Valid ValKey(CD->Clave)
  @ 06,10 Say 'Nombre: ' Get Tit
  @ 07,10 Say 'Interprete: ' Get Int Valid Valida(AControl, 2, {{AControl|Cat_Interp(AControl, .f.)})
  @Int, "Interp->Clave", "SubStr(Interp->Nombre,1,40)", 07,31)
  @ 08,10 Say 'Compañia Disquera: ' Get Comp Valid Valida(AControl, 4,
  {{AControl|Cat_CDisquera(AControl, .f.)}), @Comp, "CompD->Clave", "CompD->Comp", 08,36)
  Read
  Replace Clave with Cve,;
  Titulo with Tit,;
  Interprete with Int,;
  CompD with Comp
  Cat_Cancion(AControl, "Cancion->Clave == CD->Clave")
  RestScreen(04,02,23,78, Vent)
Return

Procedure Cat_Cancion(AControl, Filtro)

Local Vent, Area
Local F:={'Cancion'}
Local H:={'C a n c i o n'}
Local V:={{| |(.)}}
Local P:={'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'}

  Area:=Select()
  Select 7
  AbreB(AControl, "Cancion", 1)

```



```
If Cierra
  Close All
End
Select(Area)
Return
```

Static Procedure Edit\_Int()

```
Local GetList:={}, Vent
Local Cve, Nomb, LO
```

```
Vent:=DoBox(10,08,14,71)
Cve:=Interp->Clave
Nomb:=Interp->Nombre
LO:=Interp->LOrigen
@ 11,10 Say 'Clave: ' Get Cve Valid ValKey(Cve)
@ 12,10 Say 'Nombre: ' Get Nomb
@ 13,10 Say 'Lugar de Origen: ' Get LO
Read
If LastKey() # 27
  Replace Clave with Cve,;
  Nombre with Nomb,;
  LOrigen with LO
End
RestScreen(10,08,14,71, Vent)
Return
```

Procedure Cat\_Autor(AControl, Cierra)

```
Local Vent, Area
Local F:={'Clave', 'Nombre'}
Local H:={'Clave', 'Nombre'}
Local V:={{|ValKey(Autores->Clave)}, {|(.t.)}}
Local P:={'XXXXXX', 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'}
```

```
Area:=Select()
Select 3
AbreB(AControl, "Autores", 2)
Set Order to 2
Vent:=DoBox(06,08,18,69, "Catalogo de Autores")
Browse(AControl, 07,09,17,68, F, H, V, P, .t., .t., .t.)
Set Order to 1
RestScreen(06,08,18,69, Vent)
If Cierra
  Close All
End
Select(Area)
Return
```

Procedure Cat\_CDisquera(AControl, Cierra)

```
Local Vent, Area
Local F:={'Clave', 'Comp'}
Local H:={'Clave', 'C o m p a ñ i a'}
Local V:={{| |ValKey(CompD->Clave)}, {| |(t.)}}
Local P:={'XXXXX', 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'}
```

```
Area:=Select()
Select 4
AbreB(AControl, "CompD", 2)
Set Order to 2
Vent:=DoBox(06,08,18,69, "Catalogo de Compañias Disqueras")
Browse(AControl, 07,09,17,68, F, H, V, P, .t., .t., .t.)
Set Order to 1
RestScreen(06,08,18,69, Vent)
If Cierra
    Close All
End
Select(Area)
Return
```

Procedure Cat\_Genero(AControl, Cierra)

```
Local Vent, Area
Local F:={'Clave', 'Genero'}
Local H:={'Clave', 'G e n e r o'}
Local V:={{| |ValKey(Generos->Clave)}, {| |(t.)}}
Local P:={'XXXXX', 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'}
```

```
Area:=Select()
Select 5
AbreB(AControl, "Generos", 2)
Set Order to 2
Vent:=DoBox(06,08,18,69, "Catalogo de G,neros Musicales")
Browse(AControl, 07,09,17,68, F, H, V, P, .t., .t., .t.)
Set Order to 1
RestScreen(06,08,18,69, Vent)
If Cierra
    Close All
End
Select(Area)
Return
```

Procedure CDisk()

Local GetList:=(), AControl[3], Opcion:=1

```
Set Date to British (20.07.40)
Set Scoreboard Off
Set Talk Off Prompt 'Por interpretar'
Set Wrap On Prompt 'Por Comp. Disq.'
Select 1 Prompt 'Por G.n. Musical'
Use Control Op
AControl[1]:=AllTrim(Control->Empresa)
Acontrol[2]:=AllTrim(Control->Color)
Acontrol[3]:=AllTrim(Control->Datos)
Close All Op = 2
Set Color to (AControl[2]) AControl
Clear Case Op = 3
DoBox(01,00,24,79) (AControl)
@ 00,00 Say PadC(AControl[1], 80, ' ') Color "W+/BG"
@ 03,00 Say '||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||'
While (Opcion # 0) .and. (Opcion # 4)
  @ 02,05 Prompt 'Catalogos'
  @ 02,20 Prompt 'Consultas'
  @ 02,35 Prompt 'Reportes'
  @ 02,50 Prompt 'Salir'
  Menu to Opcion ' ' Vent
  Do Case
    Case Opcion = 1
      AbreB(Catalogo(AControl))
    Case Opcion = 2
      @ 05 Consulta(AControl) Get Int Valid Valids(AControl, 2, (AControl|Cat_Interp(AControl, .I))
    @ Int. SubStr(interp->Nombre, 1, 40)", 05,31)
      Read Reportes(AControl)
    End Key() # 27
  End
  Close All
  Set Color to
  Clear Screen(04,02,23,78, Vent)
Return
```

Static Procedure Con\_CDisquera(AControl)

Local GetList:=(), Comp:= ' ', Vent

```
Select 4
  AbreB(AControl, "CompD", 2)
  Vent:=DoBox(04,02,23,78)
  @ 05,10 Say 'Compa=la Disquera: ' Get Comp Valid Valids(AControl, 4
  (AControl|Cat_CDisquera(AControl, .I)), @Comp, "CompD->Claw", "CompD->Comp", 05,38)
  Read
  If LastKey() # 27
    Cat_CD(AControl, .I, "CD->CompC == CompD->Claw", .I)
  End
```

## Procedure Consulta(AControl)

Local Op, Vent

```
Vent:=DoBox(03,20,07,40)
While (Op # 0)
  @ 04,22 Prompt 'Por Interprete '
  @ 05,22 Prompt 'Por Comp. Disq. '
  @ 06,22 Prompt 'Por G.n. Musical'
  Menu to Op
  Do Case
    Case Op = 1
      Con_Interp(AControl)
    Case Op = 2
      Con_CDisquera(AControl)
    Case Op = 3
      Con_Genero(AControl)
  End
End
RestScreen(03,20,07,40, Vent)
Return
```

## Static Procedure Con\_Interp(AControl)

Local GetList:={}, Int:=' ', Vent

```
Select 2
  AbreB(AControl, "Interp", 2)
  Vent:=DoBox(04,02,23,78)
  @ 05,10 Say 'Interprete: ' Get Int Valid Valida(AControl, 2, {|AControl|Cat_Interp(AControl, .f.)})
  @Int, "Interp->Clave", "SubStr(Interp->Nombre,1,40)", 05,31)
  Read
  If LastKey() # 27
    Cat_CD(AControl, .f., "CD->Interprete == Interp->Clave", .f.)
  End
  Close All
  RestScreen(04,02,23,78, Vent)
Return
```

## Static Procedure Con\_CDisquera(AControl)

Local GetList:={}, Comp:=' ', Vent

```
Select 4
  AbreB(AControl, "CompD", 2)
  Vent:=DoBox(04,02,23,78)
  @ 05,10 Say 'Compañia Disquera: ' Get Comp Valid Valida(AControl, 4,
{|AControl|Cat_CDisquera(AControl, .f.)}), @Comp, "CompD->Clave", "CompD->Comp", 05,36)
  Read
  If LastKey() # 27
    Cat_CD(AControl, .f., "CD->CompD == CompD->Clave", .f.)
  End
```

```
Close All
RestScreen(04,02,23,78, Vent)
Return
```

```
Static Procedure Con_Genero(AControl)
```

```
Local GetList:={}, Gen:=' ', Vent
```

```
  Select 5
  AbreB(AControl, "Generos", 2)
  Vent:=DoBox(04,02,23,78)
  @ 05,10 Say 'G,nero Musical: ' Get Gen Valid Valida(AControl, 5,
  {|AControl|Cat_Generos(AControl, .f.)}, @Gen, "Generos->Clave", "Generos->Genero", 05,33)
  Read
  If LastKey() # 27
  Cat_Cancion(AControl, "Cancion->Genero == Generos->Clave")
  End
  Close All
  RestScreen(04,02,23,78, Vent)
Return
```

```
Local Int
```

```
  Select 2
  AbreB(AControl, 'Interp', 2)
  Select 8
  AbreB(AControl, 'CD', 2)
  @ 22,01 Say PadC('Imprimiendo...', 78, ' ')
  Set Device to Printer
  Set Printer On
  While !Eof()
  If (Int == NIL)
  Encab_CD(AControl)
  End
  Int =if(Interp->(DBSeek(CD->Interpreta)), Interp->Nombre, 'No Existe')
  @ PRow()+1,02 Say CD->Clave
  @ PRow(),14 Say CD->Titulo
  @ PRow(),39 Say Int
  Skip 1
  If (PRow() > 54) .and. (!Eof())
  Eject
  Encab_CD(AControl)
  End
  End
  If PRow() <= 55
  Eject
  End
  Set Device to Screen
  Set Printer Off
  @ 22,01 Say PadC(' ', 78, ' ')
  Close All
Return
```

Procedure Reportes(AControl)

Local Op, Vent

```
Vent:=DoBox(03,35,06,55)
While (Op # 0)
  @ 04,37 Prompt 'Compact Disk '
  @ 05,37 Prompt 'Interpretes '
  Menu to Op
  Do Case
    Case Op = 1
      Rep_CD(AControl)
    Case Op = 2
      Rep_Interp(AControl)
  End
End
RestScreen(03,35,06,55, Vent)
Return
```

Static Procedure Rep\_CD(AControl)

Local Int

```
Select 2
  AbreB(AControl, 'Interp', 2)
Select 6
  AbreB(AControl, 'CD', 2)
  @ 22,01 Say PadC('Imprimiendo...', 78, '')
  Set Device to Printer
  Set Printer On
  While !Eof()
    If (Int == NIL)
      Encab_CD(AControl)
    End
    Int:=If(Interp->(DBSeek(CD->Interprete)), Interp->Nombre, 'No Existe')
    @ PRow()+1,02 Say CD->Clave
    @ PRow(),14 Say CD->Titulo
    @ PRow(),59 Say Int
    Skip 1
    If (PRow() > 54) .and. (!Eof())
      Eject
      Encab_CD(AControl)
    End
  End
  If PRow() <= 55
    Eject
  End
  Set Device to Screen
  Set Printer Off
  @ 22,01 Say PadC(' ', 78, '')
  Close All
Return
```

Static Procedure Encab\_CD(AControl)

- @ 00,01 Say Chr(27)+Chr(15)
- @ 01,01 Say PadC(AControl[1], 130, '')
- @ 02,01 Say PadC('Listado de Compact Disk', 130, '')
- @ PRow(),114 Say 'Fecha: ' + DtoC(Date())
- @ 03,01 Say PadC('-', 130, '-')
- @ 04,01 Say ' Clave      Titulo
- @ 05,01 Say PadC('-', 130, '-')

Interprete'

Return

Static Procedure Rep\_Interp(AControl)

Local Primero

Select 2

AbreB(AControl, 'Interp', 2)

@ 22,01 Say PadC('Imprimiendo...', 78, '')

Set Device to Printer

Set Printer On

While !Eof()

If (Primero == NIL)

    Encab\_Int(AControl)

    Primero:=.f.

End

@ PRow()+1,02 Say Interp->Clave

@ PRow(),14 Say Interp->Nombre

Skip 1

If (PRow() > 54) .and. (!Eof())

    Eject

    Encab\_Int(AControl)

End

End

If PRow() <= 55

    Eject

End

Set Device to Screen

Set Printer Off

@ 22,01 Say PadC(' ', 78, '')

Close All

Return

Static Procedure Encab\_Int(AControl)

@ 01,01 Say PadC(AControl[1], 78, '')

@ 02,01 Say PadC('Listado de Interpretres', 78, '')

@ PRow(),62 Say 'Fecha: ' + DtoC(Date())

@ 03,01 Say PadC('-', 78, '-')

@ 04,01 Say ' Clave      Interprete'

\* 9999999 XXX

@ 05,01 Say PadC('-', 78, '-')

\* 123456789012345678901234567890123456789012345678901234567890123

\*                    1        2        3        4        5        6

Return

Function DoBox(x1, y1, x2, y2, Titulo)

Local Vent, Col

```
Vent:=SaveScreen(x1, y1, x2, y2)
@ x1,y1,x2,y2 BOX "É|»º%|É•"
If (Titulo # NIL)
  Col:=y1 + ((y2-y1)/2) - (Len(Titulo)/2)
  @ x1, Col Say AllTrim(Titulo)
End
```

Return Vent

Function SiNo(Var)

Local Ret:=.t.

```
If (Upper(Var) # 'S') .and. (Upper(Var) # 'N')
  Espera("Debe ser <S> o <N>")
  Ret:=.f.
End
```

Return Ret

Procedure Espera(Msg)

Local Vent, c1:=39-Int(Len(Msg)/2)  
Local c2:=c1+Len(Msg)+2

```
Vent:=SaveScreen(21,c1,23,c2)
@ 21,c1,23,c2 BOX "É|»º%|É•"
@ 22,c1+2 Say Msg
Inkey(0)
RestScreen(21,c1,23,c2, Vent)
```

Return

Procedure AbreB(AControl, Base, NInd)

Do Case

Case NInd = 1

Use(AControl[3]+Base) Index (AControl[3]+Base+'1')

Case NInd = 2

Use(AControl[3]+Base) Index (AControl[3]+Base+'1'), (AControl[3]+Base+'2')

Case NInd = 3

Use(AControl[3]+Base) Index (AControl[3]+Base+'1'), (AControl[3]+Base+'2'),

(AControl[3]+Base+'3')

End

Return

Function ValKey(Key)

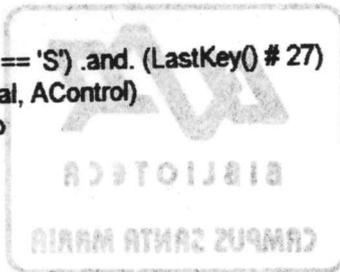
Local Ret:=.t., Reg, OldIndex

```
OldIndex:=IndexOrd()
Reg:=RecNo()
Set Order to 1
If DBSeek(Key) .and. Reg # RecNo()
  Espera("Error: Llave duplicada")
  Ret:=.f.
End
Goto Reg
Set Order to OldIndex
Return Ret
```

Function Valida(AControl, Sel, Proc\_Eval, Var, Campo, Desp, X, Y)

Local GetList:={}, Area:=Select(), Resp:='S', Ret

```
Select(Sel)
Ret:=DBSeek(Var)
If !Ret
  @ 22,12 Say 'Clave no encontrada; deseas consultar los existentes?';
  Get Resp Valid SiNo(Resp)
  Read
  @ 22,12 Say '
  If (Upper(Resp) == 'S') .and. (LastKey() # 27)
    Eval(Proc_Eval, AControl)
    Var:=&Campo
  Else
    Ret:=.f.
  End
End
@ X,Y Say &Desp
Select(Area)
Return Ret
```



1  
500  
1521  
1981