

---

---

# SISINF

## DEFINICION DE LA BASE DE DATOS

Versión 7.3

---

---

**Kratos**, S.A. de C.V.

La Tecnología en Software.

Derechos Reservados ©. Prohibida la reproducción total o parcial sin permiso escrito de **KRATOS, S.A. de C.V.** El uso de programas que integran **SISINF** se vende y renta bajo contrato con **KRATOS, S.A. de C.V.**

## CONTENIDO

---

CONTENIDO.....	2
PREFACIO.....	4
1) ESTRUCTURA DE LOS ARCHIVOS.....	5
1.1) Registros y Dependencias.....	5
1.2) Llaves.....	8
1.3) Area en Disco.....	10
1.4) Consumo de CPU.....	12
1.5) Archivos mayores a 2G.....	14
1.6) Resumen.....	16
2) TIPOS DE CAMPO.....	18
2.1) Características.....	18
2.2) Area en Disco en Campos Imagen.....	21
2.3) Area en disco en campos texto.....	21
3) DOCUMENTACION.....	23
4) SEGURIDAD.....	24
5) CAS.....	25
6) VARIABLES DE PROGRAMAS.....	26
7) DEFINICIONES DE DBD.....	27
7.1) Introducción.....	27
7.2) Reglas de Sintaxis.....	29
7.3) Definición de ARCHIVO.....	29
7.4) Definición de RECURRENTE.....	31
7.5) Definición de AREA.....	33
7.6) Definición de un Campo.....	34
7.6.1) Nombre del campo.....	35
7.6.2) Letrero de terminal y de impresora.....	36
7.6.3) Recurrente y tipo de campo en disco.....	37
7.6.4) Tipo de campo en memoria y long/dec.....	38
7.6.5) Niveles de seguridad.....	40
7.6.6) Máscara de edición.....	40
7.6.7) Dígito Verificador.....	44

7.6.8) Rango.....	44
7.6.9) Número de Seguridad.....	45
7.6.10) Valor.....	45
7.6.11) Grupo.....	45
7.6.12) Area.....	45
7.6.13) Opciones anteriores de Imagen.....	46
7.6.14) Opciones de Imagen.....	46
7.6.15) DIP y DII.....	47
7.6.16) Opción CAS.....	48
7.6.17) Mayúsculas.....	48
7.7) Grupo.....	48
7.8) Definición de IGUAL.....	49
7.9) Definición de TABLA.....	50
7.10) *D.....	51
8) EJECUCION DE DBD.....	52
8.1) Pasos para la ejecución.....	52
8.2) Datos para la ejecución.....	53
8.3) Errores.....	55
8.4 Opción "A" (Archivos) en módulo DBD.....	57
9) EJEMPLO.....	59
9.1) Ejemplos de DBD de Clientes.....	59
9.2) Ejemplo de Ayudas en DBD.....	63
9.3) Ejemplo de DBD de Imágenes.....	65
10) INSTRUCCIONES A LA DBD.....	67
10.1) Para qué sirven?.....	67
10.2) Sintaxis.....	67
10.3) Valor.....	68
10.4) Quitar Definiciones.....	69
10.5) Reemplazar Definiciones.....	71
10.6) Incluir otro archivo.....	72
10.7) Comentarios y listados.....	72
11) FUNCIONAMIENTO DE DBD.....	74
12) VALORES MAXIMOS.....	76

## PREFACIO

---

El módulo DBD de **SISINF** sirve para especificar la estructura de los archivos, mismos que se usarán en uno o varios de los Sistemas. También se definen características y validaciones de los campos que forman dichos archivos.

El contenido de este Manual es el siguiente:

- ◆ Los capítulos 1 al 6 son una descripción general de las diferentes definiciones y sirven para un mejor diseño de la aplicación. También se comenta la relación de ciertas definiciones con otros módulos de **SISINF** como son CAS y DOCumentación.
- ◆ El capítulo 7 es una descripción formal de todas las definiciones de la DBD.
- ◆ El capítulo 8 explica como ejecutar el módulo DBD.
- ◆ El capítulo 9 es un ejemplo.
- ◆ El capítulo 10 describe las instrucciones para la DBD, que indican incluir o no ciertas definiciones a tiempo de compilación; y ayudan en aplicaciones que se tendrán en varias versiones.
- ◆ El capítulo 11 es una descripción del funcionamiento interno del módulo DBD para que el programador lo conozca.
- ◆ El capítulo 12 son los valores máximos en DBD.

## 1) ESTRUCTURA DE LOS ARCHIVOS.

---

### 1.1) Registros y Dependencias.

En **SISINF** los archivos se manejan bajo el concepto de Banco de Datos, por lo que será necesario conocer su estructura.

Para describir mejor ciertos conceptos suponga que se desea diseñar un archivo de clientes. Lo primero que se podría pensar es en **toda** la información que estará asociada a un cliente como puede ser su número, su nombre, sus facturas, sus pagos de cada factura, sus pedidos, los renglones de cada pedido, sus estadísticas, etc. El problema hasta el momento es cómo organizar la información que se mencionó.

Para ello se puede pensar en tener un registro maestro del cliente con la información general del mismo como puede ser su número, su nombre, etc., y **dependiendo** de este registro, el resto de la información.

Del registro maestro del cliente dependerán los registros con la información de las facturas, de las cuales podrá tener en un momento dado, cero, uno o varias, por lo que se llamará **recurrente de facturas**.

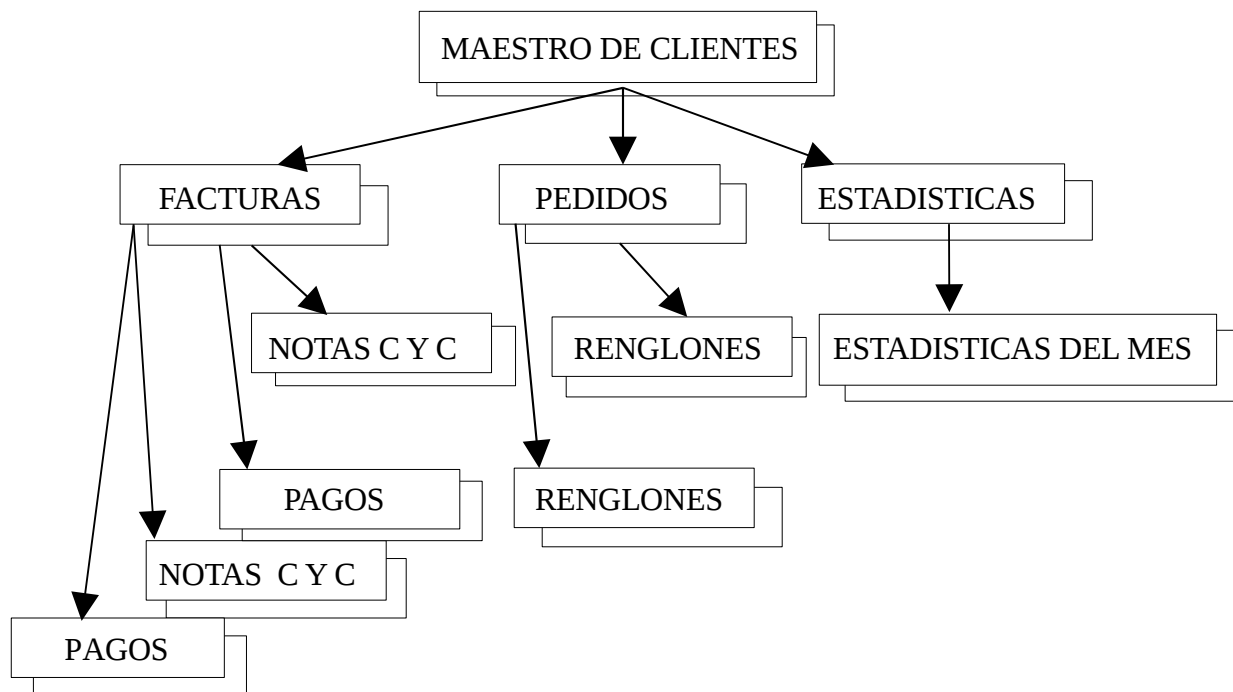
De una factura dependerán los registros con la información de los pagos efectuados, de los cuales puede haber cero, uno o varios. A estos registros se les llamará el recurrente de pagos.

De igual forma, de una factura dependerá los registros con la información de las notas de cargo o de crédito. A estos registros se les llamará el recurrente de notas de cargo o crédito.

Por otro lado, dependiendo del registro maestro del cliente se tendrán sus registros de pedido, (recurrentes de pedidos) y dependiendo de cada pedido se tendrán los registros de renglón del pedido (recurrente renglón de pedido).

Por último, y también dependiendo del registro maestro, se tendrán sus recurrentes de estadísticas.

Gráficamente el archivo de clientes quedaría:



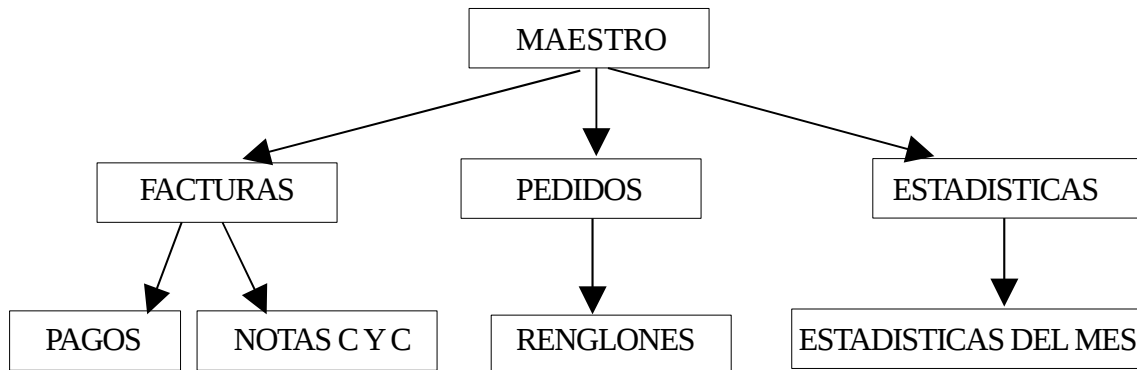
### ESTRUCTURA DEL TIPO NAVEGACIONAL

Enfatizando en lo descrito anteriormente, se tienen los siguientes comentarios a la gráfica:

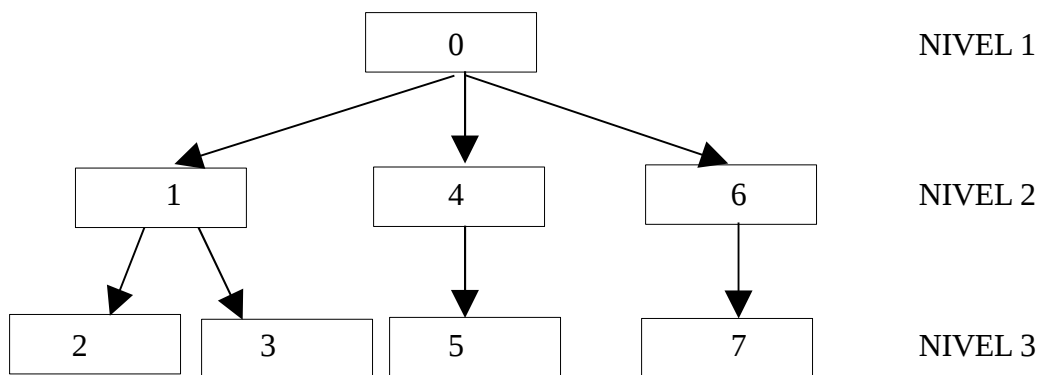
- ◆ De **un** registro maestro dependen sus facturas, que pueden ser cero, una o más.
- ◆ De cada factura (recurrente de factura) dependerán sus pagos (recurrente de pagos) que pueden ser cero, uno o más.
- ◆ De cada factura dependerán sus notas de cargo y crédito (recurrente de notas de C.C.) que pueden ser cero, una o más.

De igual forma se puede decir de los demás recurrentes.

Puesta la gráfica anterior en otra forma, quedaría:



Y en forma de árbol:



En donde:

El recurrente tipo 0 es el registro maestro del cual dependen los demás.

- ◆ El recurrente tipo 1 es el de facturas y de él dependen notas de cargo y crédito, así como los pagos.
- ◆ El recurrente tipo 2 son los pagos.
- ◆ El recurrente tipo 3 son las notas de cargo y crédito.
- ◆ El recurrente tipo 4 es el de pedidos y de él dependen los renglones de cada uno de estos.
- ◆ El recurrente tipo 5 son los renglones de los pedidos.
- ◆ El recurrente tipo 6 es el de estadísticas generales y de él dependen las estadísticas mensuales.

Es importante notar aquí que el recurrente 0 está en el **primer** nivel de la

estructura, los recurrentes 1, 4 y 6 están en el **segundo** nivel, y por último, los recurrentes 2, 3, 5 y 7 están en el **tercer** nivel.

En resumen:

- ◆ La estructura de un archivo está formada por un registro maestro y diferentes tipos de registros dependientes.
- ◆ Los registros dependientes pueden ocurrir cero, una o más veces por lo que se llamarán recurrentes. De estos recurrentes pueden depender otros recurrentes y así sucesivamente.
- ◆ La estructura descrita es del tipo navegacional.

## 1.2) Llaves.

Una vez que se tiene la estructura de los registros se deberá especificar qué variables son llaves, es decir, variables cuyo contenido sea sujeto de búsqueda. Dicho de otra forma, variables que conociendo su valor, permitan llegar al resto de la información.

Si se continúa con el ejemplo anterior se puede pensar que el registro maestro deberá ser buscado por el **número del cliente**, los recurrentes de facturas por el **número de factura** y los recurrentes de pedidos por el **número de pedido**.

Dado lo anterior, estos campos se deberán declarar como llaves. Es decir, si se conoce el número de un cliente, se puede llegar al resto de la información relacionada con dicho cliente. Si se conoce el número de una factura, se puede llegar a conocer el cliente a que pertenece, y en forma automática el resto de la información de dicho cliente. Si se conoce el número de pedido, se puede llegar a conocer el cliente a que pertenece, y en forma automática al resto de la información de dicho cliente.

Dicho en otra forma, si el número de cliente es llave, se puede buscar en el archivo el registro maestro de dicho cliente. Si se conoce el número de un pedido se podrá buscar el registro del pedido y, en forma automática, se busca el registro del cliente que tiene dicho pedido, de forma que podrá acceder todo el resto de la información del mismo cliente.

A la forma de BUSCAR descrita anteriormente se le llama PRINCIPAL, es decir busca un registro que tenga una llave. Note que no se permiten duplicados en las llaves, es decir, dos clientes no pueden tener el mismo número.

Otra forma de búsqueda es la SECUENCIAL, en donde se presentan los registros por el orden ascendente de la llave. En esta forma de trabajo si se BUSCA en forma secuencial por el registro maestro se accederá por orden del número del cliente y una vez buscado cada registro se podrá acceder toda la información de un cliente.

Si se BUSCA en forma secuencial por el recurrente de pedidos, se accederá por orden ascendente del número del pedido y una vez que se busca cada registro de pedido, se busca también en forma automática el registro maestro del cliente, por lo que se podrá acceder toda la información del mismo.

Para el caso de cliente se podrán tener otros campos como llave; tal sería el caso de la zona en el registro maestro o la fecha del pedido en caso del recurrente de pedidos. Al hacer esta definición se podrá BUSCAR por todos los clientes de una cierta zona o BUSCAR todos los pedidos de una cierta fecha. A esta forma de búsqueda se le llama NO-PRINCIPAL.

Las llaves que se usan para búsqueda NO-PRINCIPAL se asocian a llaves cuyas búsquedas se harán por PRINCIPAL, de forma que al buscar por NO-PRINCIPAL el orden en que se presentan los registros dependerá de la llave completa, es decir, si las llaves se definieron como número del cliente (la primera) y zona (la segunda) el orden dentro de una cierta zona es por el número del cliente. Si como llaves se define el estado (la primera), el número del cliente (la segunda), y la zona (la tercera), el orden en el que se presentarán los registros de una cierta zona será para el primer estado, los clientes ordenados por número, luego para el siguiente estado, los clientes ordenados por número, etc., la selección del orden de las llaves para este tipo de búsqueda dependerá de los reportes que se deseen.

En muchos casos, la llave no se forma de varias variables como en el ejemplo anterior para propósitos de búsquedas NO-PRINCIPAL, tal sería el caso del archivo de cuentas de contabilidad en el que se podrán tener como llaves la cuenta, la sub-cuenta y la sub-sub-cuenta. En muchos casos se buscará PRINCIPAL por las tres y se tendrá el registro que cumpla por igual con todas; en otros casos, se buscará por cuenta y sub-cuenta, y se tendrá el registro que cumpla por igual en ambas variables llaves y tenga el valor menor en la sub-sub-cuenta; y en otros casos se buscará por cuenta y se tendrá el registro que cumpla por igual con la cuenta y tenga el valor menor en la sub-cuenta y sub-sub-cuenta.

Otro caso de la discusión anterior es aquel en el que un archivo de clientes sirve a varias compañías, en cuyo caso las llaves de búsqueda PRINCIPAL estarán formadas por número de la compañía y número del cliente, número de la compañía y número de la factura, etc., y las búsquedas para procesar una compañía serán NO-PRINCIPAL teniendo como llave el número de la compañía.

En resumen:

- ◆ En **SISINF** se agrupan una o más variables llave del mismo registro para formar un grupo de llaves.
- ◆ La llave formada por el grupo de llaves tiene que ser única para ese grupo. NO puede haber duplicados.
- ◆ En un grupo de llaves se puede buscar en forma SECUENCIAL, PRINCIPAL y NO-PRINCIPAL.

### 1.3) Area en Disco.

Un dato que es necesario especificar es cuántos registros maestros tendrá un archivo y cuántos registros en promedio se espera que tenga cada recurrente. Con esto el módulo DBD calcula el área total que usará el archivo.

Cuando se ejecuta el módulo DBD se puede conocer con la opción A cuánto ocupará en disco el archivo así como cada grupo de llaves y cada recurrente. Con lo anterior se puede revisar que realmente se tenga dicha área y que este consumo sea justificado.

Continuando con el ejemplo del archivo de clientes, los datos que se necesitan conocer son:

- ◆ Cuántos clientes se tendrán en el archivo?
- ◆ Cuántos pedidos tiene en promedio un cliente?
- ◆ Cuántos renglones tiene en promedio un pedido?
- ◆ Cuántas facturas tiene en promedio un cliente?
- ◆ ...

En forma ordinaria el encargado del sistema manual conoce o tiene información para responder a las preguntas anteriores. Pero será necesario tomar en cuenta ciertas **características de operación** del sistema computacional para poder estimar mejor estos promedios. Algunas consideraciones para este ejemplo pueden ser:

- ◆ El número de clientes es un dato que la empresa lo conoce, o bien conoce sus crecimientos y el archivo tendrá capacidad para futuras expansiones.

- ◆ El número de renglones promedio de un pedido es un dato que se puede calcular con bastante precisión.
- ◆ Para el número de pedidos es necesario tomar en cuenta que un pedido se tecléa al recibirse, luego se surte, se factura y en este momento se generan estadísticas y se borra, es decir es necesario considerar características de operación, por lo que la pregunta es: Cuántos pedidos **activos** tiene en promedio un cliente?.
- ◆ Si se tienen 1000 clientes y 2 pedidos en promedio, significa que se podrán tener 2000 pedidos activos en el archivo, de forma que varios clientes pueden tener cero pedidos, otros solamente 1, otros 2, otros 3, etc., pero en promedio se tendrán 2 pedidos activos por cliente.
- ◆ Las notas de cargo y crédito, así como los pagos necesitan aparecer en reportes mensuales, por lo que será hasta fin de mes que se puedan borrar.
- ◆ Las facturas no se pueden borrar hasta haber sido liquidadas y aparecer en reportes de fin de mes.

Los siguientes comentarios ilustran algunos problemas que se pueden derivar de un mal cálculo:

- ◆ El tamaño del archivo crece en forma proporcional al número de registros en el recurrente cero. Así, para el ejemplo descrito, tener 2000 clientes en vez de 1000 origina un archivo del doble del tamaño y posiblemente no se tenga el área en disco necesaria.
- ◆ Los directorios de los grupos de llave se calculan en base al número de registros especificados. Por el algoritmo usado internamente, en la situación más desventajosa, si el número de llaves excede este valor se dará diagnóstico y será necesario reconstruir el archivo.
- ◆ Así, para este ejemplo, si se ponen 1000 clientes, 2 pedidos en promedio, el directorio para la llave número de pedido aceptará en la situación más desventajosa 2000 llaves.
- ◆ En recurrentes inferiores, un mal cálculo origina que se use mucha área en disco. Así, para el ejemplo descrito si se tienen 1000 clientes, 2 pedidos promedio por cliente, y 30 renglones promedio por pedido, se tendrán 60,000 registros del recurrente de **renglones**, pero si se especifican 40 renglones en promedio se tendrán 80,000.

En resumen:

- ◆ Estimar lo más cercano a la realidad el número de registros maestros, así como los promedios de los recurrentes.
- ◆ Revisar con opción A el consumo total y parcial (de cada recurrente) para ver si se justifica y se tiene el área disponible.
- ◆ Se recomienda ver el capítulo 12 para los valores máximos que puede tener un archivo ya que puede afectar el diseño del mismo.
- ◆ El comportamiento del área usada por un archivo permanente puede cambiar a través del tiempo, pero mediante el uso del módulo RECA se puede ir aumentando o disminuyendo dicho consumo. Se sugiere revise la documentación de dicho módulo.

#### **1.4) Consumo de CPU.**

En el diseño de un archivo es muy importante el tiempo que se usará de CPU para realizar diferentes funciones, ya que esto afectará el tiempo de respuesta es decir el tiempo entre dar la información a la máquina y que ésta responda con el resultado.

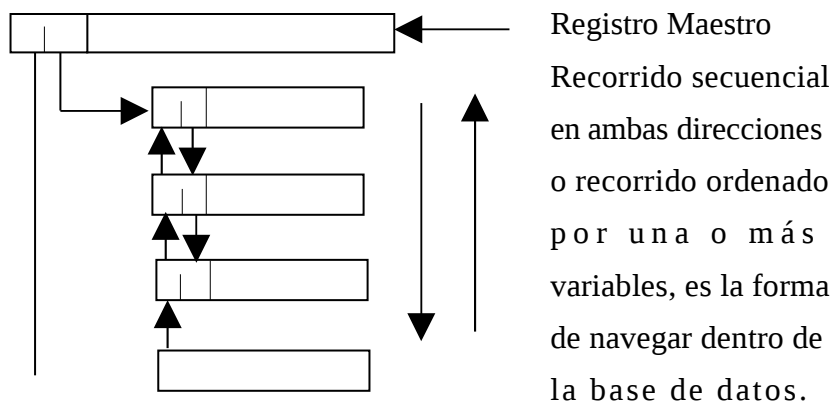
Los siguientes comentarios son aplicables a los grupos de llaves:

- ◆ El tiempo de ejecución en la instrucción de INSERTAR está formado por dos factores, uno el tiempo de escribir la información y otro el tiempo de actualizar los directorios para poder hacer las búsquedas. Si se tienen varios grupos de llaves en un registro, será necesario un directorio por cada grupo y habrá que actualizar cada uno. (El tiempo por este factor varía en forma lineal).
- ◆ Si al hacer una búsqueda PRINCIPAL por varias llaves se brinca alguna del grupo de llaves, el tiempo usado para efectuar la instrucción puede ser mayor del deseable, si de las variables omitidas existen muchas con valor igual en el archivo. Tal sería el caso del ejemplo que se está viendo en el que se defina un grupo de llaves como estado, número del cliente y zona, (en este orden), y luego se haga una búsqueda PRINCIPAL en este grupo de llaves por el número del cliente exclusivamente brincándose el estado. La solución al problema es la siguiente: Si es necesario definir un grupo de llaves como el mencionado con las variables de estado, número del cliente, y zona para propósitos de búsqueda SECUENCIAL o NO-PRINCIPAL será necesario definir en otro grupo de llaves como primera llave el número del cliente para que con este grupo se hagan las búsquedas PRINCIPAL.

- ◆ Por la forma de hacer las búsquedas NO-PRINCIPAL la forma más eficiente es que esté definida la llave que se va a usar más, como la primera dentro del grupo. Así por ejemplo, si se tienen un grupo de llaves definido como número del cliente y zona, y se hacen búsquedas. NO-PRINCIPAL por zona, el tiempo de respuesta no será adecuado si hay muchas zonas diferentes (500 clientes y 50 zonas), en este caso se deberá definir un grupo de llaves como zona y número del cliente.
- ◆ Dado que el uso de grupos de llaves para búsquedas NO-PRINCIPAL tiene ventajas de programación pero desventajas de área en disco y tiempos de inserción, se recomienda estudiar otras soluciones como son el uso de archivos secuenciales (llamados en **SISINF** temporales) los cuales se pueden ordenar por una o más variables, o bien con las búsquedas en recurrentes ordenados por una o más variables.

Otro aspecto importante es cómo están organizados los recurrentes de un archivo.

Básicamente los registros recurrentes dependen de un registro maestro o de otro recurrente; la forma de hacer esta dependencia es por medio de apuntadores. Por otro lado, los recurrentes se pueden **recorrer** en forma secuencial en un sentido o en otro. Esto se logra también por apuntadores. Gráficamente se tendrá:



La razón de tener apuntadores es por que permite lo siguiente:

- ◆ De un tipo de registro de recurrente puede haber cero, uno o varios y solamente usar el área de disco necesaria.
- ◆ Se puede hacer inserciones en cualquier momento ya que físicamente no están gravados en forma continua.
- ◆ Un registro maestro puede no tener ningún registro recurrente de cierto tipo y otro registro maestro puede tener muchos del mismo tipo. Es decir,

no es necesario que todos tengan el mismo número.

Cuando los registros recurrentes de un cierto tipo se desea que sean manejados a través de apuntadores será necesario especificar la opción P (Promedio).

En otros casos las características del tipo de recurrente son diferentes y de antemano se sabe que para todos los registros maestros dependerán un cierto número de recurrentes de cierto tipo. En este caso se deberá especificar la opción M (máximo).

La forma de operar la opción M dentro de **SISINF** es, los registros del recurrente se encuentran en el registro del cual dependen. Gráficamente queda:

REGISTRO MAESTRO O RECURRENTE REC REC REC REC
-----------------------------------------------

La ventaja de usar esta opción es que es más eficiente en tiempo de acceso, pero limita el número de registros. Por otro lado, si se especifica un recurrente con opción M no se podrán tener en él dependientes, ni grupos de llaves.

Es importante notar que el hecho de poner P o M **no afecta** la forma de programar ya que **SISINF** lo controla internamente.

Para el ejemplo que se ha usado para el caso del registro de facturas, se deberá especificar como P, en cambio las estadísticas mensuales que de antemano se sabe que todos tendrán 12 se deberá de poner M.

Dado que los recurrentes opción M forman parte del registro del que dependen, NO se recomienda que dependan del R0 si su uso va a ser muy esporádico y el número de variables que tengan originen un registro muy grande. Ya que en este caso se estará leyendo, escribiendo o insertando mucha información que no se usa.

### 1.5) Archivos mayores a 2G.

Hasta la versión 7.2, el tamaño máximo de un archivo permanente era de 2G, en las versiones posteriores se amplió a 32G, la forma de hacerlo se describe a continuación.

Dentro de un archivo permanente, si el área de datos se ve como una secuencia de bytes, SISINF necesita saber donde inicia cada R0, cada R1,... es decir su dirección dentro del archivo.

Internamente se usan 32 bits para la dirección de los RX, de estos el primero es el signo, los siguientes 23 es el número del registro y los últimos 8 son el desplazamiento dentro del registro. Los registros son de 512 bytes. Así Sisinf para leer un R0, R1,... conociendo su dirección, saca el número de registro, lo lee de disco y luego con el desplazamiento dentro del registro saca la información. Usando el formato anterior se tiene la capacidad de 2G.

Para resolver el problema de aumentar la capacidad se puede pensar en usar el bit de signo para el número del registro y así tener 24 bits pasando a 8G la capacidad.

Pero para aumentar mas todavía esta capacidad, se puede pensar en usar solo 6 bits para el desplazamiento y dejar 26 para el número del registro. Ahora bien si se reduce a 6 bits el desplazamiento solo tenemos un número del 0 al 63, la pregunta es como acceder los 256 bytes del registro. La solución es que el desplazamiento será múltiplo de 4, es decir un 0 accedera el byte 0, un 1 el byte 4, un 2 el byte 8,...

El problema de esta solución es que cuando se inserten el R0, R1,... el desplazamiento debe ser múltiplo de 4 pudiendo desperdiciar bytes. Así si el último RX insertado termina en el byte 5 de un registro, el siguiente inicia en el byte 8. El desperdicio promedio es de 2 bytes por cada R0, R1,... Claro cuando se manejen imágenes, texto o RX grandes, este desperdicio es mínimo en comparación al tamaño.

El crecimiento de los archivos permanentes en el área de datos es automático, es decir se agranda cuando se inserta nueva información teniendo como límite 2G o 4G dependiendo de la máquina y la versión de Sisinf.

Si se tienen archivos que no pasen de 8G este cambio es transparente, pero si un archivo va a crecer más de 8G, será necesario primero reconstruirlo ya sea con RECA o con OPCION DISCO RECA y darle la opción de tamaño por porcentaje de forma que exceda los 8G, con esto se pone un **indicador** y el formato del direccionamiento es el último que se describió. Si con la definiciones de DBD es de más de 8G al crearse se pone el indicador.

Con versiones posteriores a 7.2, si el archivo no tiene el indicador, el límite es de 8G, es decir al tratar de pasar el límite se diagnosticará como error. Si tiene el indicador, el límite es de 32G.

Los archivos permanentes de mas de 2G/4G no podrán ser usados en versiones de Sisinf anteriores ya que no tendrán el manejo del nuevo formato ya descritos.

Se pueden usar los comandos de listar información de directorios para

conocer el tamaño de un archivo.

En algunas máquinas, por restricciones en las llamadas al Sistema Operativo y formatos de directorios no se podrán acceder archivos mayores a 2G o 4G aún con la última versión. Ejemplo: en pruebas hechas en una PC con FAT32 no permite más de 4G.

## 1.6) Resumen.

Se tienen las siguientes definiciones del banco de datos de **SISINF**:

**MAESTRO** Registro con información general del cual dependen los demás registros con la información de detalle.

**RECURRENTE** Registro con información de detalle que puede ocurrir cero, una o más veces y que depende de un registro maestro o de otro recurrente.

**GRUPO DE LLAVES** Varias variables que forman una llave única y son sujetos de búsqueda.

**PRINCIPAL** Se busca desde el inicio hasta encontrar el primer registro que cumpla por igual a las variables llave que se especifican.

**SECUENCIAL** Se accesa registro por registro en el orden en que se genera por grupo de llaves.

**NO-PRINCIPAL** Se busca desde donde está posicionado el archivo hasta encontrar un registro que cumpla por igual con las llaves que se especifican.

Los pasos generales para hacer el diseño son:

- ◆ Definir la estructura de los archivos, determinando cuáles variables son del maestro, y cuáles de los recurrentes. Una forma de revisar si es adecuado es preguntar de quién depende la información (ejemplo; renglones del pedido; dependen del pedido).
- ◆ Especificar qué registros se necesitan buscar por **PRINCIPAL** y cuáles son sus llaves.
- ◆ Especificar los reportes o consultas que impliquen un orden especial, para definir llaves de búsqueda **NO-PRINCIPAL** o **SECUENCIAL**. Teniendo el dato de la frecuencia de uso para revisar otras opciones.

- ◆ Definir los grupos de llaves considerando los tiempos de respuesta y que la llave formada no puede estar repetida.
- ◆ Definir qué recurrentes son opción P y cuáles M.
- ◆ Definir el número de registros maestros (recurrente cero) que se tendrán, así como el promedio o máximo de los registros recurrentes.
- ◆ Revisar con opción A de DBD si el consumo de área en disco es el adecuado.

## 2) TIPOS DE CAMPO.

### 2.1) Características.

El diseñador de la aplicación, deberá seleccionar de entre los 10 **tipos de campo** que tiene **SISINF**, cuál es el adecuado para cada variable de sus archivos. Los tipos de campo son:

N1	Campo numérico de 1 a 4 dígitos.
N2	Campo numérico de 1 a 8 dígitos.
N3	Campo numérico de 1 a 12 dígitos.
N4	Campo numérico de 1 a 16 dígitos.
X	Campo alfanumérico de 1 a 120 caracteres.
F	Campo fecha.
S	Campo SI-NO.
T	Campo texto.
I	Campo Imagen.
A	Campo Audio.

A continuación se describen las características principales de cada tipo para que con esto se seleccione el más adecuado y las opciones en la definición de los mismos.

**Los campos tipo N** son aquellos que solo tendrán información numérica se deberán especificar como N1, N2, N3 o N4 dependiendo de los dígitos que se desee almacenar. Las características principales son:

- ◆ Cuando se leen de la terminal se revisa en forma automática que sólo se teclee información numérica.
- ◆ Se puede especificar que se validen en la lectura por un cierto rango (R o RANGO), por dígito verificador (DV) o que sean igual a algún valor de la primera columna de una tabla (V o VALOR).
- ◆ Se puede efectuar operaciones aritméticas con ellos en los programas.
- ◆ Se pueden editar de acuerdo a una máscara (M1 a M15 y MX).

**Los campos X** están pensados para tener información de letras, números y caracteres especiales. Solo se puede especificar la validación por dígito y por valor. Se puede especificar que si se teclean minúsculas estas se conviertan a MAYUSCULAS. Este campo se usa para nombres, direcciones, ...

**Los campos F** están pensados para almacenar fechas. Las características principales son:

- ◆ Se leen de la terminal en formato de fecha, como día/mes/año y se revisa en forma automática que la información sea numérica, en el rango adecuado de día, mes y año, año bisiesto,...
- ◆ Internamente se almacenan como XXXXYYZZ en donde XXXX es el año (0 es el año 0), YY es el mes y ZZ el día.
- ◆ Se pueden especificar que cuando se lean se validen por rango (R o RANGO) entre dos fechas o bien sea igual a algún valor de la primer columna de una tabla (V o VALOR).
- ◆ A una fecha se le pueden sumar día, meses o años y obtener otra fecha. También se pueden restar dos fechas para obtener días, meses o años.
- ◆ Se pueden editar de acuerdo a una máscara (M16 a M18).

**Los campos S** se valida al leerlos por la terminal que sólo se teclee SI o NO, sirven para variables en los que de antemano se conoce que sólo puede tener estos valores. Este campo se cambia en forma automática de minúsculas a mayúsculas.

**Los campos T** están pensados para tener información alfanumérica que será desplegada o impresa en uno o varios renglones. Las características principales son:

- ◆ Se pueden formar en ejecución con una o más literales y con una o más variables del tipo N1, N2, N3, N4, X, S, F o T. Lo anterior permite que se inserte información variable en un texto, así como que por lógica de programa se incluyan ciertos párrafos.
- ◆ Se puede especificar la indentación de párrafo (DIP) y la indentación de inciso (DII) que tendrá el texto, tanto en la definición del campo como cuando se forma en ejecución.
- ◆ En ejecución si una palabra del texto ya no cabe en la línea actual, en forma automática, se pone en la siguiente línea con la indentación que se indique.

- ◆ Se especifica con la opción de AREA el área en disco que se espera que usen.

Estos campos se pueden usar para descripciones, documentos, notas aclaratorias, ayudas, ... es muy importante señalar que el consumo de CPU así como de memoria en los programas es MUCHO MAYOR en los campos Texto que en los X.

**Los campos I** están pensados para guardar imágenes como fotos, firmas, documentos, etc. También se pueden generar gráficas. Las principales características son:

- ◆ Se guardan tal cual se generaron.
- ◆ Se puede especificar con la opción de AREA, el área que se usará en disco.
- ◆ Solamente se pueden usar junto con el Emulador SISINF en cualquier sistema operativo.

Los campos Imagen se tienen en SISINF desde la V4.0 solo en PC, a través del tiempo se han ido modificando y las opciones de MEDIDA, COLOR, ... se pueden poner pero no son necesarias y algunas de ellas NO se toman en cuenta. Así poner BN (Blanco y Negro) NO convierte una imagen a colores. Las imágenes generadas con la versión de ELS de PC deben ser cambiadas al uso de ELSI – Emulador. Para que sea aplicable lo descrito en los manuales de esta versión. La forma de hacerlo es usando ELS pasar las imágenes a archivos temporales y luego leer dichos archivos pero con ELSI - Emulador SISINF.

**Los campos A** están pensados para guardar audio, tal como voz, música, etc. Estos campos se tenían en SISINF desde V4.0 en PC, pero eran muy dependientes de hardware, actualmente están discontinuados.

En resumen:

- ◆ Se deberá seleccionar el tipo de campo de cada variable de los archivos.
- ◆ Se deberá definir opciones adicionales de validación o características de los campos de acuerdo a las necesidades específicas.
- ◆ Los campos Imagen y Audio requieren de hardware especial y que éste sea compatible con **SISINF**.

## 2.2) Area en Disco en Campos Imagen.

Los campos imagen se guardan tal cual se leen del archivo en que fueron guardados por el dispositivo para digitalizarlas y solo es necesario especificar el AREA máxima que ocuparán en disco, los siguientes comentarios son aplicables:

- ◆ En dicho dispositivo se deben de hacer los ajustes de ser necesario para blanco y negro, 16 colores, ...
- ◆ Se deberá hacer pruebas con las imágenes a guardar y con esto determinar el AREA máxima a usar en disco.
- ◆ Las opciones disponibles en versiones anteriores como son LONG., MEDIDA, BN, COLOR, ... se pueden especificar pero sin efecto alguno.

## 2.3) Area en disco en campos texto.

Tal como se describió los campos texto están pensados para información alfanumérica que se desplegará o imprimirá en uno o varios renglones. Razón por la cual los tamaños se especifican en múltiplos de 512 caracteres.

Básicamente es necesario especificar el tamaño máximo que tendrá el texto para que con esto se separen áreas internas de trabajo en los programas y el tamaño promedio que se espera que tenga, para calcular el área en disco necesaria por el texto.

La forma de calcular éste último es estimar el total de caracteres del texto y esto dividirlo entre 512 redondeado hacia arriba. Posteriormente se sugiere revisar con el módulo RECA este dato para ajustarlo a lo real.

Así por ejemplo para guardar ayudas de 30 columnas por 20 renglones que se desplegarán en un máximo de 10 páginas y un promedio de 3 páginas, se pondría:

AREA 12 4

Se calculó (30 columnas x 20 renglones x 10 páginas)

$$(30 \times 20 \times 10) / 512 = 12$$

Se calculó (30 columnas x 20 renglones x 3 páginas)

$$(30 \times 20 \times 3) / 512 = 4$$

En resumen para campos texto se necesita especificar el consumo máximo y promedio en múltiplos de 512 caracteres.

### 3) DOCUMENTACION.

En **SISINF** se tiene un módulo de Documentación el cual sirve para la documentación de programación que ayuda en la etapa de **desarrollo** y que es muy necesaria en la etapa de **mantenimiento**.

Básicamente se propone que la documentación de los archivos este junto con las definiciones de los mismos, para ello los registros que se inicien con \*D significará que son comentarios con la descripción de información del archivo, campos,... Para ser usados por el sistema de documentación.

Además de lo anterior en varias de las declaraciones para la DBD se incluye información para éste módulo, las cuales junto con las de \*D sirven para el **reporte de documentación de programación**.

Para ordenar mejor la información de documentación se deberá:

- ◆ Documentar los archivos, campos y tablas con \*D.
- ◆ Definir los sistemas y grupos de programas.

## 4) SEGURIDAD.

Todos los usuarios de los diferentes módulos de **SISINF** deben dar una clave de acceso, misma que es la que autorizará o no autorizará diferentes funciones, por razón de seguridad de la información.

Para efectos del módulo DBD todos los campos tiene asignado un número de seguridad en lectura y otro de escritura. Dichos números van de 0 a 10 en donde 0 es el nivel de mayor seguridad y 10 el de menor.

En forma opcional se puede asignar a un campo un número de seguridad entre 1 y 9999 para identificar a los usuarios que puede modificar dicho campo.

La clave de acceso de cada usuario tendrá:

- ◆ Nivel de seguridad de lectura.
- ◆ Nivel de seguridad de escritura.
- ◆ Números de seguridad. (SEG.)

Con esto se tiene la protección suficiente para restringir lo que se puede ver y lo que se puede modificar.

Si la empresa tiene ciertos esquemas organizacionales es fácil la asignación de los niveles y números.

En resumen, **SISINF** cuenta con seguridad a nivel de campo, tanto en lo que los usuarios puedan ver como lo que pueden modificar.

## 5) CAS.

El módulo CAS sirve para que usuarios con pocos conocimientos sobre programación puedan consultar la información de los archivos.

El módulo CAS propone un proceso de selección para que el usuario final consulte su información. Para este proceso de selección se necesita que las descripciones de nombres de archivos, recurrentes, etc. sí le den una idea de lo que se va a seleccionar. Dado que los nombres de archivo tienen un máximo de 6 caracteres, los recurrentes se manejan con un número, ... se pensó en **nombres extendidos** que realmente den al usuario mayor idea de lo que selecciona. Dichos nombres se especifican en varias de las declaraciones de la DBD y se deben incluir para poder usar el módulo CAS.

De la misma forma se pensó que después de seleccionar el o los archivos que desea consultar, seleccione **Grupos** de Variables o Campos. En esta forma de trabajo se necesita definir los grupos de información que tiene un archivo y luego a cada campo se le deberá de asignar el GRUPO al que pertenece con la definición de GRUPO, los campos que no tengan esta definición no será factible consultarlos.

Como el CAS está hecho para usuarios finales se diseñó para que al teclear ? se pudiera dar Ayudas que de ser posible fueran lógicas para el usuario. Para ello al definir un campo se puede especificar si el campo se puede acumular o es factible calcular su promedio, si se puede seleccionar, etc... Lo anterior se hace al definir el campo, con la opción CAS. Se puede hacer para los más usados.

Por último es muy común en los sistemas computacionales tener tablas que relacionen números con nombres, tal sería el caso de una tabla de estados donde el número 1 puede ser NUEVO LEON, el 2 JALISCO, ... Si se desea que ésta conversión la efectúe también el CAS es necesario especificarlo en la definición de la tabla.

En resumen para un mejor\_uso de CAS se deberá:

- ◆ Poner nombres extendidos en donde se requiera.
- ◆ Definir GRUPOS de cada archivo que se desee consultar y asignarlos a los campos.
- ◆ Definir opciones de CAS para mejores ejemplos, así como la conversión de una tabla.

## 6) VARIABLES DE PROGRAMAS.

En las definiciones de DBD también será necesario especificar las variables de trabajo que se usarán en los programas y que no pertenecen a ningún archivo. Para ello se dispone de la declaración de AREA y a continuación los campos en el mismo formato que cuando se definen para un archivo.

Es importante señalar que en los programas se tiene la declaración de IGUAL por lo que se pueden definir en DBD variables de área de trabajo de diferentes tipos y luego en los programas mediante el IGUAL definir las que se requieran.

Un ejemplo de lo anterior es definir una variable de trabajo N1 y en los programas hacer varios iguales a esta variable para los indicadores que sean necesarios.

Si a un área de trabajo se le asigna un nombre se indicará que los valores de las variables de dicha área se podrán pasar entre programas. Un ejemplo de lo anterior es en un sistema de contabilidad en el que al inicio de la ejecución se pida el número de la compañía en la que se trabajará y este dato se requerirá en varios de los programas.

En una DBD se puede tener una o varias áreas de trabajo con nombre dependiendo de las necesidades de los programas. Pero es importante señalar que al pedir en el programa que se incluyan las variables de un área de trabajo TODAS las que forman el área se tendrán en el programa, pudiendo ocasionar problemas de falta de memoria o de exceso de variables.

Note que se permite que un nombre de campo este en varias áreas con nombre pero estas NO se podrán usar en forma simultánea en un mismo programa.

En resumen:

- ◆ Se deberá definir las variables de trabajo de los programas.
- ◆ Se pueden definir una o varias áreas de trabajo con nombre cuyas variables se pasan entre programas.

## 7) DEFINICIONES DE DBD.

### 7.1) Introducción.

Existen 9 tipos de definiciones en el módulo DBD, las cuales se describen a continuación.

- ◆ **ARCHIVO.** Esta definición sirva para especificar las características generales de un archivo, así como para indicar que las definiciones de campos o recurrentes que siguen a continuación pertenecen a dicho archivo.
- ◆ **RECURRENTE.** Se define la estructura de los recurrentes en cuanto a número promedio de registros y recurrentes dependientes.
- ◆ **AREA.** Se define que las variables que se especifican a continuación son campos que se usan en los programas del Lenguaje **SISINF** y no pertenecen a ningún archivo, es decir, son variables o campos de programa. Es posible que un área tenga nombre en cuyo caso el contenido de las variables que la forman se podrán pasar entre programas.
- ◆ **IGUAL.** Esta definición sirve para indicar que la información dada para un campo de otro archivo es igual a la del campo que se especifica con esta definición de IGUAL.
- ◆ **CAMPO (o variable).** Se definen las características de los campos o variables de los registros de un archivo.
- ◆ **TABLA.** Se define el nombre y estructura de una TABLA de valores, asimismo se indica que los valores de cada renglón siguen a continuación hasta encontrar otra definición.
- ◆ **DOCUMENTACION.** Se define que esta DBD se quiere documentar con la opción de DOCUMENTACION **SISINF**. A continuación se tiene información para la documentación.
- ◆ **GRUPO.** Se definen para el módulo CAS el nombre de un grupo de variables.
- ◆ **\*D.** Información para Documentación de programación con la descripción general de archivos, recurrentes, campos, tablas,...

Ejemplo de la secuencia que deben seguir las definiciones en DBD:

ARCHIVO                    Se define un archivo  
     RECURRENTE                    Se define la estructura  
     RECURRENTE  
     CAMPO                    Se define  
     CAMPO                    campos o variable  
     CAMPO                    del archivo.

AREA  
     CAMPO                    Se define campos  
     CAMPO                    o variables  
     CAMPO                    de programas

ARCHIVO  
     RECURRENTE  
     CAMPO                    Se define  
     CAMPO                    otro archivo  
     CAMPO  
     IGUAL

AREA  
     CAMPO  
     CAMPO

TABLA  
     \_\_\_\_\_  
     \_\_\_\_\_

ARCHIVO  
     RECURRENTE  
     CAMPO  
     CAMPO  
     \_\_\_\_\_  
     \_\_\_\_\_

TABLA  
     \_\_\_\_\_  
     \_\_\_\_\_

AREA . . .

CAMPO                    Se define un área común

    CAMPO                    entre programas

    IGUAL

## 7.2) Reglas de Sintaxis.

Para especificar las definiciones de DBD en el archivo DBDSxx o en ISzzzz se deberán seguir las siguientes reglas:

- ◆ Entre cada palabra o número se deberá dejar al menos un espacio en blanco.
- ◆ Las palabras o números que se especifican, se pueden dar en cualquier columna pero respetando el orden indicado en la explicación de la definición.
- ◆ En números, el signo + y – puede ir junto o separado del número. El signo + puede omitirse.
- ◆ Las columnas en donde se puede especificar la información es de la 1 a la 70. Se diagnóstica si se usan más columnas.
- ◆ Si una definición no cabe en una línea, se deberá poner un \$ después de la última palabra o número dejando un blanco y continuar en la siguiente.
- ◆ Para letreros con blancos intermedios, el letrero se debe especificar entre apóstrofes.
- ◆ Un \* en columna 1 significa comentario.
- ◆ El nombre de un archivo, tabla, campo o área, no puede empezar con ' \$ : ; . & /.

## 7.3) Definición de ARCHIVO.

Esta definición sirve para especificar las características generales de un archivo, lo cual se hace por medio de esta definición; también sirve para indicar que las definiciones de campos y recurrentes que siguen a continuación pertenecen a dicho archivo.

**ARCHIVO** Nombre del archivo.

Número que indica el número de tipos de recurrentes.

(Sin contar el maestro o R0)

Número de grupos de llaves.

Dispositivo (Directorio, Grupo y Cuenta...) del archivo.

Nombre extendido para documentación y CAS. (Opcional)

Número de grupos para CAS. (Opcional)

MARCAR-ABIERTO. (Opcional)

En donde la palabra ARCHIVO debe aparecer tal y como se señala en el formato, y los campos posteriores se refieren a información del usuario tomando en cuenta que:

- ◆ El nombre del archivo se debe especificar de acuerdo a los reglas generales para definir archivos dentro del sistema operativo y no deberá exceder de 6 caracteres.
- ◆ El nombre del archivo no puede comenzar con T y a continuación un número.
- ◆ Si el número de tipos de recurrente que se especifica es n, significa que los tipos de recurrentes son R1, R2,... Rn.
- ◆ Si el número de grupos de llaves que se especifica es n, significa que los grupos de llaves son LL1, LL2, ... LLn.
- ◆ El número de tipos de recurrentes y el número de grupos de llaves puede ser cero.
- ◆ El dispositivo del archivo (directorio, grupo y cuenta, etc.), se deberá de dar de acuerdo a las reglas del sistema operativo. **NO** se revisa que exista o que sea correcto. Dado que es opcional si no se especifica se asume un parámetro de instalación.

El nombre extendido para documentación y CAS se utiliza por el sistema de documentación en varios reportes y consultas para dar mayor claridad a la información. El módulo CAS lo usa para que el usuario final pueda seleccionar el o los archivos con cuya información se generará su reporte o consulta.

- ◆ El módulo CAS en las ayudas antepone las palabras 'Para el archivo' al nombre extendido. Lo anterior se deberá tomar en cuenta al asignar el nombre.

- ◆ El nombre extendido debe ser de 1 a 40 caracteres.
- ◆ El número de grupos para CAS es el número de definiciones de GRUPO que tendrá el archivo, si es cero o se omite, el archivo no se podrá consultar con CAS.
- ◆ La opción de MARCAR-ABIERTO sirve para poner una marca en el archivo que se usa para que si por algún motivo NO se cierra en forma normal, al tratar de usarlo nuevamente se dará error. (Esta opción no esta disponible en todas las versiones de **SISINF**)

#### 7.4) Definición de **RECURRENTE**.

La definición de recurrente sirve para especificar la estructura del Banco de Datos.

**RECURRENTE**      Número del recurrente cuya estructura se define  
 (0 es el registro maestro).  
 Número máximo o promedio de registros.  
 El campo anterior es máximo (M) o promedio (P).  
 Número de recurrentes dependientes.  
 Número asignado a cada recurrente dependiente.  
 Nombre del registro para CAS. (Opcional)  
 Nombre del registro para documentación. (Opcional).

En donde la palabra **RECURRENTE** debe aparecer tal y como se señala en el formato, y los campos posteriores se refieren a información del usuario tomando en cuenta que:

- ◆ El número máximo o promedio de registros determina el área necesaria en disco para crear el archivo.
- ◆ La opción P o M involucra dos formas diferentes de proceso como se indicó en la sección 1.4, y no afectan a la forma de programar. En el caso del registro maestro debe ser M.
- ◆ El orden en que aparezca esta definición dentro de un ARCHIVO puede ser cualquiera, siempre y cuando sea antes de que una definición de campo haga referencia al recurrente.
- ◆ El nombre para CAS servirá para que este módulo use dicho nombre en

diferentes pantallas y se logren descripciones más claras. Como este nombre se inserta en diferentes textos, se recomienda se especifique en singular y se ajuste a una redacción del tipo "Para cada".

- ◆ Si el nombre para CAS se omite o se especifica en blanco y el archivo o recurrente se pueden consultar con CAS se dará error de ejecución en CAS como ERROR SISTEMAS: RECURRENTE SIN NOMBRE.
- ◆ El nombre para documentación sirve para hacer más clara la documentación y se usa en este sistema para varios reportes. Si se pone como \* se usará el de CAS.
- ◆ Los nombres de CAS y Documentación deben ser de 1 a 40 caracteres.

El siguiente ejemplo ilustra la definición de RECURRENTE y ARCHIVO; para ello suponga que se tiene la siguiente estructura: (la misma de la sección 1).

	0				0 MAESTRO
					1 FACTURAS
					2 PAGOS
1		4		6	3 NOTAS C Y C
					4 PEDIDOS
					5 RENGLONES DE PEDIDO
					6 ESTADÍSTICA ANUAL
2	3		5	7	7 ESTADÍSTICA MENSUAL

Las instrucciones serían:

```

ARCHIVO CLIEN  7 4  "          'CLIENTES' 4
RECURRENTE  0 1000 M 3 1 4 6 'CLIENTE'      *
RECURRENTE  1  5  P 2 2 3  'FACTURA'      *
RECURRENTE  2  3  P 0      'PAGO'          *
RECURRENTE  3  5  P 0      'NOTA DE CARGO O CREDITO' *
RECURRENTE  4  2  P 1 5    'PEDIDO'          *
RECURRENTE  5  30 P 0      'RENGLON DE PEDIDO'*
RECURRENTE  6  1  P 1 7    'ESTADISTICA ANUAL' *
    
```

RECURRENTE 7 12 M 0 'ESTADISTICA MENSUAL' \*

## 7.5) Definición de AREA.

La definición de AREA sirve para indicar que un grupo de variables que se especifican a continuación forman las variables llamadas de programa. El formato de esta definición es:

**AREA** nombre

A continuación de la definición de AREA se especifican los campos en el mismo formato que un campo de un archivo. Los siguientes comentarios son aplicables a la definición de AREA:

- ◆ Puede haber más de una definición de AREA en una DBD.
- ◆ Si el área tiene nombre, las variables que se especifiquen a continuación pertenecen a dicha área y su valor se pasa entre programas.
- ◆ El nombre del área debe ser de 1 a 6 caracteres y NO ser igual al de un archivo.
- ◆ Si el AREA no tiene nombre no se permite la instrucción de IGUAL.
- ◆ Si el AREA tiene nombre se permite la instrucción de IGUAL a un archivo.
- ◆ Si el mismo campo se define en un archivo y en varias áreas con nombre, dichas áreas NO se podrán tener en forma simultánea en un mismo programa.

En las áreas con nombre se guarda lo siguiente:

- ◆ El número de campos.
- ◆ Su tipo en memoria, decimales o longitud.
- ◆ Si tiene máscara M1 a M12 o MX.

Si se cambia alguno de ellos es necesario recompilar los programas que usan el área.

## 7.6) Definición de un Campo.

Esta definición indica a **SISINF** todas las características de los campos o variables que forman un archivo, no solamente se refieren al manejo de la información de los campos en el archivo, sino también en los programas; en cuanto a la forma de hacer cálculos, imprimir la información, leer la información, etc.

Datos **obligatorios** de la definición:

Nombre del campo

Letrero para terminal

Letrero para reporte

Recurrente a que pertenece

Tipo de campo en disco

Tipo de campo en memoria

Longitud o número de decimales

Datos **opcionales** de la definición:

Nivel de seguridad en escritura

Nivel de seguridad en lectura

Validación por rango

Validación por dígito

Validación por valor

Número de seguridad

Grupo a que pertenece la variable

Area a usar en disco

Definir Identación de Párrafo

Definir Identación de Inciso

Mayúsculas

Es importante señalar que con excepción de los niveles de seguridad todos los datos opcionales se pueden especificar en cualquier orden.

El formato de esta definición, es válido para campos de archivos como de Area.

### 7.6.1) Nombre del campo.

El **nombre del campo** se refiere al nombre que tendrá ese campo dentro del registro y el nombre que se usará en los programas dentro del Lenguaje **SISINF**, para el manejo de la información que contenga. Las reglas bajo las cuales se debe de formar el nombre son:

- ◆ Se puede formar con letras, caracteres especiales y números pero no puede haber blancos intermedios.
- ◆ No puede comenzar con +, -, punto decimal o número, ya que el sistema asume que lo que empiece con esto es información numérica y el nombre debe ser alfanumérico.
- ◆ Tampoco puede comenzar con \$ : ; & /.
- ◆ La longitud máxima del nombre es de 30 caracteres.
- ◆ No se permite que en un grupo de aplicaciones, es decir, en todos los archivos definidos en un DBDSxx, se tengan nombres de campos iguales. La única forma de tener dos variables con igual nombre es por medio de la definición de IGUAL.
- ◆ Si el nombre de un campo es igual a la palabra ARCHIVO, AREA, IGUAL, TABLA, RECURRENTE o GRUPO, se deberá especificar dicho nombre entre apóstrofes, si no se hace se asume que es la definición.

Algunos ejemplos de nombre de campo son:

VALOR	correcto
NUM-CLI	correcto
NUM/SUMA	correcto
NUM-CLI	correcto
FIN.\$	correcto
1PAGO	incorrecto
FECHA	correcto
TEL CLI	incorrecto
\$-FINAL	incorrecto
AREA	incorrecto

Para efectos de este manual, la palabra campo tiene el mismo significado que

variable y se usará uno u otro indistintamente.

### 7.6.2) Letrero de terminal y de impresora.

El letrero de terminal es el letrero que aparecerá en el CRT y que informa a la persona que proporciona la información qué es lo que se le está pidiendo.

El letrero de CRT se deberá formar de acuerdo a las siguientes reglas:

- ◆ Se deberá especificar entre apóstrofes y podrá tener cualquier caracter.
- ◆ Si el letrero CRT es igual al nombre del campo se podrá poner un \* y no especificar el letrero.
- ◆ Este letrero es el que usa el módulo CAS para que el usuario seleccione la variable.

Por ejemplo, si se especifica en una definición de campo lo siguiente:

NUM-CLI            'NUMERO DEL CLIENTE'

En cualquier programa de **SISINF** que se lea la variable NUM-CLI aparecerá en el CRT:

```
NUMERO DE CLIENTE
CURSOR
```

Al teclear el número de cliente, éste se almacenará en NUM-CLI.

El **letrero de impresora o de reportes** se refiere al letrero que se desea aparezca como encabezado en la impresión de reportes de este campo.

El letrero de reportes se deberá formar de acuerdo a las siguientes reglas:

- ◆ Se deberá especificar entre apóstrofes y podrá tener cualquier caracter.
- ◆ Si el letrero de impresora o de los reportes es igual al letrero de CRT se podrá poner un \* y no especificar el letrero.
- ◆ La **longitud** máxima de este letrero es de 20 caracteres. Se deberá tomar en cuenta a la hora de definir este letrero cuál es la longitud del campo al imprimir ya que si el letrero se define de 15 caracteres y el campo se puede imprimir con una longitud de 4, lo más probable es que se use una

**literal** en las instrucciones de impresión y nunca se use este letrero.

Suponga que se definen 2 variables: NUM-CLI y NOMBRE-CLI:

```
NUM-CLI      'NUMERO DE CLIENTE'    '# CLIENTE'
NOMBRE-CLI   'NOMBRE DE CLIENTE'    *
```

En un reporte donde se impriman las variables NUM-CLI y NOMBRE-CLI, aparecerá el encabezado.

# CLIENTE	NOMBRE CLIENTE
_____	_____
_____	_____
_____	_____
_____	_____

En la descripción de CLS se especifican las instrucciones para lograr lo anterior de programas en **SISINF**.

Note que el letrero de NUM-CLI se ha reducido porque al imprimir el contenido ocupa pocas columnas.

### 7.6.3) Recurrente y tipo de campo en disco.

El número del **recurrente a que pertenece** indicado con R0 si pertenece al registro maestro, R1 al recurrente 1, R2 al recurrente 2, ...

El orden en el que aparezcan los campos de R0, R1, etc. puede ser cualquiera, ya que la posición en el registro se maneja en forma interna y el usuario no tiene que preocuparse por ella.

El **tipo de campo en disco** es una clave que indica si el campo es simple o a qué grupo de llaves pertenece. Las claves válidas son las siguientes:

```
S      Simple
LL1    Grupo de llaves 1
LL2    grupo de llaves 2
...
```

Para el caso de llaves se deberán revisar las diferentes formas de buscar (ver sección 1) y deberán tomarse en cuenta las siguientes restricciones:

- ◆ La información contenida en una variable llave puede estar repetida en varios registros del mismo tipo.
- ◆ La llave formada por la información de todas las variables llave de un mismo grupo de llaves no puede ser igual a la de otro registro del mismo tipo.
- ◆ Una variable definida como llave no puede ser modificada en el archivo.
- ◆ Las variables de un grupo de llaves deben pertenecer al mismo tipo de registro.
- ◆ Un registro (Maestro o un recurrente) puede tener varios grupos de llaves.
- ◆ La posición en la que se define una variable llave de un grupo de llaves sí influye en las formas de búsqueda tal como se señaló en la sección 1.

Para efecto de campos de AREA se deberá especificar R0 ó R1 y S.

#### 7.6.4) Tipo de campo en memoria y long/dec.

El **tipo de campo en memoria** se refiere a la forma en la que se manejan las variables dentro del programa, es decir, en forma alfanumérica o numérica. Las claves posibles de este campo son:

N1 N2 N3 N4 X F S T I A

**Los campos N** (N1, N2, N3 y N4), son campos numéricos y tienen las capacidades que se indican a continuación: (Se diagnostica si se sobrepasa).

N1	número máximo	+/-9999	(4 dígitos)
N2	número máximo	+/-99999999	(8 dígitos)
N3	número máximo	+/-999999999999	(12 dígitos)
N4	número máximo	+/-9999999999999999	(16 dígitos)

La capacidad anterior es solamente si se tiene cero decimales; por cada decimal hay que quitar un dígito. Así, por ejemplo, si un campo N2 tiene 3 decimales su capacidad es + / - 99999.999.

**Los campos alfanuméricos**, es decir, aquellos que pueden tener letras,

números o caracteres especiales, se indican con una X.

**Los campos F** sirven para almacenar fechas. El sistema internamente los trabaja como números de la siguiente forma:

XXXXYYZZ

En donde XXXX es el año comenzado en 0 para el año 0. YY es mes y ZZ es el día. Las variables fecha ocupan el mismo espacio que las N2. Para propósito de leer una fecha ésta se dará en la forma de día/mes/año.

**Los campos S** sirven para almacenar información alfanumérica "SI" o "NO". Las variables S ocupan el mismo espacio que una X de 2.

**Los campos T** sirven para almacenar información alfanumérica que se desplegará o imprimirá en más de un renglón. En ejecución se especifican el número de columnas por lo que las palabras que lo forman se acomodan para este efecto. El consumo de CPU y memoria en un programa de los campos texto es mucho mayor que los tipo X.

**Los campos I** sirven para guardar imágenes. Su uso esta restringido a usar el Emulador Sisinf en todos los sistemas operativos donde esta implantado SISINF.

**Los campos A** sirven para guardar audio. Actualmente NO se pueden usar.

La **Longitud o número de decimales** se deberá especificar cuando el campo sea X o N1, N2, N3 y N4, respectivamente.

En el primer caso significa el número de letras, números o caracteres especiales que puede contener, teniendo como máximo 120.

En el segundo caso se deberá especificar el número de decimales, los cuales pueden estar en los siguientes rangos:

N1 de 0 a 2

N2 de 0 a 4

N3 de 0 a 8

N4 de 0 a 8

### 7.6.5) Niveles de seguridad.

El **nivel de seguridad** en escritura y el nivel de seguridad en lectura sirven para mantener la **confidencialidad** de la información, así como la responsabilidad de las modificaciones.

Ambos niveles son números, de 0 a 10 cada uno e indica el nivel de usuario que se permite para modificar (Insertar, Re-escribir, Borrar) o leer la información. En ambos números, el 0 es el nivel más alto, es decir, el dato de mayor confidencialidad.

Ambos son opcionales (se asume 10 si no se especifican) y sólo es necesario indicarlos cuando se quiere un valor diferente en cuyo caso se deben especificar ambos.

Cuando un usuario de **SISINF** ejecuta un programa con ELS el sistema le pide su clave de acceso; dicha clave de acceso tiene asignados un número en lectura y otro en escritura, el cual se comparará con los más críticos del programa.

Para aclarar más, suponga que un usuario tiene como nivel de seguridad en escritura 5 y como nivel de seguridad de lectura 2, entonces un programa en el Lenguaje **SISINF** sólo podrá ser ejecutado por este usuario si se insertan, re-escriben o borran campos de nivel de seguridad en el rango de 5 a 10 y se leen campos con nivel de seguridad en el rango de 2 a 10. Si este usuario trata de ejecutar un programa en el Lenguaje **SISINF** que tenga un campo con nivel de seguridad en escritura 4, el programa ELS dará diagnóstico y no se podrá ejecutar.

### 7.6.6) Máscara de edición.

La máscara de edición es una clave que se pone en el caso de números (N1, N2, N3, N4) o fechas y sirve para definir la forma como se debe escribir (en la terminal o en la impresora) el valor del campo, la siguiente es una lista de claves, así como las máscaras que usa el sistema para hacer la impresión.

M1	\$\$,\$\$\$,\$\$\$.\$99
M2	\$\$,\$\$\$,\$\$\$.\$99-
M3	ZZ,ZZZ,ZZZ.99
M4	ZZ,ZZZ,ZZZ.99-
M5	ZZ,ZZZ,ZZZ.ZZ

M6	ZZ,ZZZ,ZZZ.ZZ-
M7	ZZZZZZZZZZZZZZ
M8	ZZZZZZZZZZZZZZ-
M9	99999999999999
M10	99999999999999-
M11	Z,ZZZ,ZZ9.99_9
M12	Z,ZZZ,ZZ9.99_9-
M13	****(UN MIL . . . . PESOS 00/100)****
M14	****(UN MIL . . . . DOLARES 00/100)****
M15	UN MIL . . . .
M16	XX/XX/XX (mes con número)
M17	XX/XXX/XX (mes con 3 letras)
M18	XX de XXXXX de XXXX (mes con nombre completo en mayúsculas)
M19	XX/XX/XXXX
M20	XX/XXX/XXXX
M21	XX de XXX de XXX (mes con nombre completo en minúsculas)

Los siguientes comentarios y restricciones son aplicables a las máscaras M1 a M18:

- ◆ Para N1, N2, N3, N4 se usan las máscaras M1 a M15, para fechas las M16 a M21.
- ◆ En el caso de M11 y M12 el número de decimales de la máscara es lo especificado para el campo como número de decimales.
- ◆ Si no se especifica máscara se asume M12 para números y M16 para fechas.
- ◆ Por facilidad de programación el número de columnas que ocupa al editarse un valor numérico es de 8 para N1, 12 para N2, 18 para N3 y 24 para N4, independientemente de que la máscara sea M1 a M12. Lo anterior sirve para que el programador se aprenda estas 4 constantes y a la hora de hacer reportes puede centrar sus campos numéricos con facilidad. Si se usa la opción /LONG. en impresión se podrá imprimir con menos dígitos. Si se pierde información se diagnostica.
- ◆ En el caso de máscaras M13 a M15 la longitud del campo editado es de 120 caracteres, con la instrucción de FORMAR TEXTO y su opción de

/CONTINUACION se puede imprimir o desplegar un número con esta máscara en varios renglones, si se usa la opción /LONG.. con estas máscaras y el número no cabe, se corta en la palabra MILLONES o en MILES,... Con la instrucción de SACAR PARAMETROS se puede obtener la parte del número que no se pudo imprimir. Es importante señalar que solo se imprimen números de hasta 999,999,999.99; si el número es mayor se trunca.

- ◆ En el caso de las fechas las longitudes de los campos editados son de 8 para M16, 10 para M17 y 24 para M18. Si se usa la opción /LONG.. se trunca y no se diagnostica.
- ◆ Las máscaras M19 y M16, M20 y M17 son iguales pero el año es con 4 dígitos en unas y 2 en otras.
- ◆ En el caso de las variables X no se puede especificar la máscara de edición y la longitud en impresión siempre será la longitud del campo. Si se usa la opción /LONG.. en impresión se podrá imprimir el campo con menos caracteres, es decir, truncado y no se diagnostica.
- ◆ En el caso de variables S no se puede especificar la máscara de edición y la longitud siempre será de 2.
- ◆ Los campos T, I o A no pueden tener máscara de edición.

Adicionalmente a las máscaras M1 a M12 se puede especificar la máscara MX que es una extensión de éstas y que permite definir la longitud total deseada, así como los decimales.

MX '\$Z,ZZZ,ZZ9.9999-B' o MX '\$9,999,999.9999-B'

Los siguientes comentarios son aplicables:

- ◆ Se debe especificar las letras MX y luego entre apóstrofes cualquiera de los dos formatos indicados.
- ◆ El signo de pesos (\$), la coma (,) y el signo de menos (-) son opcionales, si no se especifican no se incluyen en el número editado.
- ◆ Si se especifica la “,” es necesario ponerla cada 3 posiciones en la máscara.
- ◆ La letra B al final indica que si el valor del campo es cero no se imprimirá información, es decir, se deja en blanco.

- ◆ La cantidad de Z o 9 que se especifique servirá para indicar la longitud total del campo editado, así como el número de decimales que tendrá.
- ◆ El número de decimales especificado puede ser diferente a los que tiene el número. Si son menores los decimales faltantes se omitirán, si es mayor serán ceros.
- ◆ Si no hay decimales el “.” se deberá omitir.
- ◆ El número de decimales que se especifican debe estar entre 0 y 8.
- ◆ La parte entera del número no deberá exceder de 16 dígitos.
- ◆ Si el número de dígitos en la parte entera excede a lo que tiene el número de acuerdo a su tipo se asumirán ceros. Así por ejemplo es válido especificar 99999 para un N1. El primero siempre será 0.
- ◆ El signo “-“ y la letra B al final se pueden especificar en cualquier orden.
- ◆ La longitud que se defina en MX será la utilizada para la instrucción IMPRIMIR, el efecto sería como si se pusiera la opción de /LONG.. a la variable.
- ◆ La longitud de lectura en las instrucciones de Terminal se calcula usando esta máscara cuando no se tiene rango.
- ◆ En las instrucciones de impresión se deberá tomar en cuenta que por el uso de esta máscara las longitudes de los campos editados no son 8, 12, 18, 24 como en las máscaras M1 a M12, sino la longitud calculada de acuerdo a lo especificado.

Los siguientes son ejemplos del uso correcto o incorrecto de esta máscara. Se incluye la longitud total del campo editado.

ZZZ9B	4
99	2
99.9	4
ZZZZ	INCORRECTO SE DEBE USAR ZZZ9B
Z99.9	INCORRECTO SI HAY “Z” DEBE HABER SOLO UN NUEVE ANTES DE “.”
Z,ZZZZZZ	INCORRECTO SI HAY UNA COMA DEBE SER DE 3 EN 3

Z,ZZZ,ZZZ.	INCORRECTO SI NO HAY DECIMALES NO SE PONE “.”
Z,ZZ9.99-	9
\$ZZ9.99B	7

En el caso de las variables X no se puede especificar la máscara de edición y la longitud en impresión siempre será la longitud del campo. Si se usa la opción /LONG. en impresión se podrá imprimir el campo con menos caracteres, es decir, truncado y no se diagnostica.

Si se usa el Emulador Sisinf es mejor usar la opción /LONG-MAX. o /LM ya que al usar fonts proporcionales (www ocupa mas espacio que iii) se ajusta a la información que cabe en las columnas especificadas. El número de caracteres impresos puede ser más del valor puesto en la opción, en algunos casos como información en mayúsculas puede ser menos.

### 7.6.7) Dígito Verificador.

Si el usuario desea la validación por dígito verificado en algunos de los campos numéricos o alfanuméricos, deberá especificar cualquiera de las claves DV. En el caso de leer el campo se valida el dígito verificador si no está correcto se pide se vuelva a dar la información.

La clave DV usa la forma de cálculo estándar de **SISINF**.

### 7.6.8) Rango.

Si se desea que en un campo numérico se haga **una validación por rango** se deberá especificar la palabra RANGO o la letra R y a continuación dos números que especifican el límite inferior y superior para este campo. Ambos números se deberán de especificar con un número de decimales igual a lo que aparece en el parámetro de número de decimales.

Si se desea que en un campo fecha se haga **una validación por rango** se deberá especificar RANGO o R y a continuación la fecha inferior como 3 números, día, mes y año y enseguida la fecha superior en el mismo formato.

El año se puede especificar en 2 o 4 dígitos. Si se especifica como 2 dígitos los años de 0 a 20 se asumen del 2000 a 2020 y los de 21 a 99 como 1921 a 1999.

En la ejecución de un programa de **SISINF** en el que se lea un campo por

terminal se revisa que esté en el rango, si no está, se marca error y se pide que se vuelva a dar la información.

### 7.6.9) Número de Seguridad.

Si se desea tener un campo con un número de seguridad se deberá especificar la clave SEG. Y a continuación un número entre 1 y 9999.

Esta protección sirve para que una sola persona a través de una clave de acceso pueda escribir o modificar un campo.

Cuando un usuario pide la ejecución de un programa a través de ELS, el sistema pide su clave de acceso; con esto se conocen las opciones SEG. A las que tiene acceso. Por ejemplo, suponga que un usuario tiene SEG. 10 y 150, entonces podrá ejecutar programas de **SISINF** cuyos campos **NO** tenga opción SEG., o bien tengan la 10, la 150 o ambas. Si un programa tiene opción SEG. Con un valor de 13, dicho usuario no podrá ejecutar este programa.

### 7.6.10) Valor.

Si se desea que un campo numérico, alfanumérico o fecha tenga un **valor** de acuerdo a una tabla, se deberá de poner la palabra VALOR o la letra V y a continuación el nombre de la Tabla. En la ejecución de un programa de **SISINF** en el que se lea un campo con valor se revisa que sea igual a la primer columna de algún renglón de la tabla, si no lo es se diagnostica y se pide que se vuelva a dar la información.

### 7.6.11) Grupo.

Para especificar el grupo a que pertenece una variable para el módulo CAS es necesario poner la palabra GRUPO o la letra G y a continuación el número del mismo.

El número del grupo se debió definir con la definición de GRUPO.

### 7.6.12) Area.

La opción de AREA se usa sólo en los campos Texto, en los campos Imagen y en las campos de Audio, en todos los casos se podrá especificar el **área máxima** que se usará como un número que indica los múltiplos de 512

caracteres y luego poner el **área promedio** que se usará para que con ello se calcule el área total en disco que usará el archivo. Si esta se omite se asume igual a la máxima.

El formato general es:

**AREA**      Máximo Promedio

Los campos imagen se almacenan del mismo tamaño que ocupa el archivo en el cual esta grabada la imagen digitalizada. Se sugiere revisar el tamaño de los archivos de varias imagenes que se deseen guardar y con esto poner el AREA. Si no se especifica se asume AREA 4000 100

En el caso de los campos Texto el área máxima se usa para separar áreas de trabajo en los programas que usen el campo, si se excede se dará error en ejecución. El área promedio se usará para el cálculo del área en disco y es fácil hacer un cálculo conociendo los textos que se deseen almacenar.

### **7.6.13) Opciones anteriores de Imagen.**

Las opciones de imagenes como COLOR, BN, MEDIDA, ... NO se usan en las imágenes que maneja el Emulador SISINF.

Si se tiene una DBD con estas opciones y los archivos que las tienen ya están creados, se deben de dejar tal como están.

En una DBD nueva o cuando se van a crear los archivos, se deben omitir.

### **7.6.14) Opciones de Imagen.**

Se tienen 4 parámetros que se pueden poner al definir imágenes en DBD, los cuales son:

ESC-AGRANDAR  
ESC-NO-AGRANDAR  
ESC-REDUCIR  
ESC-NO-REDUCIR

Cuando una imagen se lee de un archivo, ésta tiene ciertas dimensiones en pixeles dependiendo del dispositivo con que se capturó. Luego SISINF permite que se disminuya el área de interés con lo cual sus medidas se pueden reducir. De esta forma se guarda en disco.

Cuando una imagen se va a desplegar o a imprimir, el recuadro para hacerlo tiene unas ciertas dimensiones en pixeles. SISINF tiene que hacer un proceso de ajuste que está bajo control del programador por esta diferencia de tamaños.

Si se permite AGRANDAR y REDUCIR, la imagen se cambia para llenar el recuadro. Si NO se permite agrandar la imagen y ésto era necesario, entonces se centra dentro del recuadro. Si no se permite reducir la imagen y era necesario, ésta se trunca en el recuadro.

En general las imágenes como fotos de personas, paisajes,... se puede pedir que se puedan agrandar y reducir. En el caso de íconos o gráficas, pueden ser distorsionados en este proceso de agrandar o reducir, por lo que es mejor cancelarlos o hacer pruebas de cómo se ven.

En el caso de logos si estos se generan con medidas grandes (Cuando se lee la imagen se da el dato del tamaño en pixeles) no se verán afectados en un proceso de reducción, si tienen medidas pequeñas y contienen letras, es posible que se distorsionen en el proceso de agrandar.

Cabe mencionar que las gráficas creadas con SISINF se generan en un tamaño igual al del área en donde se van a desplegar o imprimir, razón por la cual NO se distorsionan.

Para variables de AREA se asume ESC-NO-AGRANDAR y ESC-NO-REDUCIR

Para variables de ARCHIVO se asume ESC-AGRANDAR Y ESC-REDUCIR

### **7.6.15) DIP y DII.**

Las opciones de DIP (Definir Identación de Párrafo) y DII (Definir Identación de Inciso) sirve para campos texto y definirán estas identaciones cuando se forma en ejecución las líneas de texto. El formato general es:

DIP	rb	cb		
DII	rb	cb1	cb2	

Los siguientes comentarios son aplicables:

- ◆ Se deberá especificar DIP para identación de párrafo.
- ◆ Se deberá especificar DII para identación de inciso.
- ◆ El número de renglones en blanco entre el fin del párrafo o inciso anterior

y el inicio de éste párrafo o inciso se especifica en 'rb'.

- ◆ En indentación de párrafo el número de columnas en blanco que tendrá el primer renglón del párrafo es 'cb', los demás se asume que serán 0.
- ◆ En indentación de inciso el número de columnas en blanco que tendrá el primer renglón del inciso será 'cb1', los demás serán 'cb2'.
- ◆ En ambas opciones es necesario especificar todos los valores.
- ◆ Tanto DIP como DII son opcionales si se omite se asume los siguientes valores. En DIP se asume rb=2 y cb=3. En DII rb=1, cb1=3 y cb2=4.
- ◆ Cuando un texto se forma con la instrucción FORMAR TEXTO se puede especificar si una redacción es párrafo o es inciso y cuándo termina éste.
- ◆ Cuando se lee un texto también se podrá especificar si una redacción es párrafo o es inciso y cuándo termina éste.

#### **7.6.16) Opción CAS.**

La opción de CAS sirve para especificar a cada campo los cálculos que se pueden efectuar. Lo anterior sirve para que los ejemplos que despliega el CAS ayuden al usuario final a ver otras posibilidades. Si ésta opción NO se especifica, el CAS no tendrá ejemplos tan completos.

La forma de poner ésta opción es con la palabra CAS y luego una o varias de las siguientes palabras para indicar el tipo de cálculo que se puede efectuar con el campo. Los tipos de cálculo son: ACUMULAR, PROMEDIO, PORCIENTO, OPERACIONES, RANGO Y SELECCION.

#### **7.6.17) Mayúsculas.**

En campos tipo X se puede especificar que cuando se lee información en la terminal, si teclean minúsculas éstas sean convertidas a mayúsculas, al poner la palabra **MAYUSCULAS** o la letra **M**. esta opción sólo es válida cuando se usa OPCION TERMINAL AREA.

#### **7.7) Grupo.**

La declaración de grupo tiene el siguiente formato:

**GRUPO** NG 'Nombre'

La palabra GRUPO indica que se trata de esa declaración, luego se pone el número del grupo cuyo nombre se va a especificar. Este número debe estar entre 1 y el total de grupos que tenga el archivo. Por último se especifica entre apóstrofes el nombre del grupo de 1 a 40 caracteres.

**7.8) Definición de IGUAL.**

La definición de IGUAL sirve para indicar que un campo es igual a otro campo. El formato de la definición es:

IGUAL	Nombre del archivo
	Nombre del campo
	Recurrente a que pertenece
	Tipo de campo en disco.

En donde la palabra IGUAL debe aparecer tal y como se señala en el formato, y los campos posteriores se refieren a información para hacer la igualdad tomando en cuenta las siguientes reglas:

- ◆ El archivo a que se refiere el parámetro de nombre de archivo debe estar especificado por la definición de ARCHIVO en DBDSxx.
- ◆ Esta definición es la única que permite tener nombres de campos iguales.
- ◆ El recurrente a que pertenece es R0 para el maestro, R1 para el recurrente 1,...
- ◆ El parámetro tipo de campo en disco es igual que en la definición de campo, es decir, S, LL1, LL2, ...
- ◆ No se permite igualar variables dentro del mismo archivo a menos que ambas sean llaves perteneciendo al mismo registro y a diferentes grupos de llaves. Dicho de otra forma, una llave puede estar en varios grupos de llave del mismo registro maestro o recurrente.

## 7.9) Definición de TABLA.

La definición TABLA sirve para indicar las características de una tabla, así como especificar que la información que sigue a continuación, son los renglones de la tabla uno por registro, el formato de la definición es:

TABLA	Nombre de la TABLA
	Número de columnas
	Tipo de campo en memoria y decimales o longitud de cada columna.
	Nombre extendido de la TABLA para documentación o CAS. (Opcional)
	Columna para CAS. (Opcional)

En donde la palabra TABLA debe aparecer tal y como se señala en el formato y los campos posteriores se refieren a la información de características de la TABLA, tomando en cuenta las siguientes reglas:

- ◆ El nombre de la TABLA debe tener 6 caracteres como máximo y debe ser alfanumérica, es decir, no debe comenzar con punto, signo +, signo – o número. Tampoco comenzar con \$ ; . & /.
- ◆ El número máximo de columnas es 10.
- ◆ El tipo de campo en memoria se refiere a especificar si es N1, N2, N3, N4, X o F y a continuación sus decimales o longitud. Lo anterior se hace para cada columna de acuerdo a las reglas vistas en la definición de campo.
- ◆ El nombre extendido se usa por el módulo de documentación y por el módulo CAS y sirve para tener una idea más clara del contenido de la TABLA. Se debe especificar entre apóstrofes y de 1 a 40 caracteres.

La columna para CAS se usa por éste módulo como el número de la columna que servirá para convertir información numérica o de claves de la primer columna a una que sea una explicación más completa. Tal sería el caso de una tabla de estados en que la primer columna es el número y la segunda el nombre. El dato aquí sería 2.

A continuación de la definición de TABLA se deberán especificar los renglones de la misma, uno por registro. Se deberán observar las siguientes reglas:

- ◆ Al inicio de cada registro se deberá poner una T.

- ◆ En campos numéricos el número de decimales que se pongan deberá ser igual a lo que se especificó en la definición de TABLA para esa columna.
- ◆ Los campos alfanuméricos se deberán especificar entre apóstrofes y con una longitud igual o menor a la especificada en la definición de TABLA para esa columna.
- ◆ Los campos fecha se deberán especificar como tres números, el primero de ellos el día, luego el mes y al final el año.

### 7.10) \*D.

Se propone que la descripción general de cada archivo, recurrente o campo se haga mediante el uso de registros de comentarios que se inicien con \*D. También se recomienda usar para las TABLAS.

El módulo de documentación **SISINF** saca del archivo DBDXxx ésta información y junto con otra información de los programas forma el reporte de documentación de programación.

## 8) EJECUCION DE DBD.

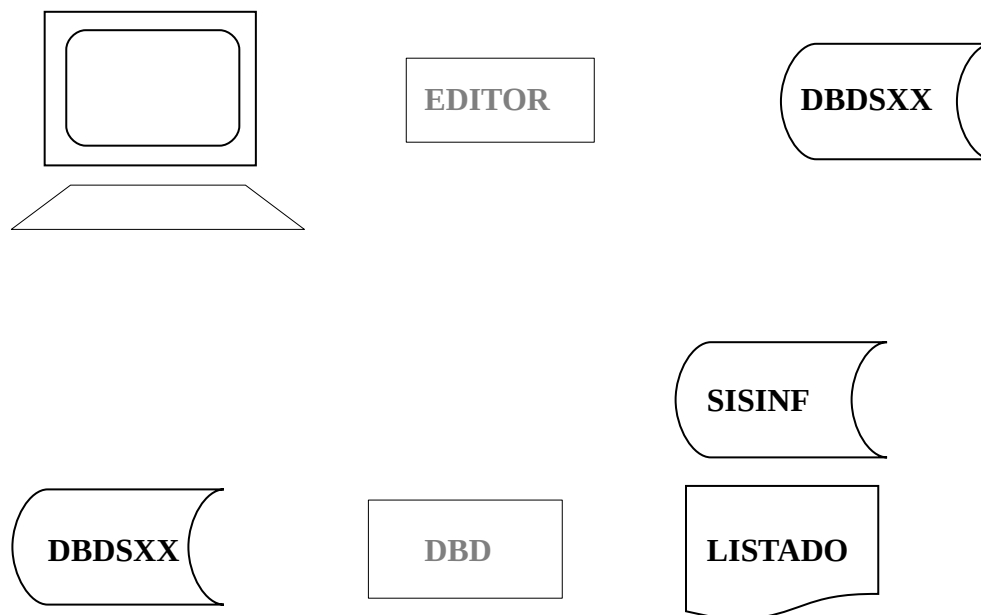
### 8.1) Pasos para la ejecución.

A continuación se describen los pasos para ejecutar el módulo DBD:

- ◆ CREAR por instrucciones del sistema operativo el archivo DBDSxx en donde xx son las iniciales de la aplicación.
- ◆ Con el EDITOR que se provee con la computadora, teclear las definiciones en el archivo DBDSxx.
- ◆ Ejecutar el módulo DBD, con lo cual se revisa las definiciones y se pasan a otros archivos en un formato que facilite su uso por los demás módulos de **SISINF**.

Si hay errores se deberán corregir a través del EDITOR y volver a ejecutar el módulo DBD hasta que no se tenga errores.

Gráficamente la ejecución queda así:



## 8.2) Datos para la ejecución.

La ejecución del módulo DBD es a través de la instrucción del sistema operativo en la que se pide la ejecución de un programa (ordinariamente RU DBD o DBD). Al hacerlo, la pantalla aparecerá:

SISINF	DBD	VX.X	MAQUINA	EMPRESA
	CLAVE DE ACCESO		XXXXXXXXXX	
	INICIALES DE DBD		XX	
	DISPOSITIVO DE DBD		XXXXXXXXXXXXXXXXXX	

**CLAVE DE ACCESO.**- El usuario deberá teclear un blanco si es que no se tiene el sistema de claves, o la clave de acceso a los diferentes módulos de **SISINF** que le fue asignada por el departamento de sistemas.

**INICIALES DE DBD.**- Deberá contestar con las iniciales "XX" de su archivo DBDSxx.

**DISPOSITIVO.**- (En algunas máquinas se llama DIRECTORIO, en otras GRUPO Y CUENTA, etc.) en este renglón deberá contestar con el dispositivo en donde se creó el archivo DBDSxx, si sólo oprime RETURN, se le asigna un valor de acuerdo a cada modelo y marca de computadora.

Una vez hecho lo anterior, aparecerá la hora de inicio de los 4 programas que integran el módulo DBD, el último borra la pantalla y ésta aparecerá como:

```

SISINF      DBD          VX.X      XXXX      YYYY
INICIALES DBD          XX
NUMERO DE ERRORES          XXXX
UNIDAD DE SALIDA: (?/T/I) X
SE LISTA: (?/D/E/V/T/A/C) X

PARA CANCELAR EL DESPLEGADO POR TERMINAL DAR \ o /
ESTA COMPILACION NO SE REGISTRA EN DOCUMENTACION
SISINF POR ESTAR INACTIVA

TIEMPO      XXM  XXS

```

El usuario debe revisar el NUMERO DE ERRORES que se tuvo y con esto decidir las siguientes acciones a tomar.

Deberá contestar la UNIDAD DE SALIDA donde se desea ver los listados, una **T** para terminal o una **I** para impresora. Si se da ¿ se dará ayuda.

Las siguientes son las opciones con las que se pueden contestar a SE LISTA:

- ◆ No dar ninguna letra en cuyo caso se asume la D.
- ◆ Poner una **D** en cuyo caso se listan la DBD y los errores si los hubo.
- ◆ Poner una **E** en cuyo se listan solo los errores si los hubo.
- ◆ Poner una **V** para listar las variables ordenadas.
- ◆ Poner una **T** para listar tablas, archivos usados y áreas.

- ◆ Poner una **A** para listar el consumo de área en disco.
- ◆ Poner varias de las letras anteriores para listar lo indicado con cada letra.
- ◆ Poner una **C** para que listen las instrucciones al compilador y las líneas que se quitan (Ver Capítulo 9).

Para que el programador no tenga que recordar estas opciones, al teclear ? se dará ayuda.

Si la información se despliega por la terminal aparecerá el letrero.

```
PARA CANCELAR EL DESPLEGADO POR TERMINAL DAR \ o /
```

Para recordar al programador que cuando se despliega por la terminal, al llenarse la pantalla se puede teclear RETURN para continuar o bien dar diagonal o diagonal invertida para cancelar.

Si se tiene la declaración de DOCUMENTACION pero ésta no esta activada por el módulo CPS se desplegará:

```
ESTA COMPILACION NO SE REGISTRO EN DOCUMENTACION  
SISINF POR ESTAR INACTIVA
```

El programador deberá avisar al departamento de Sistemas para que la active o bien estar concientes que NO se está registrando.

Al final aparecerá el tiempo usado en la compilación.

### 8.3) Errores.

Los errores son desplegados en el siguiente formato:

Secuencia Texto de línea con error.

```
**** CAMPO NN campo SECUENCIA ALTERNA NNNN
```

Descripción del error.

Donde:

Secuencia es el número de la línea.

NN es el número del campo dentro de la línea.

Campo es la nombre del campo con error.

NNNNes la secuencia de otra línea relacionada con el error.

Por ejemplo:

```
0002  NUM-CLT*  *  R0  XL  N1  0
****  CAMPO    05  XL
EN LA INSTRUCCIÓN DE "CAMPO" EL TIPO DE CAMPO
EN DISCO NO ES S, LL1, LL2 ...
0008  NUM-CLT*  *  R0  S   X   1A
****  CAMPO 01  NUM-CLT SECUENCIA ALTERNA 0002
NOMBRE DE CAMPO DUPLICADO.
****  CAMPO 07  1A
SE ENCONTRO LETRA O CARACTER ESPECIAL EN NUMERO
```

Para el **número del campo** se deberá contar de izquierda a derecha en la línea con error. Cada palabra, número o letrero entre apóstrofes se considera un campo.

Cabe aclarar que en algunos diagnósticos este **número del campo** es aproximado y también que en algunos diagnósticos el **campo con error** aparecerá **truncado**.

La opción E es muy útil para quitar errores en **sesiones** de terminal en las que se compila y se edita, se vuelve a compilar ... hasta que no hay errores. Al final aparecerá el tiempo total usado en la ejecución de DBD, desde el inicio hasta este letrero.

El módulo DBD despliega los diagnósticos de errores que se tengan en el archivo DBDSxx así como en los archivos que genera éste módulo en el formato normal de **SISINF** esto es:

```
OP =   XX  ERROR EN   YYYYYY  (ZZZ ...)
```

Al efectuarse la operación 'xx' en el archivo 'yyyyyy' ocurrió el error zzzz. Se deberá consultar el manual de l sistema de archivos para revisar el tipo de error, que puede ser falta de área en disco, muchos archivos abiertos, etc. si no es un error normal se deberá consultar a personal de su Distribuidor.

## 8.4 Opción “A” (Archivos) en módulo DBD.

El diseñador de la DBD, al compilar la DBD y seleccionar la opción “A” puede realizar lo siguiente:

- ◆ Conocer si tiene el área necesaria para que los archivos de su aplicación se puedan alojar en los discos de su instalación.
- ◆ Planear las necesidades de área en disco de una aplicación.
- ◆ Examinar lo que consume cada recurrente pudiendo eliminar variables o recurrentes cuyo consumo de área no corresponda al beneficio de tener almacenada la información.
- ◆ Examinar lo que consume cada grupo de llaves para balancear el consumo de área en disco con la necesidad de usar más tiempo de computadora al tener que usar programas para ordenar la información, o bien programas que busquen la información en forma secuencial.
- ◆ En el caso de campos Imagen revisar el consumo y reducirlo mediante AREA.
- ◆ Revisar el consumo en campos Texto, Imagen o Audio.

La forma en la que se reporta es:

ARCHIVO	NIVEL	LONGITUD (P)	REG. DE xxxx(P)	P = yy	BITS
aaaaa					
LLbb		cc	dddd	eeee	
Rff	gg	hhhh	iiii		
POR TEXTO/IMAGEN/AUDIO jjjj					
TOTAL REG		kkkk			

En donde:

**aaaa** Es el nombre del archivo.

**LLbb** Indica la información del grupo de llaves con número ‘bb’ en este

	renglón.
<b>Bb</b>	Número del grupo de llaves.
<b>cc</b>	Nivel dentro del archivo del grupo de llaves.
<b>dddd</b>	Número de palabras que ocupa una llave.
<b>eeee</b>	Número de registros que ocupa el grupo de llaves en su directorio.
<b>Rff</b>	Indica información del recurrente con número 'ff' en este renglón.
<b>ff</b>	Número del recurrente.
<b>gg</b>	Nivel dentro del archivo del recurrente.
hhhh	Número de palabras que ocupa un recurrente.
iiii	Número de registros que ocupan todos los recurrentes.
jjjj	Número de registros que se ocupan por los campos texto, imagen o audio del archivo.

Si se asume el número de registros de cada grupo de llaves, más el número de registros de cada recurrente, más una cantidad que varía entre 3 y 32 (para otros usos dentro de **SISINF**), se obtendrá el total de registros usados por el archivo, el cual se reporta en 'kkkk'.

En el caso de recurrentes opción M el número de registros usados está también incluido en el recurrente del que dependen.

Dado que en cada marca y modelo diferente de computadora la palabra tiene un cierto número de bits y el registro tiene un cierto número de palabras, en el encabezado se especifica:

REG DE xxx (P)	P = yy BITS
----------------	-------------

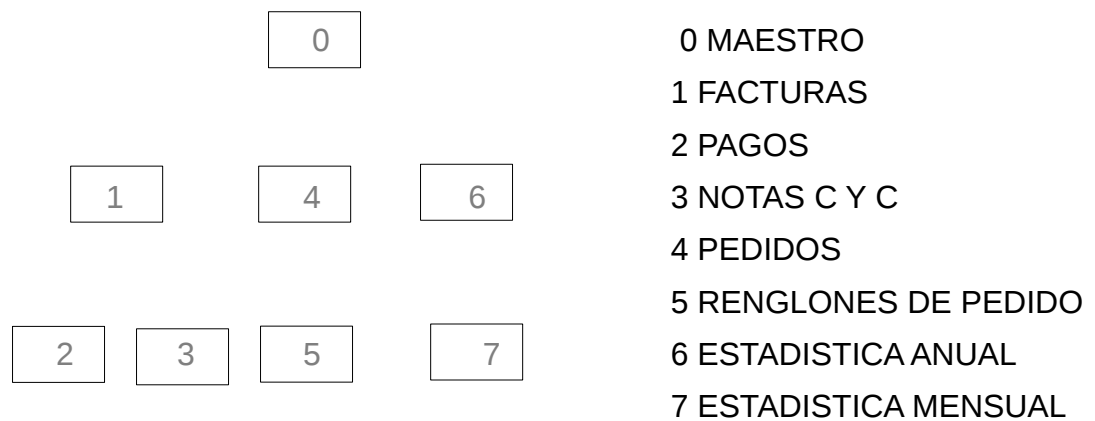
En donde 'xxx' es el número de palabras por registro y 'yy' es el número de bits por palabra.

Con ésta información, el usuario, podrá saber si se cuenta con el área de disco suficiente.

## 9) EJEMPLO.

### 9.1) Ejemplos de DBD de Clientes.

Para terminar con las definiciones de DBD, se ilustra el ejemplo de la sección 5 con algunos campos de un archivo de clientes. Para ello se grafica la estructura del archivo.



```

ARCHIVO CLIENT 7 4 ' ' 'CLIENTES' 4
RECORRENTE 0 1000 M 3 1 4 6 'CLIENTES'
RECORRENTE 1 5 P 2 2 3 'FACTURA'
RECORRENTE 2 3 P 0 'PAGO'
RECORRENTE 3 5 P 0 'NOTA DE CARGO O CREDITO'
RECORRENTE 4 2 P 1 5 'PEDIDO'
RECORRENTE 5 20 P 0 'RENGLON DE PEDIDO'
RECORRENTE 6 1 P 1 7 'ESTADISTICA ANUAL'
RECORRENTE 7 12 M 0 'ESTADISTICA MENSUAL'
    
```

\*

```

GRUPO 1 'DATOS GENERALES'
GRUPO 2 'FACTURAS'
GRUPO 3 'PEDIDOS'
GRUPO 4 'ESTADISTICAS'
    
```

```

*
***  MAESTRO
*
NUM-CLI      'NUMERO DEL CLIENTE' '#CLIENTE' R0      $
              LL1 N2 0 5 10 DV R 1 1000 SEG. 10 G 1
NOM-CLI      'NOMBRE DEL CLIENTE' * R0 S X 30 G 1
DIR-CLI      'DIRECCION DEL CLIENTE' * R0 S X 20 5 7 G 1
EST-CLI      'ESTADO DEL CLIENTE' '#EST' R0 LL2 N1 $
              0 5 7 VALOR ESTADO G 1
ZONA         * * R0 LL2 N1 0 5 7 R 1 10 G 1
IGUAL        CLIENT NUM-CLI R0 LL2 G 1
LIM-CR       'LIMITE DE CREDITO' * R0 S N2 2 5 5 G 1
*
***  FACTURAS
*
NUM-FACT     '#FACTURA' * R1 LL3 N1 0 5 8 R $
              1000 9999 MX 'ZZZZB' G 2
FECHA-FACT   'FECHA FACT' * R1 S F 5 8 R 1 1 00 $
              31 12 03 M17 G 2 CAS SELECCIÓN
IMP-FACT     'IMPORTE' * R1 S N2 2 5 8 M1      $
              R 1.00 999999.99 CAS ACUMULAR PORCIENTOS$
              PROMEDIO OPERACIONES G 2
*
***  PAGOS
*
REF-PAGOS    'REFERENCIA' 'REF' R2 S N1 0 5 7 G 2
FECHA-PAGO   'FECHA DE PAGO' * R2 S F 5 7$
              R 1 1 03 31 12 03 M17 GRUPO 2
IMP-PAGO     'IMPORTE' * R2 S N2 2 5 7 M2 $
              R 1.00 999999.99 GRUPO 2 CAS ACUMULAR
*
***  NOTAS DE CARGO Y CREDITO

```

```

*
CLAVE-CC      * * R3 S X 4 5 7 VALOR T-CL GRUPO 2
DESC-CC      'DESCRIPCION CARGO O CREDITO' * R3 S $
              X 40 5 7 GRUPO 2
IMP-CC       'IMPORTE' * R3 S N2 2 5 7 M2 R $
              1.00 999999.99 GRUPO 2 CAS ACUMULAR
*
***
*
NUM-PED      'PEDIDO NUMERO' '#PED' R4 LL4 N1 0 $
              6 8 R 1 9999 MX 'ZZZ9' GRUPO 3
FECHA-PED    'FECHA DEL PEDIDO' * R4 S F 6 8 M18 GRUPO 3
TRAN-PED     'TRANSPORTE' * R4 S X 40 6 8
EST-PED      'ESTADO' * R4 S N1 0 6 8 VALOR EST-P G 3
*
***
*
IGUAL        ART NUM-ART R5 S GRUPO3
CAN-AR       'CANTIDAD' * R5 S N1 0 6 8 GRUPO3
PRECIO-ART   'PRECIO' * R5 S N2 2 6 8 M2 GRUPO3
*
***
*
ESTADISTICAS ANUALES
*
TOTAL-$      'TOTAL-$' * R6 S N2 2 G 4
TOTAL-#      'TOTAL-#' * R6 S N2 2 G 4
*
***
*
MES          * * R7 S X 4 10 10 VALOR MESES GRUPO 4
TOTAL-$-M    * * R7 S N2 2 GRUPO 4 CAS $
              ACUMULAR PROMEDIO PORCIENTO

```

```

*
***  AREA DE TRABAJO
*
      AREA
      FECHA-DIA * * R0 S F M18
      HORA-DIA  * * R0 S X 8
      CANT-LET  * * R0 S N3 2 M13
*
***  TABLAS
*
TABLA ESTADO 2 N1 0 X 20 'ESTADO' 2
T 1 'NUEVO LEON'
T 2 'ESTADO DE MEXICO'
*
TABLA T-CL 3 X 2 X 20 N1 0 'CLAVES DE PAGO' 2
T CH 'CHEQUE DEVUELTO' 1
T PP 'PRONTO PAGO' 2
*
TABLA EST-P 3 N1 0 X 30 N1 0 'ESTADO DE PEDIDOS' 2
T 1 'PENDIENTE AUTORIZAR CREDITO' 2
T 2 'PENDIENTE ORDEN DE SURTIDO' 3
T 3 'PENDIENTE DE FACTURAR' 4
*
TABLA MESES 1 X 4 'MESES'
T ENE
T FEB
T MAR

```

Los siguientes comentarios son aplicables:

- ◆ Como se había comentado, se tienen 4 grupos de llave. Para hacer búsqueda PRINCIPAL o SECUENCIAL por número de cliente, por número de factura y por número de pedido. Y búsqueda NO-PRINCIPAL en el grupo de llaves 2. Es permitido igualar dentro del mismo archivo por llaves para formar otro grupo de llaves como en el caso del 2.

- ◆ Se han definido para muchas de las variables validación por rango de acuerdo al más adecuado.
- ◆ Los niveles de seguridad se han definido de acuerdo a la confidencialidad de la información. Al número del cliente se le ha puesto la opción SEG. 10 para que solamente unas claves de acceso puedan dar altas o bajas de un cliente.
- ◆ En la tabla T-CL (clave de cargo y crédito) se tiene una tercera columna para indicar si es cargo o es crédito. Al igual en la tabla EST-P (Estado del pedido) se tiene una tercera columna para indicar cuál es el estado al que pasa una vez satisfecho el estado actual. Con el uso de tablas se pueden poner varios indicadores que ayudan a hacer la programación más sencilla.
- ◆ Se han definido los nombres extendidos de recurrentes y tablas, así como la opción CAS para los campos más usuales. Lo anterior es para usar mejor el módulo CAS.

## 9.2) Ejemplo de Ayudas en DBD.

Las ayudas de los campos que se están leyendo en la ejecución de los programas, se pueden hacer de diferentes formas. Se propone una forma estándar que se puede modificar en cada instalación de acuerdo a las necesidades propias. En esta forma se tendrán tres archivos para que tengan los textos de las ayudas para cada variable, para cada área y para cada programa.

La parte de DBD de estos archivos es:

\*ISAYUD

\*

\*       DEFINICIONES PARA DEFINIR LOS ARCHIVOS DE AYUDAS QUE  
\*       SE PUEDE USAR POR LOS SISTEMAS DE LA INSTALACION.

\*

\*       **AYUDA POR VARIABLE (AYUDAV)**

\*       QUE CONTIENE LOS TEXTOS PARA LA AYUDA DE CADA CAMPO  
\*       DE LA DBD.

\*       LAS LLAVES SON NOMBRE Y PROGRAMA. SI ESTA ULTIMA ES  
\*       BLANCO SE ASUME QUE ES UNA AYUDA GENERAL DE LA

\* VARIABLE, ES DECIR SE PUEDE USAR EN TODOS LOS PROGRAMAS.

\*

\* **AYUDA POR AREA (AYUDAA).**

\* CONTIENE LOS TEXTOS PARA LA AYUDA DE CADA AREA DE  
\* CADA PROGRAMA Y POR LO TANTO ESTO ES LA LLAVE.

\*

\* **AYUDA POR PROGRAMA (AYUDAP).**

\* CONTIENE LOS TEXTOS PARA LA EXPLICACION DE QUE HACE  
\* CADA PROGRAMA PARA LAS AYUDAS EN LOS MENU, LA LLAVE  
\* ES EL PROGRAMA

\* EN CUALQUIER CASO SI EL REGISTRO NO EXISTE, ES POR QUE NO  
\* TIENE AYUDA

\*

\* AYUDA POR VARIABLE Y PROGRAMA

\*

ARCHIVO AYUDAV 0 1 ' ' 'AYUDA POR VARIABLE Y PROGRAMA' 0  
RECORRENTE 0 500 M 0 'AYUDA POR VARIABLE Y PROGRAMA' \*

\*

NOM-VAR 'NOMBRE DE VARIABLE' \* R0 LL1 X 30

INI-PRO 'PROGRAMA' \* R0 LL1 X 4

TEXTO-AYUDA 'TEXTO DE AYUDA' \* R0 S T AREA 12 3 DIP 0 3 \$  
DII 0 4 5

\*

\* AYUDA POR AREA Y PROGRAMA

\*

ARCHIVO AYUDAA 0 1 ' ' 'AYUDA POR AREA Y PROGRAMA' 0  
RECORRENTE 0 500 M 0 'AYUDA POR AREA Y PROGRAMA' \*

\*

NUM-AREA 'NUMERO DE AREA' 'AREA' R0 LL1 N1 0 MX 'ZZZ9'

IGUAL AYUDAV INI-PRO R0 LL1

IGUAL AYUDAV TEXTO-AYUDA R0 S

\*

```

*      AYUDA POR PROGRAMA
*
ARCHIVO AYUDAP 0 1 ' ' 'AYUDA POR PROGRAMA' 0
RECORRENTE 0 200 M 0 'AYUDA POR PROGRAMA' *
*
IGUAL AYUDAV INI-PRO R0 LL1
IGUAL AYUDAV TEXTO-AYUDA R0 S
*
AREA
REN-CE          * * R0 S N1 0
COL-CEL         * * R0 S N1 0
REN-AYUDA       * * R0 S N1 0
COL-AYUDA       * * R0 S N1 0
TOT-REN-DISP    * * R0 S N1 0
TOT-REN-AYUDA  * * R0 S N1 0
CAR-TEX         * * R0 S N1 0
AYUDA-TEX       * * R0 S N1 0
CONDICION-AYUDA * * R0 S N1 0

```

El AREA en los campos texto se calculó suponiendo que cada ayuda se desplegará en un máximo de 10 páginas, un promedio de 3 y de 30 columnas por 20 renglones.

El cálculo fue  $(10 \times 30 \times 20) / 512 = 12$  y

$$(3 \times 30 \times 20) / 512 = 3$$

También se cambió el número de renglones en blanco en la indentación de inciso y de párrafo a 0 para aprovechar mejor la terminal.

### 9.3) Ejemplo de DBD de Imágenes.

El siguiente ejemplo es de un sistema en el que se guardarán los datos generales de las personas, así como su foto y su firma. La DBD sería:

```
ARCHIVO PERS 0 2
```

```

RECORRENTE 0 100 M 0
NUM-PER 'NUMERO DE LA PERSONA' * R0 LL1 N1 0
INI-PER 'INICIALES DE LA PERSONA' * R0 LL2 X 4
IGUAL PERS NUM-PER R0 LL2
NOM-PER 'NOMBRE DE LA PERSONA' * R0 S X 40
DIR-PER 'DIRECCION DE LA PERSONA' * R0 S X 40
COL-PER 'COLONIA' * R0 S X 20
FOTO ** R0 S I
FIRMA ** R0 S I

```

Se puede ajustar el dispositivo para capturar firmas a que éstas se guarden en blanco y negro. También los formatos válidos son GIF, JPG y PNG.

Si se desea, después de capturar varias fotos y firmas, se puede calcular el AREA que usarán.

## 10) INSTRUCCIONES A LA DBD.

### 10.1) Para qué sirven?

Las instrucciones a la DBD son para aquellos casos en los que una aplicación se va a usar en varias instalaciones pero con cambios. La solución sin estas instrucciones es tener varias copias de los programas y de las definiciones, cada una con las diferentes variantes. Un ejemplo de lo anterior es la aplicación de la contabilidad, que en algunas instalaciones se requiere para una sola empresa, pero que en otras debe poder procesar varias. También en alguna instalación se necesitan 3 niveles y en otra 4.

Para ello se tienen instrucciones a la DBD que son procesadas antes de que se revisen las definiciones.

También se tienen instrucciones a CLS que son procesadas antes de revisar las instrucciones de los programas.

Básicamente las instrucciones a la DBD sirven para :

- ◆ Quitar parte de las definiciones.
- ◆ Cambiar uno o varios de los datos de las definiciones.
- ◆ Tomar la información de archivos diferentes al DBDXxx o sea de los ISxxzz.

Con el uso de estas instrucciones tanto en DBD como en CLS es posible que con una sola copia de definiciones y de programas, mediante el uso de ciertos parámetros, éstos se acoplen a las necesidades específicas de cada instalación en donde se use una aplicación.

Cabe aclarar que estas instrucciones NO se recomiendan cuando sea la primera vez que se desarrolla la aplicación o bien cuando sean los primeros desarrollos en **SISINF**.

### 10.2) Sintaxis.

El formato general de las instrucciones a la DBD, es:

& Instrucción Par1 Par2 ...

Los siguientes comentarios son aplicables:

El módulo DBD revisa si el registro a procesar tiene ampersand (&) en columna 1, si es así, asume que es instrucción a la DBD y la procesa.

- ◆ El nombre de la instrucción se puede especificar a partir de la columna 2.
- ◆ Entre la instrucción y sus parámetros (PAR1, PAR2, ...) se debe dejar uno o varios blancos.
- ◆ Si un parámetro se inicia con apóstrofe, indicará que se busque otro apóstrofe o el fin del registro como fin del parámetro. Los apóstrofes (') también se incluyen.
- ◆ Una instrucción debe estar en un solo registro, (no hay continuación).
- ◆ NO se revisa que se especifiquen más parámetros de los que requiere la instrucción, pero estos se ignoran.

### 10.3) Valor.

Se tiene una instrucción para definir el nombre de una variable al compilador y el valor que tendrá. Dicha variable se usará por otras instrucciones al compilar.

Su formato general es:

&           **VALOR**           VAR CON

&           **V**           VAR CON

- ◆ La palabra VALOR o la letra V son las que indican que es esta instrucción.
- ◆ Los siguientes comentarios son aplicables:
  - ◆ En VAR se debe especificar el nombre de la variable que se define, el cual se forma con 1 a 20 caracteres sin blancos intermedios.
  - ◆ No se permiten nombres de variables duplicados.
  - ◆ En CON se especifica el valor que tendrá la variable, este puede ser de 1 a 50 caracteres.

- ◆ Si CON tiene blancos intermedios es necesario especificarla entre apóstrofes, pero estos se incluirán en el valor que se asocia a la variable.
- ◆ Si CON se omite se asume igual a un blanco (Note que no es igual a dar ' ').
- ◆ Las VAR definidas en la DBD se pueden usar en CLS sin tener que volver a declararlas.

Los siguientes ejemplos ilustran ésta instrucción:

```
& VALOR VARIAS-EMPRESAS      SI
&      V  NUMEROS-MOVIMIENTOS 1000
& V  VAR-LET                  'LA COMERCIAL S.A.'
& VALOR  NUM-CIA
& V  ESTADISTICAS           NO
```

#### 10.4) Quitar Definiciones.

Se tienen dos instrucciones para quitar parte de las definiciones, una que marca el inicio y otra que marca el final de lo que se podrá quitar:

Su formato general es:

```
& QUITAR REF SI  VAR <>  CON
& Q      REF SI  VAR <>  CON
& FIN-QUITAR REF
& F      REF
```

Los siguiente comentarios son aplicables:

- ◆ Con la palabra QUITAR o la letra Q se indica que se marca el inicio para quitar definiciones si se cumple con una condición.
- ◆ Con las palabras FIN-QUITAR o la letra F se indica que se marca el fin de un quitar.
- ◆ En QUITAR se define una referencia (REF) de 1 a 20 caracteres, la cual se busca en una instrucción de FIN-QUITAR.
- ◆ Si la condición especificada en QUITAR se cumple se quitan las

definiciones y también las instrucciones al compilador hasta que se encuentre FIN-QUITAR con la misma referencia.

Por esta forma de trabajo ciertos errores no se diagnosticarán, tal como una REF que no se use en algún QUITAR, o dos FIN-QUITAR con la misma referencia.

- ◆ La palabra SI se debe poner en la instrucción.
- ◆ Se compara la variable VAR previamente definida con la instrucción de valor con CON para revisar si se cumple la condición es decir son iguales (=) o bien no son iguales (<>).
- ◆ Si NO se encuentra el FIN-QUITAR que se busca se dará diagnóstico.

El siguiente ejemplo ilustra el uso de esta instrucción:

```
& QUITAR ARCHIVO-EMPRESAS SI VARIAS-EMPRESAS =NO
```

```
_____
```

```
_____
```

```
_____
```

```
_____ Definiciones de DBD
```

```
& Q R1 SI ESTADISTICAS =NO
```

```
_____
```

```
_____
```

```
_____
```

```
_____ Definiciones de DBD
```

```
& F R1
```

```
_____
```

```
_____
```

```
_____
```

```
_____ Definiciones de DBD
```

```
& FIN-QUITAR ARCHIVO-EMPRESAS
```

Si la variable VARIAS-EMPRESAS es igual a NO se quitan las definiciones de DBD y las instrucciones hasta encontrar FIN-QUITAR ARCHIVO EMPRESAS.

Si la variable ESTADISTICAS es igual a NO se busca la referencia R1 y se quitan las instrucciones hasta & F R1.

Es importante señalar que no se revisa que las parejas QUITAR y FIN-QUITAR tengan la misma referencia. Así si por error en & F R1 se pone otro valor en la referencia NO se da diagnóstico y si mas adelante hay otra & F R1 ésta marcará el fin del quitar.

Si NO se encuentra un FIN-QUITAR que se busca se diagnostica, por lo que se sugiere que NO se usen referencias iguales en varios QUITAR.

### 10.5) Reemplazar Definiciones.

La instrucción de reemplazar sirve para cambiar valores en una definición de DBD. Su formato general es:

```
& REEMPLAZAR ..... &VAR ...
```

```
& R .... &VAR ...
```

Los siguientes comentarios son aplicables:

- ◆ La palabra REEMPLAZAR o la letra R es la que indica que es esta instrucción.
- ◆ Se busca un campo que se inicie con & y lo que sigue se asume que es una variable definida con VALOR.
- ◆ Se reemplaza &VAR por el valor asignado a VAR.
- ◆ Puede haber varios reemplazos.
- ◆ Si el nuevo registro formado excede la columna 72 se diagnostica.
- ◆ Una vez que se hacen los reemplazos indicados se procesa la definición en forma normal quitando el &R o &REEMPLAZAR.

El siguiente ejemplo ilustra el uso de esta instrucción:

```
&R RECURRENTE 0 &NUMERO-MOVIMIENTOS M $
  1 1 'MOVIMIENTOS' *
```

Si NUMERO-MOVIMIENTOS se definió con valor y es igual a 1000 para el proceso de la definición se tendrá:

```
RECURRENTE 0 1000 M $
  1 1 'MOVIMIENTOS' *
```

Es decir se reemplazó &NUMERO-MOVIMIENTOS por el valor de la variable que en este caso es 1000, se quitó &R y este nuevo registro es el que se procesará.

### 10.6) Incluir otro archivo.

Es posible que se incluya información de uno o varios archivos con nombre ISzzzz. El formato general de la instrucción es:

```
& INCLUIR ZZZZ DISP  
& I      ZZZZ DISP
```

Los siguientes comentarios son aplicables:

- ◆ La palabra INCLUIR o la letra "I" indica que se trata de esta instrucción.
- ◆ El nombre del archivo se forma con ZZZZ y es ISZZZZ.
- ◆ El dispositivo en donde está el archivo se especifica en DISP. Si DISP no se especifica se asume el de DBDSxx.
- ◆ Cuando se termina el archivo ISZZZZ se regresa a seguir procesando de DBDSxx.
- ◆ NO se permite tener en un archivo ISZZZZ instrucciones de INCLUIR.

Para que las ayudas se manejen en una forma estándar, se tiene en el dispositivo donde está la documentación de **SISINF** el archivo ISAYUD que contiene la información descrita en el inciso 9.2.

Este archivo se puede incluir en las diferentes DBD poniendo:

```
& INCLUIRAYUD disp
```

Donde 'disp' es el directorio, grupo y cuenta, etc... en donde está el sistema de documentación.

### 10.7) Comentarios y listados.

Si se tienen instrucciones que se inicien con &\* se asume que son comentarios de las instrucciones al compilador. Lo anterior es importante por los listados.

Cuando NO se pide la opción C al final de la compilación la información se desplegará o se imprimirá como se procesó, es decir:

- ◆ No se tendrán las instrucciones al compilador.
- ◆ No se tendrán las definiciones quitadas.
- ◆ Los remplazos estarán hechos.

Cuando se pide la opción C se desplegará o imprimirá toda la información.

En ambos casos cuando hay instrucciones al compilador de Incluir o Quitar después de la secuencia se desplegará o imprimirá:

- ◆ Las iniciales de la DBD si la información se toma de DBDSxx.
- ◆ La indentación ZZZZ si la información se toma de ISZZZZ.
- ◆ Una & si es instrucción de &R y no se dio la opción C.
- ◆ Un “-“ si la instrucción o definición se quita.

Es importante señalar que esta secuencia será la real del archivo de donde se tomó la información. Para aquellos errores en los que se despliega la secuencia alterna, se indicará con una “I” o con una “D” la fuente de la información.

## 11) FUNCIONAMIENTO DE DBD.

A continuación se describen características generales de la implantación de éste módulo en las diferentes máquinas.

Los archivos que forma el módulo DBD son:

- ◆ **ARCSxx** Archivo con las características de todos los archivos de la DBD.
- ◆ **CAMSxx** Archivo con las características de todos los campos de la DBD.
- ◆ **DIRSxx** Archivo con un directorio de campos.
- ◆ **TIPSxx** Archivo con el tipo de campo en disco y memoria de cada variable de un registro de cada archivo. El propósito de éste archivo es la validación por cambios a los archivos de la DBD.
- ◆ **VYTSxx** Archivo con los valores de las tablas que se definieron en la DBD.
- ◆ **GRPSxx** Archivo con la información de los grupos para el módulo CAS.
- ◆ **DOCSxx** Archivo para la documentación **SISINF** con las iniciales de los programas que tiene esta DBD. Este archivo es opcional.
- ◆ **COMSxx** Archivo con la información de áreas de trabajo con nombre, las cuales tendrán variables comunes.
- ◆ **VCCSxx** Archivo con la información de las variables definidas con valor y sus constantes para ser usadas por CLS.

Dado que el manejo de archivos es diferente en cada Sistema Operativo, es decir en unos se tiene el concepto de unidad de discos, en otros el de directorios, en otros el de cuentas, ... Los archivos antes mencionados se crean en donde está el archivo DBDSxx. En esta forma si la DBDXxx está en el dispositivo 10, los archivos ARCXxx, CAMXxx, ... estarán también en el dispositivo 10.

La ejecución de DBD está formada por 4 partes las cuales despliegan cada tiempo que aparece durante la ejecución de dicho módulo. Las funciones de cada una se describen a continuación.

**La parte I**, toma la información del archivo DBDSxx y del archivo ISzzzz si se encuentra instrucciones de &l. Primeramente revisa si en la columna 1 tiene & para procesarla como instrucción al compilador, si no tiene o después de procesar &R se revisan las definiciones. Para ello se pasa de formato libre a los archivos ARCSxx, CAMSxx, DIRSxx, VYTSxx, COMSxx y GRPSxx que están en formato fijo.

**La parte II**, ordena el archivo DIRSxx. Para ello se llama a ORDVX con la información necesaria para hacerlo.

**La parte III**, complementa las validaciones. Debido a que la parte 1 es un proceso de “una pasada”, es decir, se lee un registro en DBDSxx y se pasa la información a los archivos, no es posible hacer todas las validaciones necesarias, tal sería el caso de hacer referencias en VALOR a una tabla que se define después, las validaciones que se hacen son:

- ◆ Revisar tablas por VALOR.
- ◆ Revisar nombres de tablas duplicadas.
- ◆ Procesar los IGUAL.
- ◆ Revisar nombres de campos y archivos duplicados.
- ◆ Crear el archivo TIPSxx.
- ◆ Revisar áreas con nombre.

**La última parte o parte IV** es un listador ya sea en la impresora o en el CRT. En esta parte se escribe el archivo VCCSxx.

## 12) VALORES MAXIMOS.

Los siguientes son los valores máximos para la DBD:

- ◆ Archivos por DBD 400
- ◆ Tablas por DBD 200
- ◆ Areas con nombre por DBD 50
- ◆ Grupo de llaves por archivo 10
- ◆ Variables en un grupo de llaves 20
- ◆ Recurrentes por archivo 30
- ◆ Opciones SEG. en un archivo 20
- ◆ Variables por registro 125- # de dependientes
- ◆ Longitud máxima de un registro 2048 caracteres
- ◆ Longitud máxima de un grupo de llaves Nivel de registro por 4, más la longitud en caracteres de la llave < 248.

Para el cálculo de los caracteres se usa dependiendo del tipo de campo en memoria los siguientes valores:

N1	2
N2	4
N3	6
N4	8
X	Número de caracteres redondeado a par.
F	4
S	2
T	8

I	8
A	8

El tamaño máximo de un archivo si el sistema operativo lo permite y se tiene el área es de 32Gb. Ver sección 1.5.

El tamaño máximo de un campo imagen, audio o texto es de 5Mb (5,000,000 caracteres).