CURSO GENERAL DE INFORMACIÓN Y DOCUMENTACIÓN JURÍDICA.

San Sebastián, Julio 2001

TEMA:

Informática aplicada a la Documentación. Base de Datos Documental WINISIS ©

UNESCO

Antonio Bayarri Ferrer.

Nota previa. Alcance de estos apuntes

El contenido del curso va a tratar sobre la versión de ISIS para windows (WINISIS).
UNESCO preparó un manual con las diferencias que presentaba esta versión respecto a la versión anterior, que corría bajo el entorno DOS. En estos apuntes se tratan todos los aspectos teóricos y prácticos que no cubre dicho manual por ir dirigido aquel a los usuarios que ya eran conocedores de la versión DOS.

INTRODUCCIÓN

Una de las mayores ventajas para decidirse a adoptar el programa WINISIS es el gran número de usuarios que lo utilizan en el mundo. En España está instalado en más de 1500 centros (datos proporcionados por el CINDOC, en Abril de 2001), más del 90% de los cuales son entidades públicas.

El hecho de que el programa lo proporcione (gratuitamente) la Unesco, y la posibilidad de trabajar con él en tres idiomas (francés, inglés y español) nos puede dar una idea de su carácter universal.

ISIS es un sistema de almacenamiento y recuperación de información diseñado específicamente para el manejo por ordenador de bases de datos de estructura no numérica.

El elegir ISIS como sistema trabajo nos abrirá la posibilidad de manipular un número ilimitado de bases de datos, con estructuras de datos completamente diferentes entre sí.

FAMILIA DE PRODUCTOS ISIS:

Existe una gama muy amplia de productos para utilizar las bases de datos ISIS.

-**CDS/ISIS o MicroISIS.** (Usuarios finales). Programa para manejar bases de datos ISIS en entorno DOS. Es el primer producto que desarrolló UNESCO para ordenadores personales. Ha sido ampliamente extendido por todo el mundo; utiliza un compilador de lenguaje Pascal que permite añadir nuevas funciones no contempladas en la versión original. La última versión apareció en 1997. Dada su robustez, todavía está vigente en muchas instalaciones. Existe también una versión para máquinas UNÍS.

-**WINISIS**. (Usuarios finales). Programa para manejar bases de datos ISIS en entorno Windows. Desarrollado por UNESCO, la primera versión beta apareció en 1996. La última versión oficial es la 1.4 aparecida en abril de 2001. Funciona tanto el monopuesto como en redes locales.

-**WWWISIS**. (Para programadores) Plataforma que permite poner bases de datos ISIS en Internet/Intranet. Desarrollada por BIREME, la última versión, denominada WXIS se basa en XML como lenguaje de etiquetado. A través de esta herramienta, una base de datos construida por ejemplo en WINISIS, se puede hacer accesible desde Internet.

-**JAVAISIS**. (Para usuarios finales). Programa realizado en Java distribuido por UNESCO que permite acceder a bases de datos ISIS con una interfaz similar a WINISIS a través de Internet.

-**CISIS**. (Para programadores). Biblioteca de funciones diseñada por BIREME para permitir el desarrollo de aplicaciones para bases de datos ISIS (sin utilizar el software CDS-/ISIS). Las aplicaciones CISIS son plenamente compatibles con MicroISIS, incluyendo aplicaciones multiusuario.

-**ISISDLL**. (Para programadores). Biblioteca de funciones desarrollada por BIREME-UNESCO para poder acceder a bases de datos ISIS desde cualquier programa realizado con un lenguaje de programación orientado a objetos (VISUAL BASIC, DELPHI, VISUAL C, etc).

Todas las herramientas anteriormente comentadas tienen en común que se atienen al estándar ISIS, de manera que por ejemplo, podríamos crear una base de datos en WINISIS, meter datos en ella, hacer un programa en VISUAL-BASIC utilizando ISISDLL para crear un cd-rom con una interfaz determinada, ponerla en nuestra Intranet para que trabajen varios departamentos con ella a través de JAVAISIS, desarrollar varios módulos de consulta utilizando WWWISIS para que se pueda consultar a través de Internet sin peligro de que puedan modificarnos nuestra información, etc, todo sin modificar la base con la que seguimos trabajando en WINISIS.

Las herramientas licenciadas por UNESCO son totalmente gratuitas, y se distribuyen bajo licencia. WINISIS lo distribuye en España el CINDOC (Joaquín Costa 22, Madrid). El resto delos productos se pueden descargar por Internet (www.unesco.org).

Las herramientas licenciadas por BIREME (salvo ISISDLL) tienen un cargo aproximado de unas 20.000 pts (www.bireme.br).

Además de los productos mencionados, existen multitud de utilidades de libre distribución que permiten conversiones de ISIS a DBASE y a la inversa, de ISIS a Texto y viceversa, etc. Véase la página de UNESCO para acceder a dichos productos.

RELACIÓN CON EL EXTERIOR:

-ISIS respeta la norma ISO 2709 que regula el intercambio de ficheros informáticos. Por tanto, permite tanto la importación como la exportación de archivos mediante el estandar ISO.

- A través del *lenguaje de formatos* que incorpora, se pueden generar salidas diversas, no sólo en papel, sino en archivos que cumplan con los requerimientos de otras aplicaciones (ascii delimitado por ejemplo), como ACCESS, WORD, etc.

CAPACIDADES MÁXIMAS:

-Número de bases de datos..... ilimitado
-Número de registros por base..... 16 millones
-Longitud de registro (en caracteres)..... 32000
-Número de campos por base.....200
-Número de índices por base.....200
-Longitud de un campo (en caracteres).... 32000
-Número de palabras vacías.....799

CARACTERÍSTICAS QUE DEBE TENER LA INSTALACIÓN:

WINISIS precisa de un ordenador compatible de al menos 32MB (recomendados 64MB); el programa ocupa unos 5MB, a lo que habrá que añadir lo que ocupen nuestras bases de datos. Tanto el programa como las bases de datos pueden estar en un servidor de red.

ALGUNAS VENTAJAS SOBRE EL ENTORNO:

-Una ventaja interesante del ISIS es que permite tener los diferentes archivos que forman la base de datos en distintos discos, con lo que se puede jugar con el espacio disponible en los diferentes discos. Esto también permite proteger de alguna forma la información pues se podría utilizar el disco A para tener una parte de los archivos sin los cuales no se puede acceder a la base, y llevárselo al terminar la sesión.

-Es multilingüe. UNESCO lo suministra en tres idiomas (inglés, francés y español). Con relativa facilidad se pueden añadir otros idiomas.

-WINISIS utiliza también el concepto de PERFILES a través de los cuales se pueden definir distintos entornos de trabajo. Así, al identificarse un usuario, éste llevará implícito un perfil, en el que tendrá limitado (o no) el acceso a determinadas operaciones como entrada de datos, consulta, modificación de la estructura de la base, etc, así como el "recuerdo" de su entorno: última base accedida, cómo le gusta visualizar los resultados de una búsqueda, etc.

CARACTERÍSTICAS DOCUMENTALES:

-Se puede utilizar un archivo con sinónimos.

-Existe un archivo de palabras no significativas (palabras vacías) que actúa sobre todos los campos indizados.

-La base admite subcampos.

-Pueden utilizarse hasta nueve formas de indización:

a) por líneas
b) por líneas y subcampos
c) por elementos encerrados entre **<>**
d) por elementos encerrados entre //
e) por palabras
f) por subcampos con un prefijo
g) por elementos encerrados entre <> con un prefijo
h) por elementos encerrados entre // con un prefijo
i) por palabras con un prefijo

-Los formatos de entrada de datos pueden generarse fácilmente por el usuario. Pueden tenerse diversos formatos de entrada. Los formatos tienen una ayuda a nivel de campo que se introduce en la creación del formato. Se pueden poner valores por defecto en la generación del formato; estos valores pueden modificarse por sesión. También se pueden establecer validaciones (obligatoriedad de un campo, chequeo contra diccionarios, etc).

-Se dispone de un pseudo-lenguaje muy asequible para diseñar formatos de visualización de resultados e impresión, lo que da una potencia muy grande al producto para editar informes, catálogos, índices, etc. También se dispone de un asistente para la creación de formatos de gran utilidad.

-Las modificaciones pueden hacerse bien registro a registro, bien a partir de los resultados de una búsqueda determinada.

-El lenguaje de búsqueda es sencillo a base de operadores booleanos. Permite la consulta al diccionario desde el que se pueden seleccionar los términos por los que deseamos buscar.

-Las expresiones de búsqueda pueden ser de cuatro tipos: búsquedas de términos precisos, búsquedas de términos truncando, búsquedas de sinónimos y búsquedas sobre texto libre. También permite búsquedas por proximidad.

-Las etapas de búsqueda se pueden combinar. También pueden salvarse en un archivo para editarlo más tarde.

CONCEPTOS BÁSICOS

En este capítulo vamos a familiarizarnos con la terminología que utiliza ISIS, tanto en lo relacionado con el manejo del programa como en lo relativo a los elementos que constituyen las bases de datos.

BASE DE DATOS. En términos generales, podemos comparar una base de datos con un archivo de datos del mismo tipo almacenados para satisfacer las necesidades de información de un conjunto de usuarios. Esto puede ser desde un simple archivo de agenda telefónica hasta un archivo más complejo como un archivo de películas o de publicaciones.
Más adelante veremos que una base de datos está compuesta por varios archivos.

DOCUMENTO. Así denominamos a cada unidad de información almacenada en la base de datos. Por ejemplo, en una base de datos de libros, el documento puede ser **la obra**, aunque ésta conste de varios volúmenes; o bien puede ser **el volumen**... Será lo primero que se tendrá que decidir a la hora de crear una base de datos.

CAMPO. Los documentos se subdividen en elementos con características particulares llamados campos. Por ejemplo, en una base de datos de libros, un campo será el **título**, otro el **autor**...

Los campos en ISIS pueden ser de cuatro tipos:

- **X** alfanuméricos. Admiten todo tipo de caracteres.
- **A** alfabéticos. Sólo admiten los caracteres del alfabeto.
- **N** numéricos. Sólo aceptan números (y el signo **-**).
- **P** patrón. Se utiliza cuando se quiere controlar que la entrada de datos se ajuste a un patrón. ISIS no admitirá que se introduzca ningún carácter que no se ajuste a dicho patrón.

REGISTRO. Cada conjunto de campos de un documento se almacena físicamente en la base de datos sobre un **REGISTRO**. En la práctica, llamaremos indistintamente **REGISTRO** o **DOCUMENTO**.

MFN. Número interno de registro. ISIS numera automáticamente los registros según se introducen en la base de datos. Podemos no hacer caso de éste

número, aunque más adelante comprobaremos su utilidad para la generación de catálogos e índices.

SUBCAMPO. Un campo puede estar dividido en varios subcampos. Por ejemplo, un campo llamado **fecha** podría tener los subcampos **día**, **mes**, y **año**. Esto permitirá referirnos a cada uno de los subcampos o al campo completo. Cada subcampo se identifica con un carácter **delimitador de subcampo**, precedido del signo **^**.

Ejemplo:

Si los delimitadores de los subcampos **día**, **mes**, **año** son **d**, **m** y **a** respectivamente, la fecha **13 MARZO 1991** se introduciría:

^d13^mMARZO^a1991, como se verá en el capítulo **ENTRADA DE DATOS**.

CAMPO REPETIBLE. Es un tipo de campo en el que pueden darse diversas entradas, para lo que ISIS prevé un tratamiento especial tanto en la visualización como en la entrada de documentos. Cada entrada distinta se denomina **OCURRENCIA**. Por ejemplo, podemos definir como repetible el campo **autor**, ya que puede darse el caso de que una misma obra esté compuesta de varios autores.

Para indicar a ISIS que queremos crear más de una ocurrencia en el mismo campo podemos hacerlo de dos formas: utilizar un icono que hay a la izquierda del campo, el cual nos abrirá otro campo en limpio, o separar las ocurrencias distintas con un signo separador (si no se especifica nada en la instalación del programa, el separador es el tanto por ciento %)

Ejemplo:

Tenemos los autores: **CARLOS FUENTES, MIGUEL ANGEL ASTURIAS, GUILLERMO CABRERA INFANTE**. El campo autor se cumplimentará:

CARLOS FUENTES%MIGUEL ANGEL ASTURIAS%GUILLERMO CABRERA INFANTE

FDT (Field definition table). La relación de campos que componen una base de datos se denomina **FDT** o **Tabla de definición de campos**.

Se definen no sólo los campos, sino su tamaño, tipo (alfanumérico, alfabético, numérico o patrón), si es repetible o no y los delimitadores de los subcampos.

ARCHIVO MAESTRO. Archivo donde se almacenan los registros de una base de datos. Este archivo es de **longitud variable**, es decir, utiliza únicamente el espacio que ocupan los documentos introducidos, aunque en la definición de la base de datos se haya dado un tamaño máximo de ocupación superior.

DICCIONARIO DE TÉRMINOS . ISIS genera un diccionario a partir de los documentos introducidos en la base.

En la creación de la base de datos se decide qué campos van a vaciar sus términos en el diccionario, y cómo.

El diccionario puede estar dividido en subdiccionarios o **ÍNDICES**, cada uno correspondiente a un campo o conjunto de campos.

En consulta, se podrá buscar sobre todo el diccionario, sobre un índice sólo o sobre un grupo de índices.

Ejemplo:

En una base de datos de películas, podemos tener dos índices con los contenidos:

1. **DIRECTOR, PRODUCTOR, INTERPRETES**
2. **TITULO, RESUMEN**

Cuando queramos buscar a **PAUL NEWMAN**, lo haremos acotando al índice **1**, y recuperaremos todos los documentos en los que aparece este señor, bien como intérprete, bien como director o productor.
Cuando queramos buscar la palabra **EVASION**, lo haremos acotando al índice **2**, recuperando los documentos en los que aparece esta palabra en el título o en el resumen.

El problema será que con esta configuración de índices no podremos acotar la búsqueda al campo intérprete, pues en el índice **1** no distingue entre los tres campos. Por el mismo motivo, tampoco podremos buscar la palabra **EVASIÓN** sólo en el campo título.

Para poder distinguir entre los cinco campos debemos crear un índice para cada uno:

1. **DIRECTOR**
2. **PRODUCTOR**
3. **INTERPRETE**
4. **TITULO**
5. **RESUMEN**

ARCHIVO INVERTIDO. Es el archivo donde físicamente se guarda el **DICCIONARIO DE TÉRMINOS**. En la práctica utilizaremos estas dos expresiones indistintamente.

Una vez creados, modificados o borrados los documentos hay que actualizar el archivo invertido. Mientras no se actualice este archivo, no se podrán realizar consultas en el **diccionario de términos** de los documentos actualizados.

TÉRMINO DEL DICCIONARIO. Cada una de las entradas del diccionario. Un término puede ser un campo completo, un fragmento de un campo, una frase, una palabra, un trozo de un campo más una palabra, etc.

Por otra parte, en el diccionario estarán términos generados por campos textuales (como título, resumen...) y términos generados por campos descriptores (como temas, lugares geográficos...). Será el usuario el que distinga si un término es o no un descriptor, ya que desde el punto de vista de tratamiento informático no hay diferencia.

FST. (Field Select Table). La gestión de los índices la soporta ISIS sobre lo que denomina **FST** o **Tabla de Selección de Campos**.

Sobre una **FST** se definen los criterios para la extracción de los términos del diccionario partiendo de la información de los documentos.

A partir de lo que se define en la **FST** se genera el diccionario de términos, aunque se pueden tener otras **FST** predefinidas para utilizarlas para obtener catálogos o índices. En la creación de la base de datos hay que definir la **FST** con la que se extraerán los términos del diccionario.

En definitiva, en la **FST** se define cómo va a estar compuesto el **DICCIONARIO DE TÉRMINOS** que se almacena en el **ARCHIVO INVERTIDO**.

MENÚ. En Winisis existen menús desplegables que se refieren a grandes grupos de operaciones (base de datos, mostrar, búsqueda, editar, configurar, utilidades, ventana y ayuda) donde se ofrecen distintas opciones de tratamiento de las que el usuario debe seleccionar la que necesita.

Se puede tener definida, por comodidad, una base de datos por defecto. Si es así, al entrar en WINISIS aparecerá el primer registro de dicha base.

HOJA DE TRABAJO u HOJA DE ENTRADA DE DATOS. Así se denomina a las distintas plantillas que podemos utilizar para introducir documentos en nuestra base. Por ejemplo, podemos tener una plantilla con todos los campos de la base de datos y que todos sean de libre cumplimentación, otra plantilla en la que aparezcan sólo tres campos y que uno de ellos sea obligatorio de cumplimentar, y otro que en caso de que se cumplimente el descriptor introducido deba existir previamente en el diccionario, etc.

FORMATO DE VISUALIZACIÓN / IMPRESIÓN. Son formatos que se definen para visualizar los documentos (tanto en pantalla como en impresora) con diferentes aspectos: unas veces interesará ver sólo una serie de campos, otras que la salida tenga una presentación determinada... También se denominan **formatos de salida.** En Winisis además se pueden utilizar muchos comandos de entorno gráfico (fuentes, tipografía, fondos, cajas, etc).

Los formatos se definen utilizando un pseudo-lenguaje interpretable por ISIS. Cuanto más dominio se tenga de éste, mejores resultados se podrán obtener. Véase el apartado **LENGUAJE DE FORMATOS** donde se estudia la sintaxis y los comandos con los que se crean los formatos.

PALABRAS VACÍAS. Conjunto de palabras no significativas que no van a volcarse en el diccionario de términos.

Normalmente serán los artículos, las preposiciones, los adverbios... y todos aquellos términos por los que no vamos a buscar nunca y que están en un gran número de documentos. Las palabras vacías no son siempre las mismas para todas las bases de datos.

Ejemplo:

El término **LA** en una base de datos de discos no debe ser declarado palabra vacía, pues no podríamos recuperar una sonata en **LA** mayor.

Las palabras vacías están en un archivo llamado **ARCHIVO DE PALABRAS VACÍAS**, que se genera con un editor de textos tipo BLOCK DE NOTAS, con unas reglas muy simples: deben ir una en cada línea, ajustada cada palabra a la izquierda, en mayúsculas, y ordenadas alfabéticamente. Se almacenan en la misma carpeta que el resto de la base de datos, con el mismo nombre que hayamos dado a la base y con la extensión **.STW**

El uso de las palabras vacías reduce la ocupación en disco, disminuye el tiempo de búsqueda y permite tener mejor control del diccionario de términos.

A la hora de emitir un catálogo se puede utilizar un archivo de palabras vacías distinto al asociado a la base de datos.

TÉRMINOS ANY (SINÓNIMOS). Podemos asociar un archivo de sinónimos que será de gran utilidad de tal forma que cuando hagamos consultas de un término precedido por **ANY**, buscará también sobre todos los sinónimos que hayamos definido para dicho término. Este archivo tiene una estructura como la siguiente:

```
ANY SCANDINAVIA             SCANDINAVIA
ANY SCANDINAVIA             DENMARK
ANY SCANDINAVIA             FAROE ISLANDS
ANY SCANDINAVIA             FINLAND
ANY SCANDINAVIA             GREENLAND
ANY SCANDINAVIA             ICELAND
ANY SCANDINAVIA             NORWAY
ANY SCANDINAVIA             SWEDEN
ANY AGRICULTURE             AGRICULTURAL ECONOMICS
ANY AGRICULTURE             LAND ECONOMICS
ANY AGRICULTURE             AGRICULTURAL ENTERPRISES
ANY AGRICULTURE             AGRICULTURAL EQUIPMENT
ANY AGRICULTURE             AGRICULTURAL PRODUCTION
ANY AGRICULTURE             FISHERY
```

Se almacena en con el mismo nombre que la base de datos y extensión **.ANY** en la misma carpeta que la base de datos. Se trata de una lista de equivalencias para la búsqueda es decir, los términos no tienen por qué ser sinónimos realmente; nosotros podemos declarar como sinónimos de **Europa**: **Francia, Alemania, Inglaterra...**

LENGUAJE DE BÚSQUEDA

OPERADORES DE BÚSQUEDA SOBRE ARCHIVO INVERTIDO

+ **O** (OR). Sintaxis: [término1 + término2]. Seleccionará los documentos que contengan **término1** o **término2** o que contengan los dos.

Ejemplo:

FRANCISCO + PACO

***** **Y** (AND). Sintaxis: [término1 * término2]. Seleccionará los documentos que contengan **término1** y **término2**. La diferencia con **+** es que aquí deben existir los dos términos en el documento para que sea seleccionado, mientras que en **+** bastará con que el documento contenga uno de los dos términos para que sea seleccionado.

Ejemplo:

DAOIZ * VELARDE

^ **NO** (NOT). Sintaxis: [término1 ^ término2]. Seleccionará los documentos que contengan **término1** siempre que no contengan además **término2**.

Ejemplo:

POLITICA ^ FELIPE GONZALEZ

$ **TRUNCAMIENTO**. Sintaxis: [descri$]. Seleccionará todos los documentos que tengan términos que empiecen por **descri**. Seleccionaría así: descriptor, descripción, descrito...

Ejemplo:

NABUCODONO$

(G) **BÚSQUEDA SOBRE EL MISMO CAMPO**. Sintaxis: [término1 (G) término2]. Selecciona todos los documentos que contengan **término1** y **término2**, y que además estén en el mismo campo. No se especifica en qué campo. Todas las ocurrencias del mismo campo se tratan como un sólo campo.

Ejemplo:

Tenemos un campo que contiene: **JOSE SACRISTAN%FERNAN GOMEZ%PACO RABAL**.

Preguntando: **FERNAN (G) GOMEZ** se recuperará el documento.

Pero también se recuperará preguntando: **JOSE (G) GOMEZ**, ya que el sistema no distingue con este operador entre ocurrencias distintas.

(F) **BÚSQUEDA SOBRE EL MISMO CAMPO Y LA MISMA OCURRENCIA**. Sintaxis: [término1 (F) término2]. Es igual que la anterior, pero exige que los dos términos estén en la misma ocurrencia (si el campo sobre el que buscamos no es repetible lo toma todo como la misma ocurrencia).

En el ejemplo anterior, preguntando **FERNAN (F) GOMEZ** se recuperará el documento y preguntando **JOSE (F) GOMEZ** no se recuperará.

. **TÉRMINOS ADYACENTES**. Sintaxis: [término1 . término2]. Cada punto (**.**) es una palabra de distancia respecto a la primera. Un sólo punto serán dos términos adyacentes.

Ejemplo:

CAMILO . . CELA

Selecciona los documentos que tengan **término1** y **término2** en el mismo campo, y cuya separación sea como mucho el número de puntos indicados.

Ejemplo:

Instituto . . . . . emigración. Buscaremos todos los documentos que en el mismo campo contengan los términos **Instituto** y **emigración** y que estén separados como mucho 5 palabras.

Así, localizaría:

Instituto nacional de emigración
Real Instituto para la emigración
Instituto oficial del control de emigración

Y no localizaría:

Instituto para el fomento de la emigración

$ **TÉRMINOS SEPARADOS X PALABRAS**. Sintaxis: [término1 $ término2]. Selecciona todos los documentos que contengan en el mismo campo los términos indicados, y que estén separados exactamente por el mismo número de palabras que de signos **$**. Hay que tener en cuenta que el programa cuenta con que del primer término al segundo término hay una palabra de distancia, por tanto, si queremos buscar términos que contengan entre sí **una** palabra, deberemos poner **dos** signos **$**.

Ejemplos:

FRANCISCO $ RABAL. Recuperaría: FRANCISCO RABAL.

FERNANDO $ $ GOMEZ. Recuperaría: FERNANDO FERNAN GOMEZ, FERNANDO PEREZ GOMEZ, FERNANDO GARCIA GOMEZ...

/(n) **BÚSQUEDA SOBRE UN ÍNDICE DETERMINADO**. Sintaxis: [descriptor1 /(n)], siendo **n** el número de índice sobre el que queremos buscar.

Ejemplo:

Queremos buscar el término **Barcelona** en el campo **ciudad**, cuyo número de índice es el **7**. La consulta se formularía:

Barcelona /(7).

Se puede buscar sobre más de un índice separando con comas los números.

Ejemplo:

Barcelona /(7,8,9).

 # NÚMERO DE BÚSQUEDA. Cada vez que realizamos una búsqueda, ISIS le asigna un número correlativo. En una consulta podemos hacer referencia a una búsqueda anterior, combinándola con cualquier operador. Sintaxis: [#n], siendo **n** el número de búsqueda que queremos combinar.

Ejemplo:

Consulta 1: **Francia * España**

Resultado: 150

Consulta 2: **#1 * convenios bilaterales**

Resultado: 70

Esto es: En la primera consulta se localizaron 150 documentos que tenían los términos **España** y **Francia** al mismo tiempo, y en la segunda consulta se localizaron los documentos que cumplían los requisitos de la búsqueda anterior (España * Francia) y que además tuvieran el término **convenios bilaterales**.

BÚSQUEDA SOBRE TEXTO LIBRE. (BÚSQUEDA SECUENCIAL).

Hasta ahora, todas las búsquedas que se han realizado son sobre términos que han sido volcados al diccionario.

Al diccionario se vuelcan los campos por los que habitualmente se va a consultar, ya que volcar todos los campos ocupa más espacio, retarda las consultas y llena el diccionario de términos inútiles. Por ejemplo, volcar un campo llamado **PÁGINAS** no suele tener sentido, pues es rarísimo que queramos seleccionar documentos con este criterio.

WINISIS permite además realizar búsquedas sobre campos que no se vuelcan al diccionario rastreando secuencialmente la base de datos.

Se pueden recuperar documentos por distintos criterios utilizando distintas expresiones. Estas expresiones empiezan siempre con **?**. Los operadores que se ven a continuación se pueden combinar entre sí. Por ejemplo, se puede pedir que un campo sea mayor o igual otro, etc.

Algunos operadores pueden utilizarse tanto con expresiones numéricas como alfabéticas. (Mayor que, menor que, igual que). La expresión que venga junto al operador deberá ir entrecomillada si es alfabética, como se verá a continuación.

Hay que tener en cuenta que los operadores que se estudian a continuación sirven para que WINISIS evalue si una expresión es verdadera o falsa, y recuperará aquellos documentos que cumplen dicha condición. No sirven, sin embargo, para mostrar el resultado de estas expresiones. Por ejemplo, podemos pedirle que nos recupere aquellos documentos en los que la suma de los contenidos de los campos 1, 2 y 3 sea mayor que 1000. ISIS evaluará cada uno de los documentos, y lo dará por válido o no, sin embargo, no nos indicará: *éste suma 400, éste 800, etc.*

Estos operadores son los siguientes (se ven directamente con un ejemplo):

: **? V1 : 'CASA'**

Recuperará los documentos que en el campo 1 tengan:
CASA, **CASACA**, **ESCASA**, etc. Es decir, todos los documentos que **CONTENGAN LA EXPRESIÓN ENTRECOMILLADA** en el campo 1, independientemente de que dicha expresión forme parte de alguna palabra.

= **? V1 = 'CASA'**

Recuperará los documentos que en el campo 1 tengan: **CASA**. Es decir, todos los documentos en los que en el contenido del campo 1 sea **EXACTAMENTE IGUAL** a la expresión (**CASA**).

> **? V1 > 'FERNANDEZ'**

Recuperará aquellos documentos en los que el contenido del campo 1 sea **ALFABÉTICAMENTE MAYOR** que **FERNÁNDEZ**: FERNANDO, GERARDO...

< **? V1 < 'FERNANDEZ'**

Recuperará aquellos documentos en los que el contenido del campo 1 sea **ALFABÉTICAMENTE MENOR** que **FERNÁNDEZ**: FERNANDA, ERNESTO, ALEJANDRO...

<= **? V1 <= 'FERNANDEZ'**

Recuperará aquellos documentos en los que el contenido del campo 1 sea **ALFABÉTICAMENTE MENOR O IGUAL** que **FERNÁNDEZ**: FERNÁNDEZ, FERNANDA, ERNESTO, ALEJANDRO...

>= **? V1 >= 'FERNANDEZ'**

Recuperará aquellos documentos en los que el contenido del campo 1 sea **ALFABÉTICAMENTE MAYOR O IGUAL** que **FERNÁNDEZ**: FERNÁNDEZ, FERNANDO, GERARDO...

VAL **? VAL(V1) > 100**

Recuperará los documentos que el **VALOR NUMÉRICO** de la expresión (campo 1) sea mayor que **100**. Obsérvese que la comparación aquí no es alfabética.

Ejemplo de uso: en una base de datos de libros queremos recuperar aquellos libros que tienen más de cien páginas.

RSUM **? RSUM(V1,V2,V3) > 50**

Recuperará aquellos documentos en los que la **SUMA** de la expresión (los campos 1, 2 y 3) sea superior a **50**.

Ejemplo de uso: en una base de datos de test queremos sacar los nombres de las personas que entre las pruebas 1, 2 y 3 han obtenido más de 50 puntos.

RMAX **? RMAX(V1,V2) > 4.5**

Recuperará aquellos documentos en los que el **VALOR MÁS GRANDE** de la expresión (campos 1, 2) sea mayor que **4.5**.

Ejemplo de uso: en una base de datos con las calificaciones de los alumnos de un colegio, queremos sacar los alumnos que hayan aprobado en junio o septiembre.

RMIN ? RMIN(V1,V2,V3,V4) < 10

Recuperará aquellos documentos en los que el **VALOR MÁS PEQUEÑO** de la expresión (campos 1, 2, 3 y 4) sea menor que 10.

Ejemplo de uso: en una base de datos en las que se guardan los tiempos de los corredores en cuatro carreras queremos sacar aquellos que han conseguido hacer alguna de ellas en menos de 10 segundos.

RAVR ? RAVR(V1,V2,V3) > 9

Recuperará aquellos documentos en los que **LA MEDIA ARITMÉTICA** de la expresión (campos 1, 2 y 3) sea superior a 9.

Ejemplo de uso: En una base de datos de calificaciones de un curso se quiere premiar a todos los que la nota media de las asignaturas sea superior a 9.

OPERADORES: + - * /

Se pueden poner dentro de cualquier expresión, combinándolos con el resto de los operadores.

Ejemplo: ? RSUM(V1,V2,V3)/3 > 10

MFN ? MFN > 50

Recuperará aquellos documentos cuyo **NÚMERO INTERNO** (MFN) sea superior a 50.

OPERADORES BOOLEANOS: AND, NOT, OR.

Permiten **UNIR EXPRESIONES** para hacerlas más complejas:

Ejemplo de uso: **? VAL(V1) > 500 AND V2 = 'LITERATURA ESPAÑOLA'**

Recuperará en una base de datos de libros aquellos documentos de literatura española que costaron más de 500 pts.

FUNCIÓN P. **? P(V1)**

Selecciona los documentos que en el campo 1 **TIENE ALGUNA INFORMACIÓN**.

Ejemplo de uso: **? P(V1)**

En una base de datos de libros y revistas, recuperará sólo los que en el campo 1 (**título de revista**) tienen información.

FUNCIÓN A. **? A(V1)**

Selecciona los documentos que en el campo 1 **NO CONTIENEN INFORMACIÓN**.

Ejemplo de uso: **? A(V1)**

En la misma base de datos que en el ejemplo anterior, seleccionará los documentos que en el campo **1** (**título de revista**) no contienen información, y por tanto sabemos que son libros.

Para mayor efectividad, se puede realizar una consulta primero, y sobre el resultado de esta consulta hacer la búsqueda sobre texto libre, de manera que el recorrido secuencial sólo lo hará sobre los documentos previamente seleccionados.

Ejemplo:

Tenemos una base de datos de libros de todo tipo con 25.000 documentos. Queremos seleccionar todas las novelas que tengan en el resumen (campo **15**) la palabra **navidad**, pero este campo no ha sido volcado al diccionario. Sin embargo, tenemos un campo llamado **género** que sí que vierte sus términos en el diccionario y que tiene asociado el índice **5**. La consulta correcta sería:

CONSULTA 1: **novela /(5)**
RESULTADO: 216
CONSULTA 2: **? #1 V15 : 'navidad'**
RESULTADO: 3

De esta forma, la búsqueda sobre texto se ha realizado únicamente sobre 216 documentos, no sobre los 25.000 de la base.

La búsqueda sobre texto libre nos muestra una pantalla con el progreso de la búsqueda; número de registros a tratar, número de aciertos contabilizados, porcentaje de aciertos sobre el total de registros y registro que está tratando.

En cualquier momento se puede interrumpir la búsqueda pulsando cualquier tecla.

BÚSQUEDA UTILIZANDO SINÓNIMOS

ANY **Sinónimos**. Existe un archivo de sinónimos de gran utilidad que servirá para que cuando hagamos consultas de un término precedido por **ANY**, buscará todos los sinónimos que hayamos definido para dicho término.

Ejemplo: Si tenemos en el archivo de sinónimos los términos:

ANY CASA	**HOGAR**
ANY CASA	**VIVIENDA**
ANY CASA	**MORADA**

al formular la pregunta: **ANY CASA**

ISIS realmente estará preguntando: **HOGAR + VIVIENDA + MORADA**

Observando el ejemplo nos daremos cuenta que el término por el que preguntamos no forma parte de la búsqueda que ISIS compone. Siguiendo con el ejemplo, para que ISIS pregunte también por el término CASA, el archivo de sinónimos debe tener también una relación de dicho término consigo mismo:

ANY CASA **CASA**

ANY CASA	**HOGAR**
ANY CASA	**VIVIENDA**
ANY CASA	**MORADA**

Por otra parte, no se podrá formular la búsqueda con **ANY** por los términos de la derecha (en el ejemplo, **HOGAR, VIVIENDA, MORADA**). Para poder buscar por ellos hay que duplicar la entrada en el archivo de sinónimos. En el ejemplo, habría que definir:

ANY HOGAR	**CASA**
ANY VIVIENDA	**CASA**
ANY MORADA	**CASA**

LENGUAJE DE FORMATOS

Con el lenguaje de formatos definimos los formatos de visualización, los formatos de impresión y la forma de extraer los descriptores de los campos en la **FST**. La sintaxis y los comandos de este pseudo-lenguaje se explican a continuación:

COMANDOS HABITUALES

VISUALIZAR UN CAMPO. Para visualizar un campo debemos referirnos a él por su número precedido de una **V** (puede ir en mayúsculas o en minúsculas). Por ejemplo, para ver el campo primero pondremos **V1**. Con esta única instrucción ya podríamos tener un formato de salida en el que visualizaríamos únicamente el primer campo de los documentos seleccionados.

Ejemplo: Tenemos un campo
V1 que contiene: **CAMILO JOSE CELA**

Poniendo	Tendremos
V1	**CAMILO JOSE CELA**

Para visualizar varios campos los separamos con comas. Si no se especifica ningún otro comando, no dejará espacios entre los campos.

Ejemplo: Tenemos los campos
V1 que contiene: **CAMILO JOSE CELA**
V2 que contiene: **LA COLMENA**
V3 que contiene: **PLANETA**

Poniendo	Tendremos
V1,V2,V3	**CAMILO JOSE CELALA COLMENAPLANETA**

VISUALIZAR SUBCAMPOS. Para visualizar un subcampo debemos indicar el campo y el separador correspondiente a ese subcampo precedido del signo **^**. (Ver además el comando **modo**).

Ejemplo: Tenemos un campo **edición** con el número **5** asignado en la **FDT**, compuesto por tres subcampos cuyos separadores son **abc**, y cuyo contenido es **^aMadrid^b1984^csegunda edición**.

Poniendo	Tendremos
V5^a	**Madrid**
V5^b	**1984**
V5^c	**segunda edición**

Si queremos extraer el primer subcampo (o la información anterior a cualquier subcampo) cumplimentado en la entrada de datos se sustituye el separador de subcampo por un asterisco (*****).

Ejemplo: Tenemos el mismo caso que en el ejemplo anterior. Si queremos ver el primer subcampo cumplimentado,

Poniendo	Tendremos
V5^*.	**Madrid**

VISUALIZAR EL MFN. En la entrada de datos veremos que el programa asigna un número a cada documento llamado MFN. Para visualizarlo se pone **MFN**.

Este número está compuesto de 6 dígitos. Para visualizar otra cantidad de dígitos se pone esta entre paréntesis

.

Ejemplo: Tenemos seleccionado el primer documento de la base.

Poniendo	Tendremos
MFN	**000001**
MFN(5)	**00001**
MFN(3)	**001**

EXTRAER PARTE DE UN CAMPO:

***** un asterisco (*****) indica la posición desde la cual se quiere empezar a extraer el dato (empieza a contar desde cero).

Ejemplo: Tenemos un campo fecha con el número **5** que contiene el texto: **89-dic-31**. Si queremos sacar solamente el mes y el día,

Poniendo	Tendremos
V5*3.	**dic-31**

. un punto (**.**) indica el número de caracteres a extraer.

Ejemplo: Tenemos el mismo caso que en el ejemplo anterior. Si queremos sacar solamente el año y el mes,

Poniendo	Tendremos
V5.6	**89-dic**

*** .** Se pueden combinar los dos casos anteriores.

Ejemplos:

Poniendo	Tendremos
V5*3.3	**dic**
V5.2	**89**
V5*7,V5*2.4	**31-dic**

COMANDOS DE ESPACIADO HORIZONTAL Y VERTICAL.

Xn **insertar espacios en blanco**. Para insertar espacios en blanco se pone **X** y el número de espacios que se desee antes de formatear el campo siguiente.

Ejemplo: Tenemos los campos
V1 que contiene: **CAMILO JOSE CELA**
V2 que contiene: **LA COLMENA**

Poniendo	Tendremos
V1,X5,V2	**CAMILO JOSE CELA LA COLMENA**

Cn **Situar el cursor en una columna determinada.** La **C** en un formato sitúa el cursor (o el cabezal de la impresora) en la columna **n**. Hay que tener en cuenta que si el cursor estaba en ese momento en una posición posterior a **n**, tabulará a dicha columna pero en la línea siguiente.

Ejemplo: Con el caso anterior, si queremos tabular a la columna 10 el autor,

Poniendo	Tendremos
C10,V1	**CAMILO JOSE CELA**

/ **Ir a principio de línea.** El comando ***/*** va al principio de la línea siguiente (sólo si la anterior no está en blanco).

Ejemplo: Tenemos los campos:
V1 que contiene: **ALFONSO GROSSO**
V2 que contiene: **LA ZANJA**

Poniendo	Tendremos
V1/,V2	**ALFONSO GROSSO** **LA ZANJA**

**Cambio de línea (siempre).** Cada vez que aparece el signo # en el formato, se cambia de línea. Al contrario que el comando **/**, se pueden poner varios seguidos, y el efecto que hace es que deja tantas líneas en blanco como signos **#** encuentre.

Ejemplo: Tenemos los campos
V1 que contiene: **ZARZALEJO**
V2 que contiene: **MADRID**

Poniendo	Tendremos
V1####V2	**ZARZALEJO** **MADRID**

% **Borrar líneas en blanco.** El comando **%** sitúa el cursor detrás de la última línea escrita con información, anulando todas las líneas en blanco.

Ejemplo: Tenemos los campos
V1 que está vacío
V2 que contiene: **EL LAZARILLO DE TORMES**

Poniendo	Tendremos
V1#,%V2	**EL LAZARILLO DE TORMES**

SANGRADOS. Cuando se visualiza un dato, si no se especifica otra cosa lo hace a continuación de donde terminó el dato anterior. Para alterar esta norma se utiliza un comando de indentación o sangrado, que se pone inmediatamente a continuación del comando de visualización de campo o subcampo. Se codifica **(a,b)** donde:

a indica el número de espacios a dejar desde el margen izquierdo antes de formatear la primera línea del campo;

b indica el número de espacios a dejar desde el margen izquierdo antes de formatear las líneas siguientes.

Ejemplo: Tenemos un campo **V1** que contiene: **Me lo dijeron ayer las lenguas de doble filo que te casaste hace un mes, y me quedé tan tranquilo.**

Poniendo	Tendremos
V1	**Me lo dijeron ayer las lenguas de doble filo que te casaste hace un mes, y me quedé tan tranquilo.**
V1(10)	**Me lo dijeron ayer las lenguas de doble filo que te casaste hace un mes, y me quedé tan tranquilo.**
V1(5,9)	**Me lo dijeron ayer las lenguas de doble filo que te casaste hace un mes, y me quedé tan tranquilo.**
v1(0,8)	**Me lo dijeron ayer las lenguas de doble filo que te casaste hace un mes, y me quedé tan tranquilo.**

GRUPOS REPETIBLES. Conjunto de comandos que queremos que se ejecuten para un campo repetible. Se encierran entre paréntesis.

Ejemplo: Tenemos el campo **V1** que contiene:
FERNAN GOMEZ; FERNANDO%RABAL; FRANCISCO%LOPEZ VAZQUEZ; JOSE LUIS

Poniendo en el formato: **V1** tendremos:
FERNAN GOMEZ; FERNANDORABAL; FRANCISCOLOPEZ VAZQUEZ; JOSE LUIS

Sin embargo, poniendo: **(V1/)** tendremos:
FERNAN GOMEZ; FERNANDO
RABAL; FRANCISCO
LOPEZ VAZQUEZ; JOSE LUIS

LITERALES. Un literal es una cadena de caracteres que está dentro de unos delimitadores, y que se imprime tal cual en la salida. Existen tres tipos, y se identifican por el los signos de puntuación en los que va encerrado:

"condicionales". Sólo aparecen si el campo al que están asociados tiene información. Cuando el campo está vacío no aparece nada. Estos literales se encierran entre dobles comillas (").

Ejemplo: Tenemos los campos
V1 que contiene: **CAMILO JOSE CELA**
V2 que contiene: **LA COLMENA**
V3 que está vacío.
Poniendo el formato: **"autor:"V1/,"título:"V2/,"editorial:"v3/**, tendremos:

autor:CAMILO JOSE CELA
título:LA COLMENA

'incondicionales'. Son literales que queremos que aparezcan siempre, independientemente de si un campo tiene información o no. Se suele utilizar para poner encabezamientos y separaciones entre registros. Se ponen entre comillas simples (**'**).

Ejemplo: tenemos los mismos campos que en el caso anterior.
Poniendo el formato: **'autor:'V1/,'título:'V2/,'editorial:'v3/**,
tendremos
autor:CAMILO JOSE CELA
título:LA COLMENA
editorial:

|repetibles|. Son literales asociados a los campos repetibles, y tienen la particularidad de que aparecen tantas veces como ocurrencias tenga el campo repetible. Se encierran entre barras verticales (**|**), y deben ir inmediatamente antes (pre-literales) o después (post-literales) del campo, es decir, entre el literal repetible y el campo no puede haber ningún comando.

Si un pre-literal repetible va inmediatamente seguido del signo más (**+**), ej. **|xxxx|+**, saldrá antes de cada ocurrencia menos de la primera.

Si un post-literal repetible va inmediatamente precedido del signo más (**+)**, ej **+|xxxx|**, saldrá después de cada ocurrencia menos de la última.

Ejemplos: Tenemos el campo **V1** con el siguiente contenido: **CAMILO JOSE CELA%MARIO VARGAS LLOSA%ISABEL ALLENDE**

poniendo	tendremos
V1	**CAMILO JOSE CELAMARIO VARGAS LLOSAISABEL ALLENDE**
V1|; |	**CAMILO JOSE CELA; MARIO VARGAS LLOSA; ISABEL ALLENDE;**
V1+| * |	**CAMILO JOSE CELA * MARIO VARGAS LLOSA * ISABEL ALLENDE**
|; |V1	**; CAMILO JOSE CELA; MARIO VARGAS LLOSA; ISABEL ALLENDE**
| y también |+V1	**CAMILO JOSE CELA y también MARIO VARGAS LLOSA y también ISABEL ALLENDE**
(| - |V1/)	**- CAMILO JOSE CELA** **- MARIO VARGAS LLOSA** **- ISABEL ALLENDE**
|(|V1|)|	**(CAMILO JOSE CELA)(MARIO VARGAS LLOSA) (ISABEL ALLENDE)**

'por: 'V1+|; |	**por: CAMILO JOSE CELA; MARIO VARGAS LLOSA; ISABEL ALLENDE**

IMPRIMIR LITERAL CUANDO EL CAMPO ESTA VACÍO. Lo que permite este comando es la visualización de un literal basándose en la ausencia de un campo o subcampo sin imprimir el contenido del campo asociado.

Se codifica **N** seguido del campo (y subcampo si es necesario) al que hacemos referencia, precedido de un literal condicional.

Un uso muy claro de este comando es en el campo AUTOR, en el que podemos conseguir que aparezca el literal "ANONIMO" cuando este campo no tenga información.

Ejemplos:

contenido de **V1**	formato	resultado
CAMILO JOSE CELA	**"(Anónimo)"N1**	
CAMILO JOSE CELA	**"(Anónimo)"N1,V1**	**CAMILO JOSE CELA**
[vacío]	**"(Anónimo)"N1 (Anónimo)**	
[vacío]	**"(Anónimo)"N1,V1**	**(Anónimo)**

IMPRIMIR SÓLO EL LITERAL CUANDO EL CAMPO TIENE INFORMACIÓN. Con este comando se indica que todos los literales condicionales deben imprimirse sólo si el campo está presente (cumplimentado en el documento).

Se codifica **D** seguido del campo (y subcampo si es necesario) al que hacemos referencia, precedido de un literal condicional.

La diferencia que existe con el uso normal de literales es que aquí conseguimos imprimir sólo el literal cuando el campo tiene información, pero no necesitamos sacar la información.

De esta forma, tratamos por igual a todos los campos que tengan información, aunque las informaciones contenidas en los campos sean distintas.

En uno de los ejemplos de más abajo, cuando el campo resumen está cumplimentado se indica **tiene resumen**, aunque no se imprime el mismo.

Ejemplos:

contenido de **V1**	formato	resultado
bla,bla,bla	**"(existe bibliografía)"D1**	**(existe bibliografía)**
[vacío]	**"(contiene resumen)"D1**	
^lMadrid^a1984	**"se conoce el año"D^a**	**se conoce el año**

COMANDO MODO. WINISIS puede mostrar los datos de tres modos distintos:

modo prueba: en este modo, los datos se visualizan tal cual se introdujeron, sin quitar los caracteres de control

modo cabecera: este modo es el que se utiliza para imprimir catálogos e índices. Todos los caracteres de control desaparecen;

modo datos: es igual que el modo cabecera pero además al terminar de imprimir un campo (y también después de cada ocurrencia) le pone un punto (**.**);

Tanto en modo **cabecera** como en modo **datos**, al imprimir un campo con subcampos, sustituye los delimitadores de los subcampos por los signos siguientes (seguidos de un espacio en blanco):

^a	reemplaza por punto y coma (**;**)
^b hasta **^i**	reemplaza por coma (**,**)
resto	reemplaza por punto (**.**)

Si el primer carácter que hay en el campo es un delimitador de subcampo, no lo sustituye, lo elimina (quedaría muy feo por ejemplo que si el campo contiene **^aMadrid^b1984** el resultado: **;Madrid, 1984** ISIS lo deja: **Madrid, 1984**).

También sustituye la combinación de signos **><** por punto y coma (**;**). Esta combinación de signos se da en la entrada de datos en campos que tienen la técnica de indización **2.**

Este comando puede ir en mayúsculas o en minúsculas, indistintamente. El comando **modo** se codifica con tres caracteres:

1- **M** El primer carácter siempre es una **M**, e indica que a continuación viene un comando modo;

2- el segundo carácter especifica el modo que puede ser:
- **p** modo prueba
- **h** modo cabecera

d modo datos

3- el tercer carácter especifica la transformación de caracteres así:

u convierte los datos a mayúsculas

l deja los datos tal como estaban.

Dentro del orden indicado se pueden hacer todas las combinaciones (6): MPU, MHU, MDU, MPL, MHL, MDL. El comando **modo** puede aparecer en un formato tantas veces como sea necesario. Cada vez que se utiliza deja este modo por defecto para todos los campos que vengan a continuación, hasta que se encuentre otro nuevo comando **modo**.

Ejemplos:
Primero: Tenemos un campo **V1** con el contenido: **<Un> sistema de recuperación de información que realiza la gestión documental**

Poniendo	Tendremos
Mpl,V1	**<Un> sistema de recuperación de información que realiza la gestión documental**
Mhl,V1	**Un sistema de recuperación de información que realiza la gestión documental**
Mdl,V1	**Un sistema de recuperación de información que realiza la gestión documental.**
Mdu,V1	**UN SISTEMA DE RECUPERACION DE INFORMACION QUE REALIZA LA GESTION DOCUMENTAL.**

Segundo: Tenemos un campo **V5** que contiene: **^aParís^bUnesco^c1965**

Poniendo	Tendremos
Mpl,V5	**^aParís^bUnesco^c1965**
Mhl,V5	**París, Unesco, 1965**
Mdu,V5	**PARIS, UNESCO, 1965.**

COMANDOS Y FUNCIONES DE USO AVANZADO

EXPRESIONES Y FUNCIONES.

Expresiones numéricas: constantes como **5 100 17.94**. Permite notación exponencial **1.5E5**.

Función VAL(argumento). Devuelve el valor numérico del argumento. ISIS rastrea hasta encontrar el primer número dentro de la expresión, a partir de que lo encuentra y hasta que encuentra un carácter no numérico, va formando la cadena de números que dará como resultado. Si no encuentra ningún número, la expresión tendrá valor **0**. Hay que tener en cuenta que el signo **menos** (**-**) lo interpreta también como un número.

Ejemplo: (siendo **V1=10**, **V2^a=50**, **V2^b=30**)

poniendo	tendremos
VAL('15.79')	**15.79**
VAL(V1)	**10**
VAL(V2)	**50**
VAL(V2^a)	**50**
VAL(V2^b)	**30**
VAL('abril 1991')	**1991**
VAL('abril-mayo 1991')	**0**

Función RSUM(argumento). Devuelve la suma de uno o más valores numéricos expresados como argumento.

Ejemplo: (siendo **V1=10**, **V2=50**, **V3=30**)

Poniendo	tendremos
RSUM('1,20,17')	**38**
RSUM(V1,V2,V3)	**90**
RSUM(V1,'15')	**25**

Función RMIN(argumento). Devuelve el valor más pequeño entre los indicados en el argumento.

Ejemplo: (siendo **V1=20**, **V2=15**, **V3=12** y **V4=3%7%6** (campo repetible)

poniendo	tendremos
RMIN(V1,V2)	**12**
RMIN(V1,'15')	**15**
RMIN(V4|,|)	**3**
RMIN(V4|,|,'2')	**2**

Función RMAX(argumento). Como la expresión anterior, pero devuelve el valor mayor.

Ejemplo: (con los mismos valores que en el ejemplo de la función **RMAX**)

poniendo	tendremos
RMAX(V1,V2)	**20**
RMAX(V1,'15')	**20**
RMAX(V4|,|)	**7**
RMAX(V4|,|,'2')	**7**

Función RAVR(expresión). Retorna la media de los valores indicados en la expresión.

Ejemplo: (con los mismos valores que en el ejemplo de la función **RMAX**)

poniendo	tendremos
RAVR(V1,V2)	**17.5**
RAVR(V1,'15')	**17.5**
RAVR(V4|,|)	**5.3**
RAVR(V4|,|,'2')	**4.5**

Operadores: + - * /. Se pueden poner dentro de las expresiones, combinándolos con el resto de los operadores.

Ejemplo: (con los mismos valores que en el ejemplo de la función **RMAX**)

poniendo	tendremos
VAL(V1)+25	**45**
VAL(V1)-VAL(V2)	**5**
VAL(V1)/2	**10**
VAL(V4|+|)	**16**

En todos los casos estudiados anteriormente, hay que tener en cuenta que los valores que se devuelven no aparecen en la pantalla, sino que, al formar parte de una expresión condicional, su resultado es analizado, devolviendo el valor Verdadero o Falso.

Así, en el caso de la función RSUM, aunque hemos puesto el valor que devuelve, en un formato nunca aparecerá dicho valor. Ésta función deberá formar parte de una condición (véase más adelante) que devolverá el valor Verdadero o Falso:

if RSUM(V1,V2,V3) > 50 then 'la suma es mayor de cincuenta:' v1/,v2/,v3/ **fi**

(Ésto es: si la suma de los campos 1,2 y 3 es mayor de 50, entonces que imprima los campos 1,2 y 3).

OTRAS FUNCIONES AVANZADAS. En este apartado vamos a estudiar algunas funciones del lenguaje de formatos menos usuales pero muy potentes. Se utilizan dentro del contexto de los formatos de salida.

L Esta función utiliza el texto utiliza el texto producido por el argumento como un término de búsqueda sobre el archivo

invertido, y devuelve el **MFN** de la primera ocurrencia (si la hay).

Antes de buscar sobre el archivo invertido convierte la expresión a mayúsculas.

La función **L** se utiliza normalmente en conjunción con la función **REF** que se ve a continuación.

Ejemplo: Tenemos el campo **1** con que contiene **CAMILO JOSE CELA** que está en el registro **18**.
La función
L("camilo jose cela")

nos dará 18 como resultado (siempre que el campo **1** esté indizado adecuadamente).

REF

La función **REF** nos permite a partir de un argumento, acceder a un registro de la base de datos y extraer información de otro campo.

Esto permite que se tengan diferentes bases de datos lógicas interrelacionadas sobre una única base de datos física.

Esta función tiene dos parámetros:

-el primero puede ser una expresión o un número con la que se indica el **MFN**

-el segundo es el campo a extraer del **MFN** anterior.

Por ejemplo, **REF(3,V1)** significa: del registro cuyo **MFN** es **3**, imprime el campo **1**.

Esto no es muy operativo, ya que normalmente no se conoce el **MFN**, y además, como esta función se pone en un formato, tendríamos que estar modificando constantemente el mismo para cambiar el **MFN**. Sin embargo, hay ocasiones

en que sí se utiliza, por ejemplo, cuando en el primer registro de la base tenemos información de control...

Lo normal es introducir una expresión que nos sirva para todos los casos.

Con la función **L** hemos visto que localizamos el **MFN** de un argumento. Podemos combinar las dos funciones.

Si no se combinan las dos funciones (**REF** y **L**), nos veríamos obligados a conocer el **MFN** de cada registro.

Ejemplo:

Tenemos una base de datos de una biblioteca con los campos:

1. AUTOR
2. TITULO
3. EDITORIAL
4. EDICION
5. RESUMEN
6. BIOGRAFIA DEL AUTOR
7. FECHA DE NACIMIENTO
8. MOVIMIENTO LITERARIO

Como sería una barbaridad que cada vez que se diera de alta un **título** se volviera a introducir los datos del autor (**biografía, fecha de nacimiento y movimiento literario** al que pertenece), lo mejor es tener dos formatos de entrada: uno para libros y otro para autores.

El **formato1** que lo utilizamos para introducir títulos y tiene los campos **1,2,3,4,5**.

El **formato2** que lo utilizamos para dar entrada a autores nuevos con los campos **1,6,7,8**.

Como se ve es como si se tuvieran dos bases de datos distintas, una de **autores** y otra de **títulos**. Estas dos bases de datos tendrían un campo en común, que es el **autor**.

Introducimos en la base de datos los siguientes documentos:

MFN 1
autor: CAMILO JOSE CELA
título: LA COLMENA
editorial: PLANETA
edición: MADRID, 1984
resumen: OBRA MUY INTERESANTE....
biografía:
fecha de nacimiento:
movimiento literario:

MFN 2
autor: CAMILO JOSE CELA
título: LA FAMILIA DE PASCUAL DUARTE
editorial: PLAZA Y JANES
edición: MADRID, 1989
resumen: OTRA OBRA TAMBIEN MUY INTERESANTE...
biografía:
fecha de nacimiento:
movimiento literario:

MFN 3
autor: CAMILO JOSE CELA
título:
editorial:

edición:
resumen:
biografía: NACIO EN PADRON, RECIENTE PREMIO NOBEL, ETC...
f.nacimiento: 12/10/92
mov.literario: GENERACION DE LOS 50

Como se ve, los registros 1 y 2 se han introducido con el **formato1** y el registro 3 con el **formato2**. Por tanto, no habría problema para visualizarlos con los correspondientes formatos de salida.

Tenemos dos formatos de salida:

salida1: **V1,V2,V3,V4,V5** que lo utilizamos para ver los datos relativos a los títulos de los libros

salida2: **V1,V6,V7,V8** que lo utilizamos para ver los datos relativos a un autor.

Estos formatos son normales, no tienen dificultad porque en la visualización, en cada uno de ellos estamos viendo campos que pertenecen al mismo registro.

Ahora supongamos que queremos ver por cada título los datos pertenecientes al autor.

Tenemos el formato **salida3**: **V1,V7,V2,V3,V4,V8**

Los campos **7** y **8** son **fecha de nacimiento** y **movimiento literario**, respectivamente, y sólo lo cumplimentamos en los documentos que contienen datos biográficos del autor (en el ejemplo, el **MFN 3**).

Si visualizamos el documento con **MFN 1** con el formato **salida3**, los campos **7** y **8** estarán vacíos.

Vamos a ver tres soluciones posibles.

SOLUCION 1

Conociendo el **MFN** del registro donde están los datos del autor.

El formato de salida **salida3** deberá quedar así:
V1,REF(3,V7),V2,V3,V4,REF(3,V8).

Literalmente le estaríamos diciendo:

-imprime el campo **1**;
-del registro cuyo **MFN** es **3**, imprime el campo **7**;
-imprime el campo **2**;
-imprime el campo **3**;
-imprime el campo **4**;
-del registro cuyo **MFN** es **3**, imprime el campo **8**.

Problema: cada vez que queramos cambiar de autor debemos modificar el formato **salida3**

SOLUCION 2

No conociendo el **MFN** del registro donde están los datos del autor, pero teniendo un campo en el registro que nos indique el **MFN** de dicho registro. Para ello añadimos un campo **9** llamado por ejemplo, **MFN DEL AUTOR**. Cada vez que introducimos un libro en la base, miramos previamente el **MFN** del registro donde tenemos los datos del autor (para ello sería conveniente tener un listado de autor y mfn), y lo ponemos en el campo **9**.

El formato de salida **salida3** deberá quedar así:
V1,REF(V9,V7),V2,V3,V4,REF(V3,V8).

Literalmente le estaríamos diciendo:

-imprime el campo **1**;
-del registro cuyo **MFN** está indicado en el campo **9**, imprime el campo **7**;
-imprime el campo **2**;
-imprime el campo **3**;
-imprime el campo **4**;
-del registro cuyo **MFN** está indicado en el campo **9**,imprime el campo **8**.

Problema: nos obliga a buscar el **MFN** del autor cada vez que introducimos un registro nuevo. Y sobre todo, no podemos reorganizar la base de datos descargándola y cargándola, pues se desajustarían los **MFN**, y el contenido del campo **9** no apuntaría realmente a los documentos de los autores.

SOLUCION 3

No conociendo el registro donde están los datos del autor, y localizándolos por medio de la función **L** combinada con la función **REF**.

Modificamos el formato **salida3** de la siguiente manera: **V1,REF(L(V1),V7),V2,V3,V4,REF(L(V1),V8)**.

Literalmente le estaríamos diciendo:

-imprime el campo **1**;
-del registro al que has accedido por el archivo invertido a partir del campo**1**, imprime el campo **7**;
-imprime el campo **3**;
-imprime el campo **4**;
-del registro al que has accedido por el archivo invertido a partir del campo **1**, imprime el campo **8**.

PROBLEMA: La función **L** nos da el **MFN** del **primer** registro que responde al argumento de búsqueda. En este caso, el argumento de búsqueda es el contenido del campo **1**, que es: **CAMILO JOSE CELA**. Y si buscáramos en el archivo invertido, nos daría que hay **3** registros que tienen en el campo **1** este descriptor. Como la función **L** nos da el **MFN** del **primer** registro, la función **REF** tomará el contenido de los campos **7** y **8** pero DEL PRIMER REGISTRO, que no contiene nada.

SOLUCION: Crear el registro correspondiente al autor antes que los registros de libros, y el **MFN** de la función **L** será siempre el de la primera ocurrencia.

Existe otra solución más complicada, que es crear un campo **9** nuevo que sólo se cumplimentaría en los registros tipo biografía, y que nuevamente llevaría el nombre del autor, es decir, estaría duplicados los campos **1** y **9** en los registros de biografías. Para que en el archivo invertido no sean iguales, debe añadirse algún identificador en la **FST** para el campo **9**, de forma que quedara por ejemplo: **9 0 "autor="V9** (ver capítulo **TECNICAS DE INDIZACION**).El formato debería ser entonces:
V1,REF(L('autor='V9),V7),V2,V3,V4,REF(L('autor='V9),V8)

funciones realizadas en Pascal. La sintaxis es:

&nombre(formato) donde:

&	indica que es una función Pascal;
nombre	es el nombre de la función;
formato	es el formato de salida del argumento.

Existen diversos programas realizados en IsisPascal que se pueden obtener a través del Web de UNESCO. Por ejemplo, a través de una función en IsisPascal se puede verificar si un NIF es correcto (conociendo el algoritmo que lo genera), etc.

P(campo selector).

Es una función booleana que retorna el valor **Verdadero** si el registro que está siendo formateado contiene al menos una ocurrencia de el campo o subcampo indicado en el argumento.

Ejemplo: si el campo **1** contiene **CERVANTES**, la función **P(V1)** será verdadera, ya que el campo **1** contiene algo.

A(campo selector).

Es una función booleana que retorna el valor **Verdadero** si el registro **no** contiene ninguna ocurrencia del campo indicado en el argumento.

Ejemplo: si el campo **1** contiene **CERVANTES**, la función **A(V1)** será falsa.

F Convierte valores numéricos a cadenas de caracteres, formateando la salida. Es especialmente útil para obtener índices presentables. Esta función tiene tres argumentos:

1- Valor numérico a convertir.

2- Ancho mínimo asignado. Si la longitud en caracteres del valor numérico es menor que el número asignado aquí, el número queda ajustado a la derecha dentro de este ancho.

3- Número de posiciones decimales.

Ejemplo:Queremos visualizar los MFN sólo con los números significativos, ajustados a la derecha sobre un ancho de cuatro dígitos. En un formato podríamos tener:

V1,c60,f(MFN,4,1)

Este formato produciría salidas como esta:

```
MAMBRU SE FUE A LA GUERRA        100
STICO                              4
```

IF.

Es un comando que permite crear formatos que tendrán en cuenta el contexto.

Por ejemplo, se puede crear un formato con este comando que produzca diferentes salidas dependiendo de el contenido de un campo. Se codifica así:

IF condición THEN formato-1 ELSE formato-2 FI donde:

condición es una expresión booleana;

formato-1 es el formato que se ejecutará si y sólo si la expresión booleana es verdadera;

formato-2 es el formato que se ejecutará si y sólo si la expresión booleana es falsa.

La cláusula **ELSE** es opcional y puede omitirse. Se puede poner a continuación del **THEN** el **ELSE** para conseguir que haga algo sólo cuando la condición es falsa. Formatos aceptados:

IF condición THEN formato-1 FI
IF condición THEN ELSE formato-2 FI

Se pueden anidar los comandos.

Ejemplo: **IF P(V1) THEN V24 ELSE IF P(V2) AND A(V3) THEN V5 FI FI**

(Si el campo **1** contiene algo entonces imprime el campo **24**, en caso contrario (si el campo **2** contiene algo y el campo **3** está vacío, entonces imprime el campo **5**)).

El comando **IF** se utiliza especialmente con bases de datos que contienen diferentes tipos de registro.

TÉCNICAS DE INDIZACIÓN

CONSIDERACIONES GENERALES

La gestión de los índices la soporta WINISIS sobre lo que denomina **FST** (Field Select Table) o **Tabla de Campos de Selección**.

Sobre una **FST** se definen los criterios para la extracción de uno o más descriptores de un archivo maestro.

Dependiendo del contexto en el que se utilice la **FST**, estos descriptores se utilizan:

- para crear entradas sobre un archivo invertido (o diccionario de términos) de los registros de los que fueron extraídos,

- para clasificar los registros en una secuencia determinada o

- para reformatear los registros durante una operación de importación o exportación.

Un **término del diccionario** puede ser un campo completo, un fragmento de un campo, una frase, una palabra, un trozo de un campo más una palabra, etc.

En la **FST** se indica cómo se deben construir los términos del diccionario para cada campo.

Se puede tener más de una **FST**. La **FST** cuyo nombre coincide con la base de datos es la que se utiliza para generar el archivo invertido. Además, podemos tener otras **FST** creadas de antemano para obtener catálogos sin tener que definirla en el momento de generar los listados.

La creación de la **FST** se realiza al crear la base, y se puede modificar o crear una nueva desde el menú **Definición de una base de datos**, en **modificar una definición**, la opción **E** crea o actualiza una **FST**.

Una **FST** consta de una o varias líneas. En cada línea se define un índice. El conjunto de todos los índices forman el **archivo invertido** o **diccionario de términos**. A la hora de consultar, la búsqueda la realizará sobre todo el diccionario de términos, y se podrá acotar a un índice en particular cuando así se desee.

Cada línea de la **FST** consta de tres parámetros:

IDENTIFICADOR DE INDICE. En éste parámetro se indica el número con que vamos a identificar cada índice.

Normalmente, un índice se asocia a un sólo campo. En este caso, lo habitual (aunque no obligatorio) es que hagamos coincidir el número de campo con el número de índice.

Ejemplo: Tenemos el campo **autor** con el número **V1**. En el parámetro **ID** se pone un **1**.

(Véase en **formato de extracción de datos** un ejemplo con un índice asociado a varios campos).

TECNICA DE INDIZACIÓN. La técnica de indización va codificada, y puede ser:

0 Construye términos del diccionario por **cada línea** extraída por el formato.

Hay que tener en cuenta que el formato se representa en memoria, no en pantalla, por lo que el concepto línea no está reducido a 80 caracteres; mientras en el formato no haya ninguna instrucción de salto de línea, la línea sigue siendo la misma.

Por otra parte, en la FST que se utiliza para consultar, los descriptores se truncan a partir de 30 caracteres.

A la hora de recuperar en consulta, exige conocer exactamente el contenido completo del descriptor. Hay que tener cuidado pues, al indizar campos tipo **autor**, o **personas**, pues si se cumplimentan con apellidos y nombre, en el momento de buscar se deberá especificar perfectamente.

Con esta técnica vendrán muy bien las posibilidades de consulta al diccionario de términos, y el operador de truncamiento (**$**).

También suele utilizarse para obtener catálogos o índices, ya que para esto no hace falta conocer el contenido completo de los campos.

1 Construye términos del diccionario por cada campo y subcampo extraído del formato.

2 Construye términos del diccionario por cada término o frase encerrado entre los signos "**<>**".

Cualquier otro texto que quede fuera de estos signos, no será indizado.

Suele utilizarse en campos tipo **materias**, o **temas**, pudiendo seleccionar así los términos que se volcarán al diccionario.

En el los formatos de pantalla e impresión, se ve cómo se sustituyen en visualización los signos **><** por punto y coma (**;**), de manera que si tenemos un campo que contiene: **<televisión><medios de comunicación>**, utilizando el formato adecuado, este campo se vería como: **televisión; medios de comunicación**.

Ejemplo:

Tenemos el texto
la misión de este informe es describir un <curso universitario> que permita la <formación integral>

utilizando la técnica **2** de indización, se generarán los términos:

curso universitario
formación integral

3 Es igual que la técnica **2**, con la única diferencia que los términos o frases a indizar deben encerrarse entre signos ***//***.

Visualmente es más cómodo este sistema, pero sin embargo no se pueden utilizar las funciones de edición de eliminar en visualización los delimitadores, ya que éstos, a diferencia de los signos **<>**, no son caracteres reservados.

4 Crea un término en el diccionario por cada palabra del texto extraído por el formato.

Debido a que el diccionario se llena de términos que no tienen ninguna relevancia (artículos, preposiciones, y en general, términos por los que no se buscará nunca), WINISIS contempla la posibilidad de utilizar un archivo que contenga todos los términos que no deseamos llevar al diccionario. Este archivo de palabras no significativas se denomina **archivo de palabras vacías** (de significado). Con el uso de este archivo se ahorra espacio en el disco, se reducen los tiempos de búsqueda y se tiene un diccionario más controlado. WINISIS comprobará la existencia de este archivo, y no llevará al diccionario ningún término si existe en dicho archivo de palabras vacías.

5, 6, 7, 8 Son técnicas equivalentes a la 1, 2, 3 y 4 respectivamente, pero que permiten añadir automáticamente un prefijo a cada término que se genere. Ésto ofrece la ventaja de que en el diccionario se visualizarán agrupados todos los términos del mismo índice; además, en consulta se podrán realizar búsquedas de tipo:

a=Vargas Llosa, Mario

en vez de

Vargas Llosa, Mario /(1)

FORMATO DE EXTRACCION DE DATOS, codificado utilizando el lenguaje de formatos de ISIS (ver LENGUAJE DE FORMATOS)

Antes de crear los términos en el diccionario WINISIS desarrolla el formato que se pone en este apartado, y los términos los crea a partir del resultado del formato aplicando la técnica de indización que se indicó en el parámetro anterior.

Para crear un índice de un campo, se pondrá el número de campo precedido de **V**.

Por ejemplo, si **autor** es el campo **1**, independientemente de la técnica de indexación, se pondría en el parámetro **3**: **V1**.

En búsqueda, para restringir la búsqueda al campo **autor**, se deberá indicar el número de índice asociado.

Ejemplo:

Buscamos **CARLOS BARRAL** en el índice **autor**. La expresión de búsqueda será:

CARLOS BARRAL /(1)

Las técnicas de indización 5-8 tienen una forma especial de construcción:

'/t=/',v1

Donde **t=** es el prefijo a añadir, que debe encerrarse entre los signos **'/..../'**, a continuación debe ir una coma, y después, el formato invocando al campo o campos de los que se quieren extraer términos.

Dicho de otra manera, los índices se construyen así:

- Se desarrolla en memoria el formato.
- A cada línea generada por el formato se le aplica la técnica de indización elegida, que generará más o menos términos.
- Estos términos generados en el paso anterior se almacenarán en el índice indicado.

CÓMO ASOCIAR MÁS DE UN CAMPO A UN SOLO ÍNDICE

Ejemplo:

En una base de datos de discos tenemos entre otros, los campos

autor con el número **V1**
solista con el número **V2**
grupo con el número **V3**

Queremos tener un sólo índice para los autores y los solistas y los grupos, de manera que al buscar por ejemplo **JUAN PARDO** nos recupere todos los documentos en los que aparezca ya sea como autor, como solista como grupo.

En primer lugar, tendrá que decidirse el número de índice a asociar a estos dos campos. Una posibilidad es asociarle el número del campo menor (**1** en este caso).

En segundo lugar, decidir la técnica de indexación para recuperar después en búsquedas.

- Aplicaremos la técnica de indexación **4** (por palabras) para crear un descriptor por cada una de las palabras contenidas en los campos **V1**, **V2** y **V3**. La línea de la **FST** para este índice quedaría:

ID	TI	formato de extracción de datos
1	4	v1/,v2/,v3/

En este caso, como el índice contendrá palabras sueltas, se consultaría utilizando operadores lógicos:

JUAN /(1) * PARDO /(1)

esto es:

"Busca **JUAN** en el índice **1** (formado por los campos **autor**, **solista** y **grupo**)"; [RESULTADO PARCIAL 1]
"Busca **PARDO** en el índice **1**"; [RESULTADO PARCIAL 2]
"Busca la intersección (los documentos comunes entre el resultado parcial 1 y el resultado parcial 2)".

El problema es que podrá haber documentos que tengan por ejemplo como autor a **JUAN GARCIA**, y como solista a **PEDRO PARDO** que como cumplen también los requerimientos de la búsqueda, aparecerán como **ruido** en los resultados.

Para evitar ésto, se utiliza el operador **$** (términos separados **$** palabras). La búsqueda será entonces:

JUAN $ PARDO /(1)

- Aplicaremos la técnica de indización **0** para crear un término por cada línea que genere el formato.

suponiendo los campos:

V1 (autor) que contiene...: **CARLOS CANO**
V2 (solista) que contiene.: **MARIA DOLORES PRADERA**
V3 (grupo) que contiene...: **LOS GEMELOS**

Pretendemos generar en el diccionario los términos:

CARLOS CANO
MARIA DOLORES PRADERA
LOS GEMELOS

Vamos a estudiar qué ocurre en cada caso:

Con un formato de extracción de datos como el anterior, con la técnica de indización **0**,

ID	TI	formato extracción de datos
1	0	V1,V2,V3

el formato de extracción de datos (que funciona igual que un formato de pantalla), dejaría en salida el resultado:

CARLOS CANOMARIA DOLORES PRADERALOS GEMELOS

Como la longitud máxima de un descriptor en el diccionario es de 30 caracteres, el descriptor en el índice sería:

CARLOS CANOMARIA DOLORES PRADE

Está claro que este **no** es el resultado que pretendemos obtener.

Necesitamos que salte de línea entre cada campo, para que al aplicarle la técnica de indización **0** genere un término por cada línea. El formato adecuado sería entonces:

ID	TI	formato extracción de datos
1	0	V1/,V2/,V3/

El formato de salida quedaría:

CARLOS CANO
MARIA DOLORES PRADERA
LOS GEMELOS

Y en el diccionario se crearían tres términos, uno por cada línea generada por el formato de salida.

CÓMO CREAR TÉRMINOS EN LOS CAMPOS REPETIBLES

ISIS permite crear términos compuestos de varias palabras por cada ocurrencia de un campo repetible.

Ejemplos:

Tenemos el campo **V3** (grupo) que es repetible.

En la entrada de datos se introdujo:

JOHN LENNON%PAUL McCARTNEY%RINGO STARR%GEORGE HARRISON

La línea de la **FST** quedaría:

ID	TI	formato extracción de datos
1	0	(V3/)

Al aplicarle el formato al campo, la salida quedaría:

JOHN LENNON
PAUL McCARTNEY
RINGO STARR
GEORGE HARRISON

y al aplicarle la técnica de indización **0** (por líneas), creará un término por cada una de las líneas generadas por el formato de extracción de datos.

CÓMO PONER ETIQUETAS PARA LOS ÍNDICES

El uso de los índices plantea el problema de que el usuario se ve obligado a memorizar el número de índice asociado a un campo.

Este problema se agrava cuanto mayor sea el número de índices.

Existe una solución (*) para remediar esto. Vamos a explicarlo con un ejemplo.

Tenemos los campos

V1 etiquetado con **autor** que contiene: **CARLOS BARRAL**
V2 con la etiqueta **editor** que contiene: **CARLOS BARRAL**

Para buscar **CARLOS BARRAL** en **autor**, la expresión de búsqueda será:

CARLOS BARRAL /(1)

Utilizando las posibilidades del lenguaje de formatos, podemos hacer que el formato de extracción de datos genere los términos en el diccionario con una etiqueta por delante.

En la siguiente **FST**

ID	TI	Formato extracción de datos
1	0	"autor="V1
2	0	"editor="V2

se generarán los términos:

AUTOR=CARLOS BARRAL
EDITOR=CARLOS BARRAL

(Recuérdese que ISIS convierte siempre a mayúsculas antes de crear el término).

En consulta, se pondría:

AUTOR=CARLOS BARRAL

Es importante ver que con este sistema ya no se podría hacer una búsqueda genérica sobre todos los campos de la base del término **CARLOS BARRAL**, ya que en unos casos el término sería **AUTOR=CARLOS BARRAL** y en otras **editor=CARLOS BARRAL**.

Cuando se indiza de ésta manera, al buscar por el diccionario se ve claramente a qué campo está asociado un término porque éste lleva incorporada su etiqueta, y además, como aparecen clasificados por orden alfabético, salen agrupados por campos.

Con las técnicas de indización 5-8, se puede poner un prefijo a cualquier término, aunque sea generado por palabras.

Ejemplo. Tenemos un campo **resumen** con el número **9**. Para que todas las palabras del mismo aparezcan en el diccionario con el prefijo **R=**, se creará una línea así en la FST:

```
9  8  '/R=/', V9
```

(índice 9, generado con la TI 8 (por palabras prefijadas con **R=**), cuyo formato de extracción de datos es **V9** (contenido del campo **resumen**).