

C
O
M
M
O
N
B
U
S
S
I
N
E
S
O
R
I
E
N
T
E
D
L
A
N
G
U
A
G
E

Por T. HURTADO MERELO

*Doctor Ingeniero de Telecomunicación
Profesor del Instituto de Informática*

INSTITUTO DE INFORMATICA

MADRID, 1975

© SERVICIO DE PUBLICACIONES DEL MINISTERIO DE EDUCACION Y CIENCIA

Edita: Servicio de Publicaciones del Ministerio de Educación y Ciencia.

Imprime: IMNASΛ. Menorca, 47. Madrid.

Depósito Legal: M. 16.454-1975.

I.S.B.N.: 84-369-0404-4.

Printed in Spain. Impreso en España.

*A mi mejor colaboradora,
mi esposa.*



S U M A R I O



	<i>Págs.</i>
INTRODUCCION	11
0.1. Breve reseña histórica	13
0.2. Introducción a la programación en COBOL	13
CAPITULO I.—GRAMATICA DEL LENGUAJE: MORFOLOGIA	15
I-1. Caracteres utilizados en el COBOL	21
I-1.1. Tipos de caracteres	21
I-1.1.1. Caracteres básicos	21
I-1.1.1.1. Utilización de caracteres básicos	21
I-1.1.2. Caracteres adoptados	22
I-2. Palabras formadas por el programador	25
I-2.1. Nombres	25
I-2.1.1. Tipos de nombres	25
I-2.1.1.1. Nombres de dato	25
I-2.1.1.2. Nombres de condición	26
I-2.1.1.3. Nombres de párrafo	26
I-2.1.1.4. Nombres de fichero	26
I-2.1.1.5. Nombres de dispositivo	26
I-2.1.1.6. Nombres de biblioteca	27
I-2.1.1.7. Nombres de calificador o calificadores	27
I-2.1.1.8. Nombres de índice o subíndices	27
I-2.1.2. Reglas para la formación de los distintos tipos de nombres	28
I-2.1.3. Ejemplos de distintos tipos de nombres	30
I-2.2. Constantes	30
I-2.2.1. Tipos de constantes	30
I-2.2.1.1. Constantes literales	30
I-2.2.1.2. Constantes con nombre	32
I-2.2.1.3. Constantes figurativas	33

	<i>Págs.</i>
I-3. Palabras específicas del lenguaje	37
I-3.1. Verbos	37
I-3.1.1. Verbos del programa	37
I-3.1.2. Verbos del compilador	38
I-3.1.3. Verbo de clasificación	38
I-3.1.4. Verbo de control de entrada-salida	38
I-3.2. Operadores	38
I-3.2.1. Operadores aritméticos	38
I-3.2.2. Operadores de relación	39
I-3.2.3. Operadores lógicos	39
I-3.3. Palabras reservadas	39
I-3.3.1. Palabras clave	39
I-3.3.2. Palabras optativas	39
 CAPITULO II. GRAMATICA DEL LENGUAJE: SINTAXIS	 41
II-1. El concepto de sintaxis. Frases	47
II-2. Expresiones y declaraciones	51
II-2.1. Expresiones	51
II-2.1.1. Expresiones aritméticas	51
II-2.1.2. Jerarquía en el cálculo de una expresión aritmética	51
II-2.1.3. Expresiones condicionales	52
II-2.1.4. Evaluación de una expresión condicional	52
II-2.1.5. Comparación entre elementos	53
II-2.1.5.1. Comparación entre elementos numéricos	53
II-2.1.5.2. Comparación entre elementos no-numéricos	54
II-2.1.6. Utilización de operadores lógicos en las expresiones condicionales	55
II-2.1.7. Orden de evaluación de una expresión condicional compuesta	55
II-2.1.8. Pseudoexpresiones condicionales	56
II-2.1.8.1. Condiciones de signo	56
II-2.1.8.2. Condiciones de clase	57
II-2.1.8.3. Nombres de condición	58
II-2.2. Declaraciones	58
II-2.2.1. Declaraciones imperativas	58
II-2.2.2. Declaraciones condicionales	59
II-3. Sentencias e inscripciones	63
II-3.1. Sentencias	63
II-3.1.1. Sentencias imperativas	63
II-3.1.2. Sentencias condicionales	63
II-3.1.3. Sentencias de instrucciones para el compilador	63

	<i>Págs.</i>
II-3.2. Inscripciones	63
II-3.3. Puntuación de las sentencias e inscripciones	64
II-4. Párrafos, secciones y divisiones	67
II-4.1. Párrafos	67
II-4.2. Secciones	67
II-4.3. Divisiones	67
CAPITULO III.—ESTRUCTURA DE UN PROGRAMA. HOJA DE CODIFICACION	69
III-1. Estructura de un programa COBOL	75
III-1.1. Identification Division	75
III-1.2. Environment Division	77
III-1.3. Data Division	77
III-1.4. Procedure Division	77
III-1.5. Notaciones empleadas en el estudio de las instrucciones que componen las divisiones del programa	77
III-2. Hoja de codificación COBOL	81
III-2.1. Introducción	81
III-2.2. Formato de la hoja de codificación COBOL	81
III-2.2.1. Elementos que no se transcriben al programa fuente	83
III-2.2.2. Elementos que se transcriben al programa fuente	85
CAPITULO IV.—LAS DIVISIONES DE UN PROGRAMA. IDENTIFICATION Y ENVIRONMENT DIVISION	89
IV-1. Identification Division	95
IV-1.1. Generalidades	95
IV-1.2. Contenido de la Identificación Division	95
IV-1.2.1. Párrafo Program-Id	95
IV-1.2.2. Párrafo Author	96
IV-1.2.3. Párrafo Installation	96
IV-1.2.4. Párrafo Date-Written	96
IV-1.2.5. Párrafo Date-Compiled	97
IV-1.2.6. Párrafo Security	97
IV-1.2.7. Párrafo Remarks	97
IV-2. Environment Division	101
IV-2.1. Generalidades	101
IV-2.2. Sección de configuración	101

	<i>Págs.</i>
IV-2.2.1. Párrafo Source-Computer	101
IV-2.2.2. Párrafo Object-Computer	103
IV-2.2.3. Párrafo Special-Names	103
IV-2.2.3.1. Cláusula Console	103
IV-2.2.3.2. Cláusula Currency	103
IV-2.2.3.3. Cláusula Decimal-Point	104
IV-2.2.3.4. Cláusula de nombres nemotécnicos	104
IV-2.3. Sección de entrada-salida	104
IV-2.3.1. Párrafo File-Control	105
IV-2.3.1.1. Cláusula Assign	105
IV-2.3.1.2. Cláusula Multiple-Reel	107
IV-2.3.1.3. Cláusula Reserve	107
IV-2.3.1.4. Cláusula File-Limit	107
IV-2.3.1.5. Cláusula Access	108
IV-2.3.1.6. Cláusula Organization	108
IV-2.3.1.7. Cláusulas Key	109
IV-2.3.1.8. Cláusula Processing	109
IV-2.3.1.9. Cláusula File-Code	109
IV-2.3.1.10. Cláusula Track-Area	110
IV-2.3.2. Párrafo I-O-Control	110
IV-2.3.2.1. Cláusula Same	112
IV-2.3.2.2. Cláusula Rerun	112
IV-2.3.2.3. Cláusulas Apply	113
IV-2.3.2.4. Cláusula Multiple-File	114
CAPITULO V.—LAS DIVISIONES DE UN PROGRAMA. DATA DIVISION	117
V-1. Organización de los datos	123
V-1.1. Generalidades	123
V-1.2. Estructuración de los datos en COBOL	124
V-1.2.1. Los números de nivel	124
V-1.2.2. Normas para la utilización de los números de nivel en datos elementales y de grupo	126
V-1.2.3. Números de nivel especiales	127
V-1.2.3.1. Número de nivel 66 ó número de reagrupamiento	128
V-1.2.3.2. Número de nivel 77 ó número de nivel independiente	128
V-1.2.3.3. Número de nivel 88 ó número de nivel de los nombres de condición	128
V-1.2.3.4. Indicadores de nivel de los ficheros	129
V-1.2.4. Reglas de escritura de los números de nivel en la hoja de codificación.	129
V-2. Organización de la Data Division	133
V-2.1. Generalidades	133
V-2.2. Orden y número de secciones de la Data Division en los principales compiladores.	133
V-2.3. Objetivo de cada una de las secciones de la Data Division	134

	<i>Págs.</i>
V-3. Sección de ficheros	137
V-3.1. Generalidades	137
V-3.2. Estructura de la descripción de un fichero	137
V-3.3. Cláusula Recording Mode	138
V-3.4. Cláusula Block Contains	140
V-3.5. Cláusula Record Contains	141
V-3.6. Cláusula Label Record	142
V-3.7. Cláusula Data Record	143
V-3.8. Inscripciones de descripción de registros	143
V-3.8.1. Cláusulas generales de las inscripciones de descripción de registros	144
V-3.8.1.1. Cláusula Redefines o de redefinición de nombre	145
V-3.8.1.2. Cláusula Size o de tamaño	146
V-3.8.1.3. Cláusula Class o de naturaleza del campo	146
V-3.8.1.4. Cláusula Usage o de utilización	147
V-3.8.1.5. Cláusula Signed o indicadora de signo	148
V-3.8.1.6. Cláusula Point o indicadora de punto decimal	149
V-3.8.1.7. Cláusula Picture o modelo	149
V-3.8.1.8. Cláusula Occurs o de repetición de un campo	153
V-3.8.1.9. Cláusula Value o de valor de los nombres de condición	154
V-3.8.2. Cláusulas de edición de las inscripciones de descripción de registros	155
V-3.8.2.1. Cláusula Blank o «espacios»	156
V-3.8.2.2. Cláusula Justified o de alineación	156
V-3.8.2.3. Cláusula Zero o de supresión de ceros no significativos	159
V-3.8.2.4. Cláusula Check o de protección de cheques	160
V-3.8.2.5. Cláusula Float o de moneda	161
V-3.8.2.6. Cláusula Picture o modelo	161
V-3.9. Utilización de la palabra reservada Filler en las inscripciones de descripción de registros	165
V-4. Sección de trabajo o maniobra	169
V-4.1. Generalidades	169
V-4.2. Inscripciones en la sección de trabajo	169
V-4.2.1. Cláusula Synchronized o de sincronización	169
V-4.2.2. Cláusula Value o de valor inicial	171
V-4.2.3. Cláusula Occurs	171
V-4.2.4. Cláusula Redefines o de redefinición	175
V-5. Sección de Constantes	179
V-5.1. Generalidades	179
V-5.2. Inscripciones de constantes independientes	179
V-5.3. Inscripciones de constantes de grupo o agrupadas	179
V-5.4. Valores	180
V-5.5. Utilización de la Constant Section	180

	<i>Págs.</i>
V-6. Sección de almacenamiento común y sección de unión	183
V-6.1. Generalidades	183
V-7. Sección de informes	187
V-7.1. Generalidades sobre impresión de informes automáticos	187
V-7.2. Estructura de la Report Section	188
V-7.3. Estructura del informe	189
V-7.4. Inscripciones de descripción de informes	189
V-7.4.1. Cláusula Code	189
V-7.4.2. Cláusula Control	189
V-7.4.3. Cláusula Page	190
V-7.5. Inscripciones de grupo de informe y de elemento de informe	191
V-7.5.1. Cláusula Type	192
V-7.5.2. Cláusula Line	193
V-7.5.3. Cláusula Next Group	194
V-7.5.4. Cláusula Column	194
V-7.5.5. Cláusula Group Indicate	194
V-7.5.6. Cláusula Sourcce	195
V-7.5.7. Cláusula Sum	195
V-7.6. Contadores generados por la rutina de impresión de informes	196
 CAPITULO VI.—LAS DIVISIONES DE UN PROGRAMA. PROCEDURE DIVISION	 197
VI-1. Generalidades de la Procedure Division	203
VI-1.1. Objeto de la Division	203
VI-1.2. Sintaxis de la Procedure Division	203
VI-1.3. Verbos integrantes de la Procedure Division	203
VI-2. Verbos de entrada-salida	207
VI-2.1. Generalidades	207
VI-2.2. Verbos que tratan información organizada en ficheros	208
VI-2.2.1. Verbo Accept	208
VI-2.2.2. Verbo Display	209
VI-2.3. Verbos que tratan información en ficheros	211
VI-2.3.1. Verbo Open	211
VI-2.3.2. Verbo Read	213
VI-2.3.3. Verbo Write	216
VI-2.3.4. Verbo Close	219

	<i>Págs.</i>
VI-2.4. Otros verbos de entrada-salida	221
VI-2.4.1. Verbo Rewrite	221
VI-2.4.2. Verbo Delete	221
VI-2.4.3. Verbo Seek	221
VI-3. Verbos de manipulación y transferencia de datos	225
VI-3.1. Generalidades sobre los verbos de manipulación de datos	225
VI-3.2. Verbo Move o de transferencia de datos	225
VI-3.3. Verbos de manipulación de datos	231
VI-3.3.1. Verbo Examine o de manipulación simple	231
VI-3.3.2. Verbo Transform o de manipulación múltiple	233
VI-4. Verbos aritméticos	241
VI-4.1. Generalidades	241
VI-4.2. Particularidades del uso de los verbos aritméticos	241
VI-4.3. Opciones comunes a todos los verbos aritméticos	243
VI-4.3.1. Opción Giving	243
VI-4.3.2. Opción Rounded	243
VI-4.3.3. Opción On Size Error	244
VI-4.4. Estudio particular de cada uno de los verbos aritméticos... ..	244
VI-4.4.1. Verbo Add	244
VI-4.4.2. Verbo Subtract	245
VI-4.4.3. Verbo Multiply	247
VI-4.4.4. Verbo Divide	248
VI-4.4.5. Verbo Compute	248
VI-5. Verbos de control de secuencia	255
VI-5.1. Generalidades de los verbos de control de secuencia	255
VI-5.2. Verbo de decisión o comparación	256
VI-5.2.1. Pseudo-verbo IF o declaración IF	256
VI-5.3. Verbos de bifurcación	259
VI-5.3.1. Verbo GO TO	259
VI-5.3.2. Verbo ALTER	261
VI-5.3.3. Verbo PERFORM	263
VI-5.3.4. Jerarquización en el uso del verbo PERFORM	274
VI-5.4. Verbos complementarios	277
VI-5.4.1. Verbo EXIT	277
VI-5.4.2. Verbo STOP	278
VI-6. Verbos del compilador	281
VI-6.1. Generalidades	281
VI-6.2. Verbo ENTER	281
VI-6.3. Verbo NOTE	281

	<i>Págs.</i>
VI-7. Verbo de control de entrada-salida	285
VI-7.1. Generalidades	285
VI-7.2. Verbo USE	285
CAPITULO VII.—DEPURACION DE UN PROGRAMA COBOL	289
VII-1. Mensajes del compilador	295
VII-2. Lenguaje depurador	299
VII-2.1. Función del verbo Trace	299
VII-2.2. Función del verbo Exhibit	300
VII-2.2.1. Opción Named	300
VII-2.2.1.1. Ejemplo de uso de la opción Exhibit Named	301
VII-2.2.2. Opción Changed	301
VII-2.2.3. Opción Changed Named	302
VII-2.3. Función del verbo On	302
ANEXO A.—RELACION DE PALABRAS RESERVADAS COBOL	305
ANEXO B.—SECUENCIAS DE CLASIFICACION	315
ANEXO C.—OPCIONES DE LA CLAUSULA USAGE Y REPRESENTACION INTERNA DE LOS DATOS	319
ANEXO D.—TABLAS RESUMEN DE COMPARACIONES, MOVIMIENTOS Y SERIES DE CARACTERES PERMISIBLES	323
APENDICE A.—ENLACE ENTRE PROGRAMAS	329
APENDICE B.—CLASIFICACION INCLUIDA EN UN PROGRAMA COBOL	337
APENDICE C.—ANSI COBOL	343
APENDICE D.—BIBLIOTECA DE PROGRAMAS FUENTE	349
APENDICE E.—MENSAJES DE ERROR DE UN COMPILADOR COBOL	353

PROLOGO DEL AUTOR

Acometer la escritura de cualquier libro técnico siempre supone para el autor un esfuerzo considerable. A los problemas intrínsecamente técnicos se unen las propias dificultades de la clara exposición, la infinidad de horas empleadas en las correcciones de las galeras de imprenta y en la de las pruebas definitivas. Todo esto se supera con la ilusión de la creatividad, del alumbramiento de algo nuevo.

En esta obra no he podido crear nada nuevo. En COBOL está todo o casi todo escrito. Existen infinidad de manuales de COBOL creados por los departamentos técnicos de los más importantes fabricantes de ordenadores, en los que se exponen en forma más o menos detallada cada uno de los elementos componentes del lenguaje.

Sin embargo, todos los manuales proporcionados por los fabricantes de ordenadores adolecen, desgraciadamente, de tener un carácter didáctico, por lo que su lectura resulta normalmente tediosa y su empleo es más apropiado para consultas del conocedor del lenguaje que para los que se inician en él.

El espíritu que me ha animado y dirigido constantemente durante la redacción de este libro ha sido el de proporcionar al lector que se inicia en el estudio del lenguaje, posiblemente alumno mío en el Instituto de Informática, una obra que combine equilibradamente las necesidades de un manual de referencia y los de un libro de texto maximizando las ventajas de ambos y minimizando sus desventajas.

Siempre que la explicación de un concepto, por su idiosincrasia, necesitaba una aclaración adicional, independiente del propio lenguaje de programación, la he incluido en el lugar más adecuado, sin entorpecer la propia definición del concepto.

- Cuando la variedad de opciones de una determinada instrucción hacía peligrar la comprensión de la misma, he procurado sinoptizarlas a través de un esquema, o de un diagrama o de un cuadro resumen; en esencia, en una expresión gráfica que apoyase a las palabras.*
- En determinadas instrucciones y en todos aquellos aspectos del lenguaje que rebasan los límites de conceptos básicos y entran de lleno en el terreno del superespecialista y en aquellos matices particulares cuyo conocimiento sólo tiene valor en casos muy específicos de problemas también específicos, he optado por crear separatas del texto en forma de apéndices.*
- No he ahorrado esfuerzos en exponer ejemplos en todos los epígrafes del texto en que aquéllos ayudan, aunque sea en lo más mínimo, a una mejor comprensión.*
- Al final de cada uno de los capítulos de la obra, sin excepción, he incluido ejemplos de programas reales, parciales o completos, de los que he comentado los conceptos estudiados en el capítulo.*
- En la estructura de la obra he procurado atenerme en lo posible a la forma habitual de exponer un texto de estudio de cualquier idioma hablado aunque el COBOL sea un lenguaje escrito para una máquina.*

Aparte de una breve introducción histórica, he descompuesto el estudio del lenguaje en siete partes.

En el capítulo I se realiza un estudio morfológico del lenguaje, es decir, de los elementos componentes del mismo.

En el capítulo II se estudian los métodos de unión de los elementos para formar frases del lenguaje, es decir, el estudio sintáctico del mismo.

El capítulo III se dedica a exponer la estructura de un programa y la forma en que el programador puede plasmar su programa a través de un formulario de codificación.

En los capítulos IV a VI se efectúa el estudio detallado de las cuatro divisiones componentes de cualquier programa COBOL.

El último capítulo se dedica al estudio de los verbos de lenguaje que permiten al programador autocorregirse en los errores de escritura que haya podido cometer en la redacción del programa.

No me queda sino agradecer el interés mostrado y la confianza depositada en mí por el director del Instituto de Informática, don Angel Regidor Sendín, al adoptar este libro como texto en el Instituto.

T. HURTADO MERELO



INTRODUCCION

0.1. BREVE RESEÑA HISTORICA

El lenguaje COBOL nació de la colaboración conjunta de los fabricantes de ordenadores y de los usuarios, en cooperación con el Departamento de Defensa de los Estados Unidos. Este lenguaje se describió por primera vez en un informe dirigido a la Conferencia sobre Lenguajes de Sistemas de Proceso de Datos (CODASYL: Conference of Date Systems Languages), publicado por el Departamento de Imprenta del Gobierno de los Estados Unidos en abril de 1960. Perfeccionamientos introducidos posteriormente han hecho evolucionar este lenguaje hasta su forma actual.

Un extracto de dicho informe se reproduce a continuación:

“Toda aquella empresa u organismo al que interese reproducir en su totalidad o por partes el informe y las especificaciones COBOL iniciales, haciendo uso de ideas extraídas de este informe o utilizándolo como base para la creación de un manual de instrucción, o con cualquier otro fin, queda expresamente autorizada para hacerlo así. No obstante, se requiere a las referidas empresas u organismos para que reproduzcan en el prólogo de su manual el contenido de esta sección. Sin embargo, aquellas que solamente utilicen un breve pasaje de dichas especificaciones, como en el caso de una reseña bibliográfica, sólo estarán obligadas a mencionar la palabra COBOL como reconocimiento e identificación de su origen, pero sin que sea preciso reproducir esta sección en su totalidad.»

«COBOL es un lenguaje industrial y no es propiedad de una sociedad o de un grupo de sociedades, ni de un organismo o grupo de organismos.»

«Ni el comité COBOL ni ninguno de los que han aportado su contribución garantizan explícita o implícitamente la exactitud y el buen funcionamiento del sistema de programación o del lenguaje. Asimismo, no asumen ninguna responsabilidad a este respecto.»

«Se han establecido normas que regulan el mantenimiento y la actualización del COBOL. Las consultas relativas a los procedimientos y métodos que han de seguirse para proponer modificaciones deberán dirigirse al Comité Ejecutivo sobre Lenguaje de Sistemas de Proceso de Datos.»

«Los autores y poseedores de la propiedad literaria del material que se utiliza en el presente manual: *Flow-Matic* (marca registrada de la Sperry Rand Corporation). *Programa for the Univac (R) I and II*, Data Automation System, publicado en 1958 y 1959 por Sperry Rand Corporation; *IBM Commercial Translator*, Form N° F 28/3, publicado en 1959 por IBM; *Fact*, DSI 27 a 5260, publicado en 1960 por Minneapolis Honeywell, han autorizado expresamente la utilización integral o parcial de las especificaciones del COBOL. Esta autorización se extiende igualmente a la reproducción y utilización de las especificaciones COBOL en los manuales de programación y obras similares.»

0.2. INTRODUCCION A LA PROGRAMACION EN COBOL

El lenguaje de programación COBOL está orientado hacia la resolución de problemas de gestión. Precisamente COBOL son las siglas correspondientes a las iniciales de las palabras inglesas Common Business Oriented Language, cuya traducción literaria es “Lenguaje Comparti-

do Orientado a los problemas Comerciales». Su propia evolución y la de las empresas, ha ampliado su campo de acción y su aplicación se ha extendido desde la simple utilización para resolver cualquier problema de facturación, hasta cualquier tipo de problema de gestión de una empresa (gestión de personal, gestión de aprovisionamientos, etc.).

La característica fundamental de los problemas de gestión radica en el tratamiento de grandes volúmenes de datos a los que se les aplican operaciones aritméticas, generalmente sencillas, para obtener los resultados deseados.

En contraposición, los problemas matemático-científicos se caracterizan por tratar poco volumen de datos que deben ser sometidos a complejos cálculos.

Es obvio decir que existen problemas mixtos o intermedios. Ni bajo ni elevado número de datos. Ni sencillos ni complicados cálculos.

El COBOL está orientado fundamentalmente al primero y tercero de los tres tipos de problemas expuestos, y puede resolver algunos tipos particulares del segundo, aunque existen otros lenguajes más adecuados para estos fines, como es el FORTRAN.

COBOL es un lenguaje de ordenador que es común a la mayoría de los ordenadores. Es decir, la mayoría de los fabricantes de ordenadores han diseñado sus máquinas para aceptar un compilador de COBOL de forma tal que un mismo programa COBOL puede ser compilado en un ordenador IBM sistema 370 o en un ordenador Univac modelo 1108, o en un ordenador Siemens 4004, o en un CDC serie 6000, etc., con algunas pequeñas variaciones de uno a otro.

La universalidad del COBOL permite, por consiguiente, a los usuarios de ordenadores una gran flexibilidad.

El COBOL es un lenguaje similar al inglés. Todas sus instrucciones se codifican utilizando palabras inglesas en lugar de códigos más o menos complejos. Para sumar los números, por ejemplo, se utiliza la palabra ADD. También las reglas de programación en COBOL son análogas a muchas de las reglas de escritura del inglés, lo que hace que sea un lenguaje de muy fácil aprendizaje.

Por otra parte, en la prueba y puesta a punto de los programas en COBOL se requiere mucho menos tiempo que en la de los programas escritos en otros lenguajes.

La semejanza del COBOL con el inglés, aparte de facilitar su escritura, hace muy fácil su lectura, de forma tal que algunos programas pueden ser fácilmente comprendidos por personas profanas en proceso de datos.


Para definir las informaciones propias a su problema, el programador de COBOL debe utilizar palabras que él mismo crea, respetando determinadas reglas. En la elección de estas palabras la imaginación del programador juega un importante papel. Además, entre todas las órdenes COBOL puestas a disposición del programador, éste puede utilizar con más frecuencia algunas que otras que también podrían ser convenientes. Puede agrupar las instrucciones en frases más o menos largas y éstas en párrafos, al mismo tiempo más o menos extensos.

En resumen, en COBOL se puede, como ocurre con cualquier lenguaje, expresar una misma idea o resolver un mismo problema en formas muy diferentes, dependiendo del *estilo* gramatical utilizado por el programador.


CAPITULO I

GRAMÁTICA DEL LENGUAJE:

MORFOLOGIA



CONTENIDO

- I-1. CARACTERES
 - I-2. PALABRAS FORMADAS
POR EL PROGRAMADOR
 - I-3. PALABRAS ESPECIFICAS
DEL LENGUAJE
- 

I - 1

CARACTERES

I-1. CARACTERES UTILIZADOS EN EL COBOL

El lenguaje COBOL, como cualquier lenguaje escrito, se expresa a partir de unidades elementales que constituyen un juego de caracteres, compuesto por letras, números y signos especiales. Ordenando y componiendo en forma adecuada dichos caracteres se forman palabras. Estas se combinan formando expresiones u oraciones, etc. En definitiva, no existe, en principio, ninguna diferencia fundamental entre la forma de expresarse en COBOL y la de expresarse en cualquier otro idioma, especialmente en inglés, con el cual tiene muchas analogías.

I-1.1. TIPOS DE CARACTERES

Los caracteres que pueden utilizarse en el lenguaje se clasifican en dos categorías:

- Caracteres básicos.
- Caracteres adoptados.

I-1.1.1. Caracteres básicos

Para el COBOL, al ser concebido por usuarios de ordenadores de procedencia anglosajona, se adoptaron los mismos caracteres que en su lengua vernácula, es decir, las 26 letras del alfabeto anglosajón (A a Z), con la particularidad de utilizar exclusivamente letras mayúsculas; pero al haber nacido como un lenguaje orientado para comunicarse con una máquina, para resolver problemas de tipo comercial, necesitaba emplear caracteres adecuados a esta finalidad, como son las diez cifras decimales y los signos aritméticos, y además, una serie de caracteres necesarios en la formación de sus expresiones.

En definitiva, el conjunto de caracteres denominados básicos lo forman:

- Letras: A a Z (26 caracteres).
- Dígitos: Ø (1) a 9 (10 caracteres).
- Caracteres especiales: + - * / = < > () ; , . ' \$ ¢ (2).

I-1.1.1.1. Utilización de los caracteres básicos

Los 51 caracteres básicos anteriormente mencionados están expuestos en el cuadro de la página siguiente y se utilizan fundamentalmente para los casos expuestos en dicho cuadro.

(1) Con el fin de evitar confusiones en el texto de los programas, se representa el dígito 0 como Ø para diferenciarlo de la letra O.

(2) ¢ Este signo es un convenio para representar la ausencia de carácter, o el espacio, que es un carácter más de COBOL.

(N. A.) Los caracteres <> y ; no se emplean en todos los compiladores, aunque tienen sus respectivos equivalentes.

CARACTERES	UTILIZACION BASICA
A a Z	En la formación de palabras.
0 a 9	Para formar números y también en la formación de palabras.
+ *	En operaciones aritméticas y en la edición de resultados.
—	En operaciones aritméticas, en edición de resultados y en la formación de nombres.
/	En operaciones aritméticas.
= < >	En expresiones aritméticas y lógicas.
' (comilla)	Para encerrar constantes de tipo no numérico.
(símbolo ficticio para indicar un espacio o blanco)	En puntuación (en la separación de palabras).
, ;	En puntuación (en la separación de palabras y oraciones).
.	En puntuación (en la formación de frases y oraciones) y como punto decimal en números decimales (utilización normalizada en los países anglosajones equivalente a la coma decimal).
\$	En edición de resultados (se transforma en los usuarios que no tienen este signo de moneda).
()	En expresiones aritméticas, lógicas y en puntuación.

I-1.1.2. Caracteres adoptados

Además de los caracteres comúnmente empleados en cualquier programa COBOL existen otros que son adoptados particularmente en determinados compiladores. La adopción de estos caracteres está condicionada, normalmente, a la adecuación del lenguaje a problemas específicos y particulares de cada instalación. Sin embargo, estos caracteres, denominados adoptados o no básicos, no forman parte integrante de la morfología específica del lenguaje y por consiguiente no ejercen función determinada, sino únicamente cumplen con la misión específica para la que han sido adoptados. Por ejemplo, es corriente emplear el signo £ (moneda inglesa) en usuarios de Gran Bretaña y la letra Ñ en los usuarios hispano-parlantes.

Con carácter general, el COBOL admite en determinados elementos (3) la utilización de cualquier carácter necesario para los fines específicos de cualquier usuario en particular, con tal que el ordenador que vaya a tratar el programa establezca entre las 205 configuraciones de BYTES no utilizadas (4), la correspondiente al carácter adoptado.

(3) En las constantes no numéricas, que se estudian en el capítulo II.

(4) La mayoría de los ordenadores tienen como unidad de almacenamiento el BYTE u octeto, formado por 8 bits (número binario de 8 posiciones). Por consiguiente se pueden establecer 256 posibilidades de representación, de las cuales 51 están reservadas para los caracteres básicos.

I - 2

PALABRAS FORMADAS
POR EL PROGRAMADOR

I-2. PALABRAS FORMADAS POR EL PROGRAMADOR

La unión de varios caracteres, incluso un único carácter, forman palabras. Todas las posibles combinaciones, sometidas o condicionadas por reglas específicas que se verán más adelante, después de estudiar los distintos tipos de nombres, dan lugar a palabras, que se clasifican en COBOL en dos tipos:

- **Palabras formadas por el programador**
(nombres y constantes).
- **Palabras específicas del lenguaje**
(verbos, operadores y palabras reservadas).

I-2.1. NOMBRES

Un programa, mediante las instrucciones correspondientes, transforma los datos de un problema en resultados.

Para poderse referir tanto a los datos como a los resultados, en los lenguajes de alto nivel, como es el COBOL, no se hace referencia directamente a la posición ocupada por los mismos en la memoria (direcciones de memoria), sino a *nombres* simbólicos atribuidos por el programador a los mismos.

I-2.1.1. Tipos de nombres

Los diferentes tipos de nombres que pueden ser utilizados por un programador en el desarrollo de un programa son:

- Nombres de datos.
- Nombres de condición.
- Nombres de párrafo.
- Nombres de fichero.
- Nombres de dispositivo.
- Nombres de biblioteca.
- Nombres de calificador o calificadores
- Nombres de índice o subíndices.

Cada uno de estos nombres los crea el programador para denominar simbólicamente cualquiera de los componentes, tanto físicos como lógicos, que debe manejar en los programas (datos, ficheros, dispositivos físicos, etc.).

I-2.1.1.1. Nombres de dato

Son los nombres utilizados en un programa para representar las distintas informaciones tratadas en el mismo (datos, resultados, variables, etc.). Por ejemplo, PRECIO puede ser el nombre

de dato utilizado para representar los distintos valores que tome un dato relativo al precio de un determinado artículo.

1-2.1.1.2. *Nombres de condición*

Un nombre de condición es el nombre de dato asignado a uno o varios valores de entre los de la gama que puede tomar el mismo. Por consiguiente, debe estar siempre asociado a un nombre de dato.

Supongamos que en un programa se tiene necesidad de procesar datos relativos a determinados materiales y que una de las características de éstos es el color. Normalmente este dato vendría codificado. El conjunto de códigos de colores puede representarse por un nombre de datos; por ejemplo, COLOR. Si existen cinco valores de este nombre de dato, 1 a 5, se puede asignar a cada uno de los mismos un nombre.

- Al valor 1, el nombre BLANCO.
- Al valor 2, el nombre AZUL.
- Al valor 3, el nombre VERDE.
- Al valor 4, el nombre AMARILLO.
- Al valor 5, el nombre ROJO.

De esta forma, al tratarse el dato COLOR en el programa, se puede hacer uso, bien del código particular, bien del nombre dado al código.

1-2.1.1.3. *Nombres de párrafo*

Un programa está constituido por una serie detallada de instrucciones. Se agrupan para solucionar aspectos parciales del problema que resuelve el programa. El número de veces que cada grupo de instrucciones se ejecuta durante el desarrollo de un programa depende de muchas circunstancias, tales como la naturaleza de las instrucciones, valor de un determinado dato, etc., de forma tal que cumpliéndose o no determinadas condiciones impuestas a los mismos se provoca que la repetición de un grupo de instrucciones más frecuentemente que la de otros.

Este grupo de instrucciones forman, como se verá en el próximo capítulo, un párrafo del programa, conociéndose cada uno de los grupos, como nombres de párrafo o nombres de procedimiento.

1-2.1.1.4. *Nombres de fichero*

Son los nombres simbólicos dados a los soportes físicos que albergan los datos que van a ser tratados en el programa.

En un programa que fuese a procesar datos referidos a las ventas realizadas por una empresa a sus clientes, para efectuar una facturación, sería muy normal dar al fichero que contiene dichos datos el nombre FICHERO-VENTAS. En cualquier parte del programa en el que se haga referencia a dicho fichero se empleará la palabra FICHERO-VENTAS para poder tratar los datos contenidos en él.

1-2.1.1.5. *Nombres de dispositivo*

Son los nombres utilizados para identificar un determinado equipo periférico (disco magnético, lectora de fichas, etc.). Sirven de unión entre el programa y el sistema.

I-2.1.1.6. Nombres de biblioteca

Frecuentemente, en un programa es necesario escribir determinadas secuencias de instrucciones idénticas a las escritas en otro programa. A fin de evitar al programador la reescritura de estas secuencias es posible incluirlas en un fichero especial del sistema que se denomina LIBRERIA e insertarlas en el nuevo programa a partir de dicho fichero. Para poderlo realizar basta con haber asignado a cada una de las secuencias que se vayan a insertar un nombre simbólico que se denomina nombre de biblioteca.

Por consiguiente, un nombre de biblioteca es el nombre de una de las secuencias del fichero LIBRERIA, que se puede insertar en un programa cualquiera llamándolas en este programa por su nombre.

I-2.1.1.7. Nombres de calificador o calificadores

Es muy frecuente que en un determinado programa se utilicen datos, de naturaleza muy semejante, a los que normalmente se les atribuye el mismo nombre. Por ejemplo, puede ocurrir que procesando datos relativos a dos ficheros distintos de materiales, la característica COLOR exista en registros de ambos ficheros. Lo normal sería asignar el nombre COLOR a los códigos o valores de esta característica en los datos de ambos ficheros. Sin embargo, existiría una ambigüedad al referirse a este dato en el programa, por el nombre genérico COLOR, ya que no se especifica a qué tipo de material se refiere. Cuando se presentan circunstancias de este tipo es necesario determinar, calificar, a dicho nombre añadiéndole otro elemento que destruya la ambigüedad. Estos elementos se conocen con el nombre de *calificadores*.

I-2.1.1.8. Nombres de índice o subíndices

Cuando algunos de los datos que intervienen en un problema están organizados en forma de tabla, en lugar de tener que asignar un nombre específico a cada uno de ellos, se les puede reagrupar bajo un mismo nombre genérico.

Un programa que trate los habitantes de las provincias españolas necesitaría la utilización de cincuenta nombres diferentes para poderse referir específicamente a cada una de ellas. Por ejemplo: HABITANTES-ALAVA, HABITANTES-ALBACETE, HABITANTES-MADRID, HABITANTES-ZARAGOZA.

Es evidente que resultará más sencillo utilizar un único nombre genérico, por ejemplo: HABITANTES-PROVINCIAS, con tal que se pueda añadir a este nombre un *indicador* o índice de la provincia a la que pueda referirse en un caso específico. Si se ordenan alfabéticamente las provincias y se les asigna a cada una un índice numérico ordinal, se podría crear una tabla como la siguiente:

PROVINCIAS	INDICE
ALAVA	01
ALBACETE	02
ALICANTE	03
.....
MADRID	27
.....
ZARAGOZA	50

De esta forma, al referirnos a los habitantes de la provincia de Madrid, bastará con indicar el nombre genérico HABITANTES-PROVINCIAS (I), con tal que I valga 27 en el momento de hacer la referencia.

Por consiguiente, un nombre índice es un nombre de dato seguido de uno o varios valores numéricos o de otros nombres de datos, denominados índices.

I-2.1.2. Reglas para la formación de los distintos tipos de nombres

La formación de cualquiera de los distintos tipos de nombres anteriormente expuestos está condicionada a reglas, basadas en la utilización de determinado número de los caracteres básicos del COBOL.

REGLA COMÚN EN LA FORMACIÓN DE TODOS LOS NOMBRES

Se pueden utilizar los siguientes caracteres:

- Las 10 cifras.
- Las 26 letras mayúsculas.
- El trazo de unión o guión (-).

Formación de nombres de datos, nombres de condición, nombres de fichero y nombres de biblioteca

Todos se rigen por las mismas reglas, que además de la común, son:

- No se pueden utilizar más de *treinta* caracteres (de 1 a 30).
- El carácter - no puede utilizarse al principio ni al final del nombre formado, ni existir dos o más caracteres - consecutivos.
- Debe existir, como mínimo, un carácter alfabético.

Formación de nombres de párrafo

- Se utilizan las mismas reglas que en la utilización de los nombres de datos, con la *excepción* de la limitación de utilizar como mínimo un carácter alfabético. Es decir, un nombre de párrafo puede estar formado exclusivamente por números.

Formación de nombres de dispositivo

Las reglas de formación varían con cada uno de los distintos compiladores. En algunos compiladores existen nombres ya preestablecidos para designar determinados dispositivos.

En la figura I-1 se indican las reglas particulares para cada uno de los sistemas figurados, así como los nombres preestablecidos para cada uno de los diferentes dispositivos de entrada y salida.

Formación de nombres de calificador

Los nombres de calificador se forman como los nombres de dato. Deben ir unidos al nombre de dato al cual califican. Entre el nombre de calificador y el nombre de dato se debe interponer, separados por un espacio como mínimo, una de las palabras OF o IN.

REGLAS DE FORMACION DE LOS NOMBRES DE DISPOSITIVO


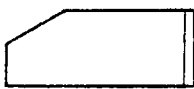
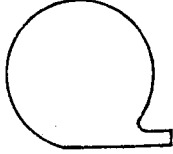
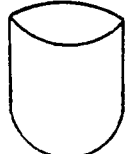

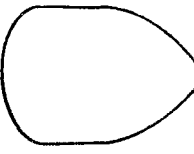
TIPO DE DISPOSITIVO	CDC Serie 6000	IBM Series 360 y 370	UNIVAC Serie 1108	UNIDATA-CII Serie Iris 50	UNIDATA-HWB Serie GE 600	UNIDATA-SIEMENS Serie 4004	UNIVAC Series 9200-9300	UNIDATA-HWB Serie 100
CUALQUIER DISPOSITIVO	1 a 7 caracteres (el 1.º ha de ser alfabético)	1 a 8 caracteres (el 1.º ha de ser alfabético y deben escribirse entre '')		1 carácter alfabético	2 caracteres (el 1.º ha de ser alfabético)	1 a 8 caracteres (el 1.º ha de ser alfabético y deben escribirse entre '')		
 LECTORA FICHAS	INPUT	SYSIN	CARD-READER-EIGHTY	SYSIN	2 caracteres (el 1.º ha de ser alfabético) + FOR CARDS	SYSIN o SYSIPT	CARD-READER	CR
 PERFORADORA FICHAS	PUNCH o PUNCHB	SYSPUNCH	CARD-PUNCH-EIGHTY	SYSPUNCH	2 caracteres (el 1.º ha de ser alfabético)	SYSPUNCH o SYSOPT	CARD-PUNCH	CP
 UNIDAD DE CINTA MAGNETICA	1 a 7 caracteres (el 1.º ha de ser alfabético)	1 a 8 caracteres (el 1.º ha de ser alfabético)	UNISERVO	1 carácter alfabético	Idem	1 a 8 caracteres (el 1.º ha de ser alfabético)	TAPE	MT-n 0 < n < 9
 UNIDAD DE DISCOS MAGNET.	Idem	Idem	MASS-STORAGE	Idem	Idem	Idem	DISC	DS-n 0 < n < 11
 IMPRESORA	OUTPUT	SYSOUT	PRINTER	SYSOUT	2 caracteres (el 1.º ha de ser alfabético) + FOR LISTING	SYSOUT o SYSLST	PRINTER o BUFFER-PRINT	PR
 CONSOLA	CONSOLE	CONSOLE	CONSOLE	CONSOLE	CONSOLE	CONSOLE	CONSOLE	CONSOLE

FIGURA I-1

Formación de nombres de índices o índices

Los nombres de índice utilizan las mismas reglas de formación que los nombres de datos, pues son nombres de datos. El índice que los acompaña debe escribirse seguido del nombre de dato, separado de él por un espacio y encerrado entre paréntesis. El índice puede ser un nombre de dato o un valor entero; si existe más de un índice debe escribirse una coma y un espacio, en este orden, entre cada dos índices.

I-2.1.3. Ejemplos de distintos tipos de nombres

Como resumen de los epígrafes anteriores, se exponen en la figura I-2 diversos ejemplos de los distintos tipos de nombres estudiados.

I-2.2. CONSTANTES

El otro tipo de palabras que puede formar el programador son las constantes. Estas se utilizan para representar, a diferencia de los nombres de datos, informaciones que no cambian de valor en el curso del desarrollo de un programa.

I-2.2.1. Tipos de constantes

En COBOL se emplean tres tipos de constantes:

- Constantes literales.
- Constantes con nombre.
- Constantes figurativas.

I-2.2.1.1. Constantes literales

Las constantes literales, o solamente literales, son aquellas que se crean en el lugar y momento del programa en que surge la necesidad de utilizarlas. A diferencia de los nombres de datos no representan en forma simbólica a un conjunto de valores, sino que representan en forma real a un único valor, que es el figurado como constante.

Existen dos tipos de literales:

- Literales numéricos.
- Literales no-numéricos o alfanuméricos.

a) LITERALES NUMÉRICOS

Un literal numérico es la representación en un programa *de un valor numérico* que permanece invariable durante la ejecución del programa.

b) LITERALES NO NUMÉRICOS

Un literal no numérico es una serie de caracteres cualesquiera que se representa como tal cadena de caracteres en determinados momentos de un programa.

Formación de literales

Los literales los crea el programador en los momentos de un programa que son necesarios, utilizando las siguientes reglas:

a) PARA LITERALES NUMÉRICOS

- Se pueden emplear los diez dígitos, el signo +, el signo — y el signo • (punto decimal).
- Cuando se utilice el signo + o el signo — deben ser el primer carácter del literal.

CONJUNTO DE CARACTERES	PUEDE SER							
	Nombre de dato	Nombre de condición	Nombre de párrafo	Nombre de fichero	Nombre de dispositivo	Nombre de biblioteca	Nombre de dato calificado	Nombre de índice
COLOR	Sí	Sí	Sí	Sí	Sí*	Sí	No	No
AZUL	Sí	Sí	Sí	Sí	Sí*	Sí	No	No
MATRICULA	Sí	Sí	Sí	Sí	No	Sí	No	No
H023	Sí	Sí	Sí	Sí	Sí*	Sí	No	No
12345	No	No	Sí	No	No	No	No	No
FECHA OF CALCULO	No	No	No	No	No	No	Sí	No
—H234	No	No	No	No	No	No	No	No
A123—	No	No	No	No	No	No	No	No
POBLACION (2)	No	No	No	No	No	No	No	Sí
POBLACION (1)	No	No	No	No	No	No	No	Sí
'H3234'	No	No	No	No	Sí*	No	No	No
PRODUCCION (I, K)	No	No	No	No	No	No	No	Sí
H27*3A	No	No	No	No	No	No	No	No
—HAB—	No	No	No	No	No	No	No	No
VALOR IN FICH-A	No	No	No	No	No	No	Sí	No
HB-1-2A	Sí	Sí	Sí	Sí	Sí*	Sí	No	No
I2-H32	Sí	Sí	Sí	Sí	No	Sí	No	No

(*) Sólo para algunos compiladores.

FIGURA I-2

- El signo + confiere valor positivo al valor real representado por el literal.
- El signo - confiere valor negativo al valor real representado por el literal.
- El signo • no puede figurar en el último lugar del literal.
- El signo • confiere valor decimal al valor representado por el literal.
- No pueden utilizarse más de 18 caracteres (excluidos los signos + o -).

b) PARA LITERALES NO NUMÉRICOS

- Se puede utilizar cualquiera de los caracteres básicos o adoptados del COBOL, excepto las ' (comillas).
- Se puede emplear como máximo 120 caracteres (1) en el total del literal.
- Los literales no numéricos deben ser escritos entre comillas, sin que las comillas formen parte del literal ni, por consiguiente, contabilicen en la limitación de 120 caracteres.

Observación: Si un literal está formado exclusivamente por dígitos, y encerrado entre comillas, se considera como literal no-numérico.

c) EJEMPLOS DE LITERALES

LITERALES NUMERICOS		
CORRECTOS	INCORRECTOS	CAUSA
1234 +1234 -1234 -12.34 +12.58 327.35 .371	351.	Por emplearse el punto como último carácter.
	-ø123	Por utilizarse un espacio.
	347+	Por figurar el signo + en última posición.

LITERALES NO NUMERICOS		
CORRECTOS	INCORRECTOS	CAUSA
'20 OCTUBRE 1974' 'ENERO' 'CALCULO DE RESULTADOS'	HORAS-EXTRAS 'CALCULO-'HORAS'	Falta de comillas. Empleo de una comilla entre los caracteres utilizados.
'ABC-Hø ÑO35'	'ABC-H35	Falta de una comilla (la última).
'1974'	ABC-H35'	Falta de una comilla (la primera).

I-2.2.1.2. *Constantes con nombre*

Se ha definido una constante como un valor fijo que introduce el programador en el momento necesario del programa y que permanece invariable durante la ejecución del mismo.

Cuando el programador decide utilizar constantes literales es porque tiene seguridad de que el valor de las constantes va a permanecer invariable durante un período largo de validez de un programa sin que sea necesario proceder a su modificación.

(1) En algunos compiladores el número de caracteres máximos admitidos puede diferir de los 120 indicados.

Cuando surja la necesidad de modificar los valores constantes introducidos será necesario actualizar el valor de las constantes en todos aquellos puntos del programa en que han sido figurados, con el consiguiente peligro de olvidar u omitir por descuido la actualización de alguna de ellas.

En estas circunstancias se utilizan constantes con nombre, que son realmente nombres de dato a los que se atribuyen valores únicos que no deberán cambiar en el curso de la ejecución del programa.

EJEMPLO DE UTILIZACIÓN DE UNA CONSTANTE CON NOMBRE

Supongamos que en un programa de cálculo de nóminas se utiliza con mucha frecuencia el valor del 12 por 100 para calcular el importe de los descuentos por rendimiento de trabajo personal. Cada vez que sea necesario calcular este importe se empleará la constante 12 para multiplicarla por el salario bruto para calcular el importe de dicho impuesto. Si se decide cambiar la tasa del impuesto será necesario corregir todas las instrucciones del programa que contenga dicha constante y reemplazarla por el nuevo valor.

Si en lugar de utilizar un literal numérico se emplea una constante con nombre, por ejemplo IRTP, solamente será necesario modificar en el lugar del programa donde se define el valor de dicha constante con nombre, por el nuevo valor que deba ser asignado.

I-2.2.1.3. Constantes figurativas

Las constantes figurativas son constantes creadas en el COBOL con nombre y valor específicamente predeterminados.

Por consiguiente, no son creadas por el programador pero sí utilizadas con los nombres y valores que se establecen en el cuadro siguiente:

Nombre de la constante	Valor asignado	Empleo o utilización
ZERO o ZEROS o ZEROES	Valor numérico = 0 Valor alfanumérico = '0'	Se utiliza como literal numérico con valor 0 en operaciones aritméticas. Es equivalente a una serie de caracteres iguales a cero, como literal no numérico.
SPACE o SPACES	Valor alfanumérico = ' '	Es equivalente a una serie de caracteres iguales a ' ', como literal no numérico.
QUOTE o QUOTES	Valor alfanumérico = ' (comillas)	Es equivalente a una serie de caracteres iguales a ' ', como literal no numérico.
ALL	Cualquiera	Esta constante permite generar una serie de caracteres cualesquiera, iguales a los caracteres no numéricos a los que se antepone.
HIGH-VALUE o HIGH-VALUES	El mayor valor	Estas constantes representan una serie de caracteres, no reales, pero representativos, respectivamente, de la mayor y menor jerarquía entre los del juego de caracteres COBOL (1).
LOW-VALUE o LOW-VALUES	El menor valor	

(1) Entre todos los caracteres admitidos por el COBOL, existe una jerarquía de valor de forma tal que se puede establecer una secuencia de clasificación entre ellos. Esta jerarquía se establece en el caso de almacenamiento en octetos, en valores binarios formados por 8 bits. Estas dos constantes, por consiguiente, están representadas en binario por 11111111 y 00000000, respectivamente.

I-3. PALABRAS ESPECIFICAS DEL LENGUAJE

En el apartado anterior se han visto las distintas palabras que puede crear el programador para especificar en el programa los datos, dispositivos, resultados, etc., que necesite emplear en el mismo.

El lenguaje COBOL contiene, por otra parte, un determinado número de palabras (1) para usos específicos que podemos clasificar en tres categorías:

- Verbos.
- Operadores.
- Palabras reservadas.

I-3.1. VERBOS

Los verbos en COBOL, como en cualquier otro lenguaje, definen una acción. Morfológicamente un VERBO es una palabra específica, y por consiguiente formada por caracteres, siempre letras, que se emplea para generar las instrucciones del programa que transforman los datos del mismo en resultados. Según la función específica encomendada a cada uno de estos verbos, se clasifican en cuatro categorías:

- Verbos del programa.
- Verbos del compilador.
- Verbos de clasificación.
- Verbos de control de entrada-salida.

I-3.1.1. Verbos del programa

Son los verbos que definen las acciones que debe realizar el ordenador en el curso de ejecución del programa. Se subdividen a su vez en cuatro tipos:

• *Verbos de entrada-salida*: Son los que generan las instrucciones necesarias para introducir los datos contenidos en soportes externos en la memoria central y los que extraen los resultados almacenados en la memoria central hacia los soportes externos.

• *Verbos aritméticos*: Son los encargados de generar las órdenes necesarias para realizar operaciones de tipo aritmético entre los datos almacenados en la memoria central.

• *Verbos manipulación y transferencia de datos*: Son los que generan las instrucciones necesarias para transferir datos de una parte a otra de la memoria.

(1) Este número oscila, en los diversos compiladores, entre 200 y 500 palabras específicas.

10/10/2010 10:10:10

10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10

10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10

10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10

10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10 10/10/2010 10:10:10

I-3.3.2. **Palabras optativas**

Son las que se utilizan exclusivamente en forma facultativa y cuya única misión es la de dar mayor claridad a la expresión escrita del lenguaje.

El ejemplo de las palabras IS y THAN y TO' en los operadores de relación es una muestra clara de lo anteriormente definido.

La lista, ordenada alfabéticamente, de todas las palabras reservadas del COBOL, con su significado específico, así como los compiladores, entre los indicados, que las emplean, se muestra en el *anexo A*.




CAPITULO II

GRAMÁTICA DEL LENGUAJE :

SINTAXIS





II - 1

FRASES



Como se observa, las frases de menor entidad las constituyen las expresiones, cláusulas y entradas. Con estos elementos agrupados se obtienen otras frases de mayor entidad: sentencias e inscripciones. La unión de varias sentencias forman párrafos. La unión de varios párrafos o de varias inscripciones constituyen secciones. Estas se agrupan para formar divisiones y el conjunto de las divisiones forman el programa.

El número y variedad de los conceptos designados difieren de un programa a otro.

Algunas de las divisiones pueden carecer de secciones. Algunas de éstas pueden carecer de párrafos y tener en su defecto inscripciones. Estas últimas pueden estar constituidas por sentencias o por otro tipo de inscripciones. Las sentencias las forman las declaraciones y éstas están formadas por expresiones. De la misma forma, las inscripciones están formadas por cláusulas.

En el capítulo III se estudia con detalle la estructura de un programa COBOL. En este capítulo nos limitamos a definir las características sintácticas de cada uno de los conceptos expuestos.

II - 2

EXPRESIONES Y
DECLARACIONES

II-2. EXPRESIONES Y DECLARACIONES

II-2.1. EXPRESIONES

Una expresión es la unión de varias palabras: nombres de datos, operadores y constantes que en el desarrollo de un programa se reducen a un *único valor*.

En función del tipo de operador empleado: aritmético o de relación, se clasifican, respectivamente, en dos tipos:

- Expresiones aritméticas.
- Expresiones condicionales.

II-2.1.1. Expresiones aritméticas

Al igual que en matemáticas, una expresión aritmética se forma uniendo nombres de datos, constantes numéricas y operadores aritméticos para reducirse a un *único valor numérico*, que dependerá del valor de los componentes de la expresión.

$$\text{NOMBRE-DATO-1} + \text{CONSTANTE-1} - \text{NOMBRE-DATO-2} / \text{CONSTANTE-2} \dots$$

expresión aritmética

Dependiendo de los valores de nombre-dato-1, nombre-dato-2 ..., constante-1, constante-2 ... y los operadores utilizados la expresión aritmética tomará un determinado valor.

En álgebra, la expresión $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ (una de las soluciones de una ecuación de segundo grado) se calcula en un orden determinado cuyas reglas se conocen. En COBOL las reglas son casi análogas, con pequeñas variaciones.

II-2.1.2. Jerarquía en el cálculo de una expresión aritmética

En una expresión aritmética, las operaciones se efectúan, por la máquina, en el siguiente orden de prioridad:

- Elevación de potencias.
- Multiplicaciones y divisiones.
- Adiciones y sustracciones.

Esta sistemática se regula, a su vez, de la siguiente forma:

- Tiene prioridad la parte de la expresión que está encerrada entre paréntesis (cuanto más interno, más prioridad).
- Dentro de cada grado de prioridad las operaciones se efectúan de izquierda a derecha de la expresión.

Ejemplo: $A + (B - C) * D + E * ((F / G) + 2.3) ** 2$ se calculará de la siguiente forma:

	RESULTADO INTERMEDIO	CAUSA
1.º) F / G	R1	Por ser el paréntesis más interno.
2.º) $B - C$	R2	Por ser, entre los dos paréntesis restantes, el extremo izquierdo.
3.º) $R1 + 2.3$	R3	Por ser el único paréntesis que queda en la expresión.
4.º) $R3 ** 2$	R4	Por ser la potenciación el grado de máxima prioridad.
5.º) $R2 * D$	R5	Por ser, entre las dos multiplicaciones existentes, la extrema izquierda.
6.º) $E * R4$	R6	Por ser la única multiplicación que queda en la expresión.
7.º) $A + R5$	R7	Por ser la primera adición (de izquierda a derecha).
8.º) $R7 + R6$	Resultado final de la expresión	

II-2.1.3. Expresiones condicionales

Una expresión condicional, también denominada expresión lógica, es la unión de nombres de datos, expresiones aritméticas, constantes y operadores de relación que se reducen a un único valor.

Primer término	Operador de relación (unión)	Segundo término
Expresión condicional		

Tanto el primer término que se denomina SUJETO, como el segundo término, denominado OBJETO, pueden estar constituidos por *uno* de los siguientes componentes:

- Nombre-dato.
- Constante numérica o no-numérica.
- Expresión aritmética.

Cualquier combinación sujeto-objeto es válida, excepto una constante en ambos.

De la misma forma que una expresión aritmética se reduce a un único valor numérico, una expresión condicional se reduce también a un único valor: *verdadero* o *falso* (*SI* o *NO*) dependiendo de que se cumpla o no la condición representada por la expresión.

II-2.1.4. Evaluación de una expresión condicional

La evaluación de una expresión condicional depende del contenido de la misma.

a) Si existen expresiones aritméticas será necesario reducirlas, previamente, a un valor numérico, según las reglas contenidas en el epígrafe II-2.1.2.

b) El resultado de la evaluación es función de tipo de operador de relación empleado.

- Si se utiliza el operador = o IS EQUAL TO.

La expresión es *verdadera* cuando el valor del primer término es igual al valor del segundo término y *falsa* en caso contrario.

- Si se utiliza el operador < o IS LESS THAN.

La expresión es *verdadera* cuando el valor del primer término es menor que el valor del segundo término y *falsa* en caso contrario.

- Si se utiliza el operador > o IS GREATER THAN.

La expresión es *verdadera* cuando el valor del primer término es mayor que el valor del segundo término y *falsa* en caso contrario.

c) El valor de cada término depende del elemento utilizado en los mismos.

- Si en ambos términos se utilizan expresiones aritméticas el valor de los mismos será el de la expresión aritmética.
- Si los dos términos utilizan nombre de datos y constantes, ambos han de tener un contenido numérico o no-numérico.
- Si uno de los términos utiliza una expresión aritmética el contenido del otro ha de ser numérico.

En resumen, la evaluación de la expresión dependerá del valor de cada término: numérico o no-numérico, debiendo ser ambos del mismo tipo; la comparación de ambos valores dará como resultado el valor de la expresión.

II-2.1.5. Comparación entre elementos

Se ha visto en el epígrafe anterior que para poder conocer el valor único a que se reduce una expresión condicional, es necesario efectuar una comparación de valores en una determinada condición.

Una comparación encierra dos matices:

CUANTITATIVO, o sea, la comparación de dos valores.

CUALITATIVO, que consiste en la comparación de características.

La comparación que realiza el ordenador es de tipo cuantitativo, aunque las cantidades a comparar no tienen por qué ser numéricas.

En estas circunstancias, el resultado de las comparaciones es siempre relativo, ya que si se comparan, por ejemplo, el carácter A con el carácter 9 el resultado de la comparación no sería el mismo en dos máquinas distintas. Según esto, se debe tener presente un concepto que se denomina SECUENCIA DE CLASIFICACION que es el orden de valor de cada carácter. Hay ordenadores configurados para considerar las letras de mayor orden en la clasificación que los números, pero otros están configurados totalmente al revés. Por ello, las comparaciones se realizan separando las dos categorías:

II-2.1.5.1. Comparación entre elementos numéricos

La comparación se realiza, exactamente igual que en álgebra, con las mismas reglas de signos.

Con signo positivo, ambos, es mayor el que tiene mayor valor absoluto.

Con signo negativo, ambos, es mayor el que tiene menor valor absoluto.

Con signos diferentes, es mayor el elemento positivo.

II-2.1.5.2. Comparación entre elementos no-numéricos

La comparación se realiza de izquierda a derecha del elemento, entre los caracteres que ocupen la misma posición relativa, hasta que aparezca un carácter en alguno de ellos que tenga mayor valor en la secuencia de clasificación. En este momento se interrumpe la comparación, sin comparar más caracteres, y se atribuye mayor valor al elemento que tenga en dicha posición relativa un carácter de mayor valor en la secuencia de clasificación. En el anexo B se indica la secuencia de clasificación que tienen establecida la mayoría de los ordenadores para los 51 caracteres básicos y que se tomará como base en todas las comparaciones que se realicen en los distintos ejemplos de esta obra.

EJEMPLO

	Contenido de TOTAL				VALOR DE LA EXPRESION
TOTAL < Ø .5	3 Ø.5 Ø.2				FALSA FALSA VERDADERA
	Contenido de				VALOR DE LA EXPRESION
	A	B	C	D	
(A + B) ** 2 = C + D	1	2	3	4	FALSA
	1	2	4	5	VERDADERA
	2	3	4	5	FALSA
	Contenido de				VALOR DE LA EXPRESION
	X	Y			
X ** 2 GREATER Y	1	2		FALSA	
	2	3		VERDADERA	
	2	4		FALSA	
	Contenido de				VALOR DE LA EXPRESION
	CAMPO-1	CAMPO-2			
CAMPO-1 = CAMPO-2	T. HURTADO	P. HELGUERO		FALSA, ya que el carácter T es mayor que el carácter P.	
CAMPO-1 > CAMPO-2	JUAN	JUAN		FALSA	
CAMPO-1 < CAMPO-2	H325	H32B		FALSA, ya que el carácter 5 es mayor que el carácter B.	

Expresiones incorrectas serían:

$$A + B + C < '324'$$

Ya que el resultado de una expresión aritmética es numérico, y no puede compararse con un valor no-numérico.

$$3.25 = 3.25\emptyset$$

Ya que aunque sería correcta, no pueden utilizarse dos constantes en la expresión.

II-2.1.6. Utilización de operadores lógicos en las expresiones condicionales

Los operadores lógicos AND, OR y NOT permiten combinar varias expresiones condicionales para obtener otra expresión condicional denominada compuesta.

En este tipo de expresiones la evaluación se obtiene en función del tipo de operador lógico empleado, además del resultado de la evaluación de las expresiones simples combinadas.

Si X e Y son expresiones condicionales, el resultado de la expresión compuesta, formada uniendo estas expresiones con uno de los operadores lógicos, da como resultado:

EXPRESION CONDICIONAL COMPUESTA	RESULTADO DE LA EXPRESION
X AND Y	VERDADERA Si X e Y verdaderas. FALSA Si X o Y, o ambas, falsas.
X OR Y	VERDADERA Si X o Y, o ambas, verdaderas. FALSA Si ambas falsas.
NOT X	VERDADERA Si X falsa. FALSA Si Y verdadera.

II-2.1.7. Orden de evaluación de una expresión condicional compuesta

El orden de evaluación de una expresión condicional compuesta se realiza con la siguiente prioridad:

- Evaluando las condiciones que tengan antepuesto el operador lógico NOT.
- Se evalúan a continuación los pares de condiciones unidas por el operador AND.
- Se evalúan, en último lugar, los pares de condiciones unidas por el operador OR.
- El orden de evaluación, a igualdad de operadores, se realiza de izquierda a derecha de la expresión.
- Si se utilizan paréntesis en la escritura de una expresión condicional compuesta, la evaluación se realiza en primer lugar con la parte de la expresión encerrada entre paréntesis. Si hay varios paréntesis se comenzará por el más interno.
- En la escritura de la expresión se debe dejar, al menos, un espacio entre cada expresión y el operador lógico que le precede.

Ejemplo: Si X, Y, Z, M, N y P representan a expresiones condicionales, la expresión condicional compuesta X AND Y OR NOT Z AND (M OR N) OR P se evaluará en el siguiente orden.

	Resultado intermedio	Causa
1.º M OR N	R1	Por estar contenido en un paréntesis. Por ser el más extremo de la izquierda.
2.º NOT Z	R2	
3.º X AND Y	R3	
4.º R2 AND R1	R4	
5.º R4 OR R3 OR R1	Final de la operación	

Si atribuimos a las distintas expresiones las valoraciones

X FALSO
 Y VERDADERO
 Z FALSO
 M FALSO
 N VERDADERO
 P FALSO

el resultado final de la expresión será:

	Resultado intermedio
1.º) M OR N	VERDADERO
2.º) NOT Z	VERDADERO
3.º) X AND Y	FALSO
4.º) R2 AND R1	FALSO
5.º) R4 OR R3 OR R1	VERDADERO

Resultado final

II-2.1.8. Pseudoexpresiones condicionales

Existen varias formas abreviadas de escribir expresiones condicionales en función del contenido del sujeto, operador y objeto de la expresión. Estas pseudoexpresiones son:

- Condiciones de signo.
- Condiciones de clase.
- Nombres de condición.

II-2.1.8.1. Condiciones de signo

Se utilizan cuando se desea saber si el valor de un nombre de dato o de una expresión aritmética es positivo, negativo o nulo.

Si, por ejemplo, se desea conocer si el resultado de la expresión aritmética $A + B$ es POSITIVO, será necesario escribir la siguiente expresión condicional

$$A + B > \emptyset$$

evaluar la condición y solamente en el caso de ser cierta la expresión sería positiva.

En cambio, si se utiliza la expresión $A + B$ IS POSITIVE, los resultados son los mismos; por consiguiente, $A + B > \emptyset$ es equivalente a $A + B$ IS POSITIVE; se han cambiado el operador y el objeto de la expresión por las palabras reservadas IS POSITIVE.

Las distintas condiciones de signo que pueden ser utilizadas son:

Sujeto	IS POSITIVE
Sujeto	IS NEGATIVE
Sujeto	IS ZERO

y también:

Sujeto	IS NOT POSITIVE
Sujeto	IS NOT NEGATIVE
Sujeto	IS NOT ZERO

Los resultados de la evaluación serán, para cada una de las seis expresiones:

Con	Resultado de la expresión	Cuando SUJETO es
IS POSITIVE	VERDADERO	$\geq \emptyset$
	FALSO	$< \emptyset$
IS NEGATIVE	VERDADERO	$< \emptyset$
	FALSO	$\geq \emptyset$
IS ZERO	VERDADERO	$= \emptyset$
	FALSO	$\neq \emptyset$
IS NOT POSITIVE	VERDADERO	$\leq \emptyset$
	FALSO	$> \emptyset$
IS NOT NEGATIVE	VERDADERO	$> \emptyset$
	FALSO	$\leq \emptyset$
IS NOT ZERO	VERDADERO	$\neq \emptyset$
	FALSO	$= \emptyset$

II-2.1.8.2. Condiciones de clase

Se utilizan para saber si el contenido de un determinado dato está formado totalmente por dígitos o totalmente por caracteres alfabéticos.

Las formas de expresión son:

Sujeto	IS NUMERIC
Sujeto	IS ALPHABETIC

En este caso el sujeto no puede ser una expresión aritmética, ya que su contenido es siempre numérico y sería ocioso efectuar la comprobación.

Los resultados de la evaluación para las dos posibles expresiones son:

CON	RESULTADO	CUANDO el sujeto
IS NUMERIC	VERDADERO	Está formado exclusivamente por dígitos.
	FALSO	Contiene, al menos, una letra o espacio.
IS ALPHABETIC	VERDADERO	Está formado exclusivamente por letras o espacios.
	FALSO	Contiene, al menos, un dígito.

II-2.1.8.3. Nombres de condición

Los nombres de condición forman por sí solos pseudoexpresiones condicionales. Según hemos visto en el capítulo I, un nombre de condición es el nombre de dato atribuido a un valor o conjunto de valores específicos de un nombre de dato. Vimos en el ejemplo del epígrafe I-2.1.1.2 que AMARILLO es el nombre de condición atribuido al valor 4 del nombre de dato COLOR.

Por consiguiente, la expresión $COLOR = 4$ es equivalente a AMARILLO. En otras palabras, la evaluación de la expresión

$COLOR = 4$, que será falsa para todos los valores de COLOR excepto 1 y verdadera para el valor 1, puede ser sustituida por

AMARILLO, que será:

	Cuando el sujeto sea
VERDADERA	= 4
FALSA	≠ 4

II-2.2. DECLARACIONES

La unidad inmediatamente superior a considerar es la *declaración*. La declaración, en COBOL, puede compararse en cierto modo a la cláusula del inglés. En su forma más simple, la declaración contiene un verbo y uno o varios elementos que en las declaraciones se denominan *operandos*, ya que son normalmente datos sobre los que se realiza alguna operación.

Existen dos tipos de declaraciones:

- Declaraciones imperativas.
- Declaraciones condicionales.

II-2.2.1. Declaraciones imperativas

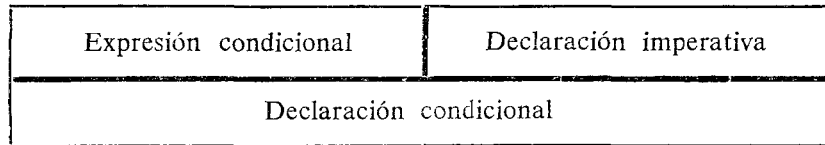
Son las declaraciones que se ejecutan en el programa sin estar sometidas a ninguna expresión condicional, es decir, se ejecutan siempre que en una secuencia del programa se pase a través de ella.

Las declaraciones imperativas que contienen un único VERBO se denominan declaraciones imperativas simples y las que contienen dos o más verbos se llaman compuestas.

Cuando se utilizan declaraciones imperativas compuestas, que son realmente varias imperativas simples, se puede utilizar la palabra reservada THEN para enlazarlas.

II-2.2.2. Declaraciones condicionales

Las declaraciones imperativas, conforme se acaba de estudiar, instruyen al ordenador para que ejecute determinadas acciones específicas, en función del verbo utilizado. Estas declaraciones dan instrucciones o indicaciones de un carácter específico e inequívoco; al ordenador no le queda opción de no ejecutarlas. Sin embargo, muchas veces se desea que el ordenador ejecute una operación solamente en determinadas circunstancias. En tal caso, es necesario modificar una declaración imperativa, de modo que su ejecución dependa de la evaluación de una expresión condicional. El conjunto de una expresión condicional y una o varias declaraciones imperativas constituyen una *declaración condicional*.



El estudio y los ejemplos de las declaraciones condicionales se realiza junto con los verbos de control de secuencia en el capítulo VI.

II - 3

SENTENCIAS E INSCRIPCIONES

II-3. SENTENCIAS E INSCRIPCIONES

II-3.1. SENTENCIAS

La sentencia es la unidad básica de expresión del lenguaje COBOL, como lo es en el idioma inglés. Una sentencia COBOL se compone de una o más declaraciones, la última de las cuales termina en un punto.

En función de la naturaleza del verbo contenido en las declaraciones que constituyen las sentencias, éstas se clasifican en:

- Sentencias imperativas.
- Sentencias condicionales.
- Sentencias de instrucciones para el compilador.

II-3.1.1. Sentencias imperativas

Las sentencias imperativas están formadas exclusivamente por declaraciones imperativas, la última de las cuales debe terminar en un punto.

Los verbos que confieren carácter imperativo a las sentencias son los verbos de entrada-salida, los verbos aritméticos y los verbos de manipulación de datos.

II-3.1.2. Sentencias condicionales

Las sentencias condicionales están formadas por una declaración condicional que termine en un punto. Esta declaración condicional puede ir precedida de una declaración imperativa con tal que no termine en uno de los verbos de control de secuencia GO TO y STOP RUN que estudiaremos en el capítulo VI.

II-3.1.3. Sentencias de instrucciones para el compilador

Una sentencia de instrucciones para el compilador es aquella que está formada por uno de los verbos que dirigen la acción del compilador. La finalidad de este tipo de sentencias es la de facilitar la compilación del programa o la de permitir la inclusión de comentarios del programador, dentro del programa, con el objetivo de documentarlo a efectos de comprensión. Una sentencia de instrucción al compilador no puede contener más de un verbo del compilador ni puede contener ninguno de los tipos de verbos del programa.

II-3.2. INSCRIPCIONES

De la misma forma que las sentencias indican la acción que debe realizarse en el programa, en función del verbo utilizado, las inscripciones indican las características de los datos y dispo-

sitivos que intervienen en el programa para, mediante la aplicación de las sentencias, transformarlos en resultados.

Las inscripciones no contienen verbos, ya que no indican acción. En su lugar están formados por determinadas palabras reservadas; una palabra clave como mínimo y varias palabras optativas

Existen, fundamentalmente, tres tipos de inscripciones:

- Inscripciones de descripción de dispositivos.
- Inscripciones de descripción de ficheros.
- Inscripciones de descripción de registros.

El estudio del primer tipo de inscripciones constituye el contenido fundamental del capítulo IV y el de los demás tipos el contenido del capítulo V.

II-3.3. PUNTUACION DE LAS SENTENCIAS E INSCRIPCIONES

Como todo lenguaje escrito, existe en COBOL una puntuación. Esta permite realizar un programa más legible y más próximo a la escritura normal. A condición de respetar ciertas reglas de carácter obligatorio la puntuación no es siempre necesaria.

- La coma y el punto y coma son frecuentemente caracteres opcionales.

Sin embargo, como ya hemos visto anteriormente, la coma y los paréntesis son indispensables en la escritura de un nombre de índice.

En los demás casos, la coma y el punto y coma sirven de puntuación en la escritura de las sentencias e inscripciones con el fin de hacer la lectura del programa más fácil.

- El punto indica al compilador el fin de una instrucción. Un punto mal colocado puede entrañar consecuencias desastrosas en la ejecución de un programa.
- Los paréntesis facilitan la escritura de las expresiones aritméticas y condicionales y cambian el orden de evaluación de las mismas.

REGLAS GENERALES DE PUNTUACIÓN

- 1.^a Dos palabras consecutivas y las expresiones entre paréntesis deben estar separadas, como mínimo, por un espacio. Si hay varios espacios, éstos se consideran como uno solo, salvo cuando éstos forman parte de un literal no-numérico.
- 2.^a Un punto, una coma o un punto y coma deben estar siempre seguidos de un espacio.
- 3.^a Los paréntesis izquierdos deben estar seguidos de un espacio si el carácter siguiente es un signo + o un signo -. En los demás casos el espacio no es obligatorio.
- 4.^a Los paréntesis derechos deben estar precedidos obligatoriamente de un espacio si el carácter precedente es otro paréntesis derecho.
- 5.^a Todas las sentencias deben terminar con un punto y éste debe estar seguido de un espacio.
- 6.^a Todos los operadores aritméticos y de relación deben estar precedidos y seguidos por un espacio.

II - 4

PARRAFOS, SECCIONES
Y DIVISIONES

II-4. PARRAFOS, SECCIONES Y DIVISIONES

II-4.1. PARRAFOS

Al igual que ocurre con un idioma cualquiera, las sentencias de COBOL pueden agruparse para formar párrafos, con la única diferencia que en COBOL es totalmente necesario dar un nombre a dicha agrupación, o lo que es lo mismo, a las sentencias que forman un párrafo, mediante el nombre de párrafo o procedimiento que les agrupa.

Por consiguiente, cuando sea necesario en un programa recurrir a una determinada sentencia y sólo a ella, ésta debe formar por sí misma un párrafo y debe dársele un nombre único a la misma.

II-4.2. SECCIONES

Sección es una secuencia de párrafos que tiene asignado un nombre. Su función es la de formar grupos lógicos mayores que los formados por los párrafos o procedimientos. Por ejemplo, si un tratamiento describe una operación por medio de una secuencia de sentencias y esta operación es parte de una secuencia mayor de operaciones, a esta última se la denomina sección. Cuando se referencia un nombre de sección se accede a todos los componentes de la misma. Por ejemplo, se puede ejecutar una secuencia de tratamientos agrupados en una sección, refiriéndose, en forma adecuada, a dicha sección.


II-4.3. DIVISIONES

La última gran agrupación que se realiza en un programa COBOL es la división. De hecho, el COBOL está estructurado en divisiones cuyo cometido y contenido específico es el objetivo del estudio del capítulo siguiente.

CAPITULO III

ESTRUCTURA DE UN PROGRAMA :


HOJA DE CODIFICACION



C O N T E N I D O

III-1. ESTRUCTURA DE UN
PROGRAMA COBOL

III-2. HOJA DE CODIFICACION
COBOL



III - 1

ESTRUCTURA DE UN PROGRAMA
COBOL

III-1. ESTRUCTURA DE UN PROGRAMA COBOL

Para que un ordenador pueda resolver en forma automática un problema es necesario que se le proporcione un **programa** de actuación, consistente en unas normas de actuación que le den respuesta a las siguientes cuestiones:

- ¿Qué datos deben ser procesados?
- ¿Dónde están almacenados?
- ¿Cómo están organizados?
- ¿Qué resultados se deben obtener?
- ¿Qué tratamientos hay que dar a los datos para obtener los resultados?

Un programa escrito en lenguaje COBOL, entre sus muchas cualidades, posee la de dar respuesta a esta cuestión de forma separada, totalmente independiente y diferenciada, de forma tal que programadores distintos pueden escribir cada uno de las partes integrantes del programa, pudiéndose reunir y enlazar todas las partes constituyendo el programa COBOL.

El COBOL se ha estructurado de una forma estandarizada y modular. Para ello se ha dividido en cuatro partes denominadas divisiones, cada una de las cuales cubre una misión específica, sirviendo a su vez de apoyo a las restantes, proporcionando la información que necesita el ordenador para poder llevar a cabo la resolución del programa.

Las cuatro divisiones de un programa COBOL son:

- IDENTIFICATION DIVISION o división de identificación.
- ENVIRONMENT DIVISION o división de características.
- DATA DIVISION o división de datos.
- PROCEDURE DIVISION o división de procedimientos.

En este capítulo solamente se estudian las características básicas de cada una de las divisiones y en los capítulos IV, V y VI se realiza el estudio detallado de cada uno de sus componentes.

En la figura III-1 se muestra esquemáticamente la descomposición de un programa en sus partes integrantes.

III.1.1. IDENTIFICATION DIVISION

La División de Identificación debe ser la primera división en cada programa fuente COBOL. Identifica al programa fuente y la información especificada en esta división se incluye en el listado del programa fuente. Su objeto fundamental es el de contener el nombre por el cual se conoce el programa en la instalación.

No contiene secciones, sino únicamente párrafos con nombres reservados.

PROGRAMA COBOL

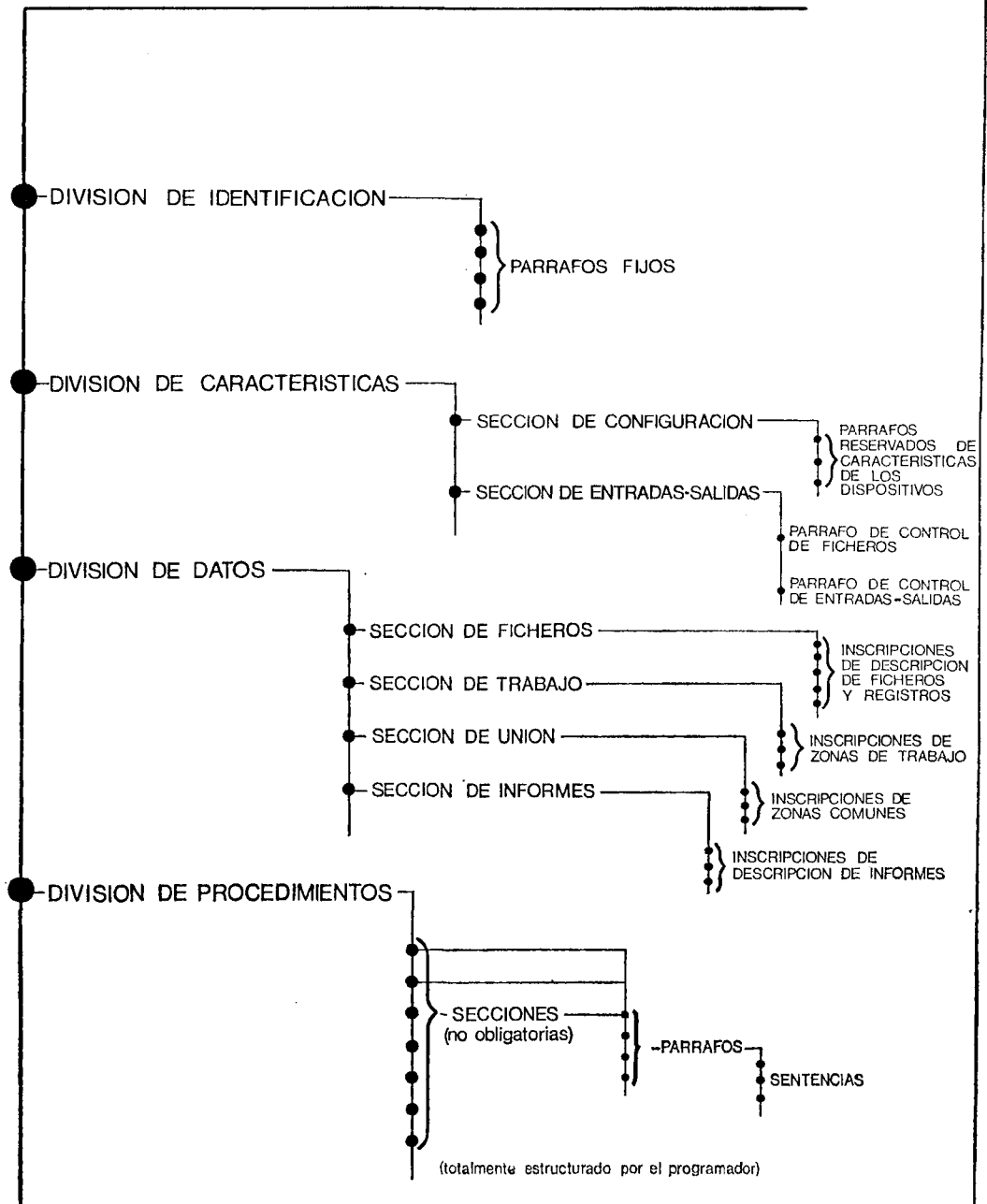


FIGURA III-1

III-1.2. ENVIRONMENT DIVISION

La División de Características debe incluirse en cada programa COBOL inmediatamente después de la División de Identificación. Define los aspectos del proceso que depende de las características físicas del ordenador. Indica si el programa objeto será ejecutado en un ordenador diferente del que efectúa la compilación incluso con una configuración distinta.

La División de Características se estructura en dos secciones: la Sección de Configuración, que especifica el ordenador fuente y el ordenador objeto y la Sección de Entrada-Salida, que describe los dispositivos de almacenamiento externo y las técnicas específicas de almacenamiento. Ambas secciones están formadas por párrafos con nombres reservados.

III-1.3. DATA DIVISION

Todos los datos a los que se haga referencia en la División de Procedimientos deben estar definidos en la Data Division. Los datos pueden estar organizados en forma de ficheros, utilizados con valores fijos, o utilizados como salida en forma de informes impresos. Cada tipo de datos se describe en una sección independientemente. Si los datos en alguna sección no son necesarios en el programa, se puede omitir alguna de las secciones.

Las secciones se componen en esta división de inscripciones de descripción, en lugar de párrafos.

III-1.4. PROCEDURE DIVISION

Las sentencias en la División de Procedimientos describen las operaciones que deben realizarse por el programa objeto. La ejecución del programa objeto comienza con la primera sentencia de la División de Procedimientos. Las sentencias se ejecutan en el orden en que aparecen en el programa, a menos que exista una alteración en la secuencia por algunos de los verbos de control de secuencia.

Esta división está estructurada en secciones optativas y creadas por el programador y éstas en párrafos, también con nombres dados por el programador.

III-1.5. NOTACIONES EMPLEADAS EN EL ESTUDIO DE LAS INSTRUCCIONES QUE COMPONEN LAS DIVISIONES DEL PROGRAMA

[]

Todas las palabras o las partes de una orden colocada entre [] pueden ser incluidas u omitidas en una instrucción COBOL, a la que se haga referencia, a elección del programador.

{ }

En la descripción de una orden COBOL, ciertas informaciones se colocan una sobre otra entre { }. Estas informaciones representan diversas opciones utilizables. Debe elegirse una sola de entre estas opciones. Las restantes deben omitirse.

Opción

Esta notación se colocará, bien entre [], bien entre { }. La elección de una de las opciones representada por el rectángulo dependerá del tipo de signos, [] o { }, entre los que aparezca.

.....	Los puntos suspensivos, cuando figuran detrás de [] o de { }, significan que el contenido de [] o de { } puede repetirse un cierto número de veces.
_____	Todas las palabras subrayadas son absolutamente obligatorias. Deberán, pues, figurar en la instrucción COBOL, a menos que la parte de la instrucción que las contenga sea facultativa. En la escritura del programa, el programador no debe subrayar nada.
Mayúsculas	Todas las palabras escritas con mayúsculas son palabras reservadas. Las que están subrayadas son obligatorias y las que no lo estén son facultativas.
Minúsculas	Todas las inscripciones en minúsculas dentro de los formatos corresponden a informaciones que debe suministrar el programador (ejemplos: nombre-dato, literal-no-numérico, número-entero, etc.).
Caracteres especiales	Aunque los caracteres especiales utilizados en las instrucciones (= + • * ; etc.) no estén subrayados, son siempre obligatorios en la escritura de las instrucciones que los utilicen.
Puntuación	Cuando se indica puntuación en una instrucción COBOL significan que ella es obligatoria y que ha de formar parte de la estructura de la instrucción.

III - 2

HOJA DE CODIFICACION COBOL

III-2. HOJA DE CODIFICACION COBOL

III-2.1. INTRODUCCION

Los lenguajes de programación necesitan unas formas de expresión que, al estar encaminadas a que sean interpretadas por los ordenadores, deben ser más rígidas que las expresiones usadas en el lenguaje corriente.

Nosotros, al expresarnos mediante el lenguaje escrito, utilizamos como medio de transmisión una cuartilla de papel totalmente en blanco, o a lo sumo cuadrículada o rayada, para facilitar la escritura. Además, estamos acostumbrados a dejar un cierto margen a la izquierda, a subrayar palabras que a nuestro juicio merecen mayor énfasis que otras, etc.

El COBOL, a pesar de ser uno de los lenguajes de programación que tienen mayor libertad de expresión, necesita someterse a unas reglas determinadas para poder confeccionar los programas fuente si queremos que el ordenador, en la etapa posterior de traducción a programa objeto, tenga unas determinadas referencias para poder distinguir los diversos tipos de elementos que se le presentan en el programa.

La herramienta que utiliza el programador para escribir su programa es una hoja con formato generalizado, denominada hoja de codificación COBOL.

Los fines que cumplen esta hoja son varios:

- **STANDARIZACION.** Está universalmente adoptada por todos los fabricantes de ordenadores de forma tal que todos los programas escritos en COBOL, para cualquier ordenador, se plasman en este formulario.
- **IDENTIFICACION.** Por la disposición que adoptan los nombres dentro del formulario, el programa traductor COBOL, puede identificar, por ejemplo, los nombres de procedimientos de los nombres de datos.
- **COMUNICACION.** La estandarización y normalización de la hoja de codificación COBOL facilita la comunicación entre el programador que escriba el programa y el operador de la perforadora de fichas, que convierte el programa escrito en el conjunto de fichas perforadas que componen el programa fuente.
- **LIBERTAD.** A pesar de las necesarias limitaciones a las que ha de someterse el programador en la escritura: reglas de sintaxis, espaciados, puntuaciones, etc., la hoja de codificación tiene muchos grados de libertad, hasta el punto que dos programas iguales con las mismas instrucciones e incluso escrito por el mismo programador, puede presentar una distribución sobre la hoja completamente distinta de uno a otro.

III-2.2. FORMATO DE LA HOJA DE CODIFICACION COBOL

La hoja de codificación, como muestra la figura III-2, contiene 80 columnas de información que corresponden a las 80 columnas de una ficha perforada normal.

Instituto de Informática

Hoja de Codificación en Cobol

PROGRAMA		INSTRUCCIONES PARA PERFORACION										HOJA	DE
		SISTEMA	SIGNO										FORMATO DE FICHA
PROGRAMADOR		FECHA	PERFORACIONES										

Página	Línea	Cont.	A	B											Identificación					
1	3	4	6	7	8	12	15	20	25	30	35	40	45	50	55	60	65	70	73	80
		0	1	0																
		0	2	0																
		0	3	0																
		0	4	0																
		0	5	0																
		0	6	0																
		0	7	0																
		0	8	0																
		0	9	0																
		1	0	0																
		1	1	0																
		1	2	0																
		1	3	0																
		1	4	0																
		1	5	0																
		1	6	0																
		1	7	0																
		1	8	0																
		1	9	0																
		2	0	0																

FIGURA III-2

Cada línea del formulario se utiliza para escribir la información que posteriormente ha de perforarse en una ficha.

Como la hoja dispone de 25 líneas, se pueden obtener por cada hoja hasta 25 fichas perforadas, que compondrán 25 fichas del programa fuente, utilizándose tantas hojas de codificación como sean necesarias para completar el programa fuente.

Este programa fuente no es directamente ejecutable. El proceso de transformación en programa ejecutable, es decir, en programa objeto, se muestra esquemáticamente en la figura III-3.

Las hojas de codificación se transcriben en fichas perforadas, a través de las perforaciones realizadas en una perforadora normal de fichas. El conjunto de fichas obtenidas, que deberán mantener el mismo orden que las líneas de la hoja de codificación, se introducen en la memoria central a través de la lectora de fichas.

El programa traductor que reside normalmente en un disco o cinta magnética estará almacenado también en memoria central y se procederá a la compilación dando como resultado un listado del programa fuente con los posibles errores de compilación, y en el caso de no existir errores, opcionalmente y a través de la perforadora de fichas, un conjunto de fichas que constituirá el programa objeto.

Los elementos que contiene la codificación son los siguientes:

III-2.2.1. Elementos que no se transcriben al programa fuente

Todos estos elementos no son necesarios en la codificación del programa fuente. Sin embargo, deberán ser cumplimentados por el programador con el fin de documentar al máximo el programa.

Programa: Se utiliza para indicar en forma de clave, abreviada o explícita, el nombre dado al programa por el programador.

Ejemplos:

— CALCULO DE LA NOMINA.

— A 750728.

— CALNOM.

— Etc.

Programador: El autor del programa.

Sistema: Se utiliza exclusivamente en los Centros de Proceso de Datos que disponen de diversos sistemas (IBM, SIEMENS, UNIVAC, etc.), ya que es necesario realizar, en determinadas fichas, algunas transformaciones en el caso de quererse emplear en sistemas distintos para los que se ha efectuado el programa.

Fecha: Se indica aquí la fecha en que se ha terminado la escritura del programa y ha sido entregado para su perforación.

Instrucciones para su perforación: Se utiliza cuando se emplean signos especiales que no tienen perforación directa en la máquina perforadora de fichas. El programador indicará al perforista las combinaciones de dígitos numéricos o de zona que conformen el carácter que deba ser perforado.

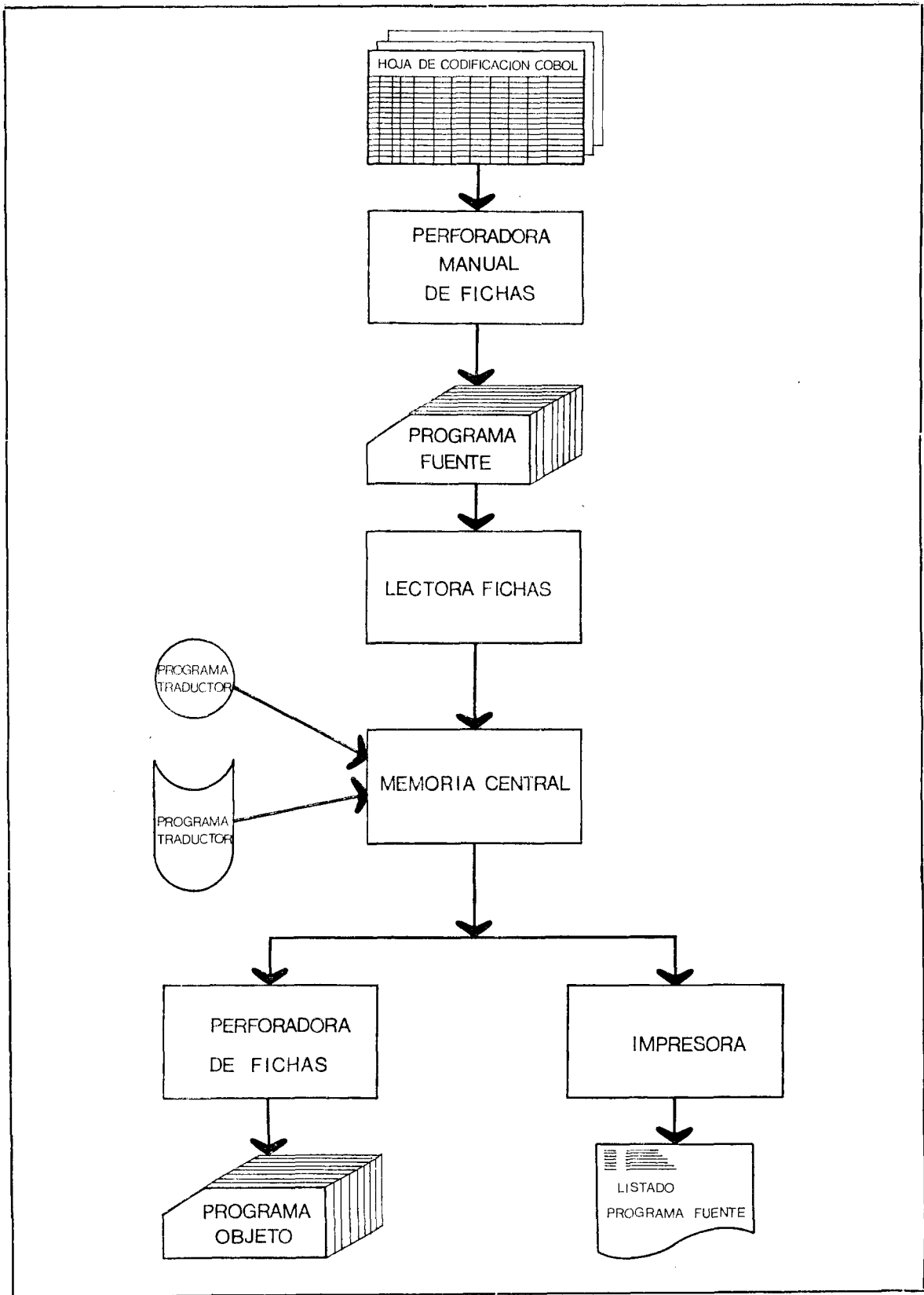


FIGURA III-3

III-2.2.2. Elementos que se transcriben al programa fuente

El resto de la información que figura en la hoja de codificación se transcribirá íntegramente a las fichas perforadoras, construyendo el programa fuente.

Cada uno de estos elementos debe ser ineludiblemente cumplimentado por el programador.

Página	Línea	A	B	Identificación
1	2	3	4	5
001	010			
	020	CALCULO		
	030	MULTIPLY	PREC BY	DESCUENTO
	040			
	050	-		PROG-EJE
	060			
	070			
	080			
	090			
	100			
	110			
	120			
	130			
	140			
	150			
	160			
	170			
	180			
	190			
	200			

FIGURA III-4

La figura III-4 muestra el nombre de cada uno de los elementos que constituyen el programa propiamente dicho. Estos elementos son los siguientes:

Número de secuencia: Cada una de las líneas de la hoja de codificación se identifica con un número de seis cifras denominado número de secuencia. Este número se compone de dos partes: número de página y número de línea.

El primero se anota en las columnas 1 a 3 y permanece invariable en cada una de las líneas que componen una hoja de codificación; el segundo, normalmente preimpreso, viene anotado en las columnas 4 a 6 de la misma.

El objetivo de este número de secuencia es establecer un orden en las fichas que componen el programa fuente. Con esta numeración, una vez perforadas las fichas, no existe el peligro de cambiar el orden de secuencia ascendente del programa en el caso de que accidentalmente se trastocase el orden en el que se han establecido inicialmente.

El número de página se indica solamente en la primera línea de cada página. En las restantes líneas (de la 2 a la 20 ó 25, se traza una línea ondulada en sentido vertical que el perforista entiende como repetición de la misma información.

El número de línea, como ya se ha indicado, está normalmente preimpreso por comodidad y ahorro de tiempo del programador. La numeración de este documento se realizará en intervalos de 10 en 10, dejando libres los números de línea intermedios.

La justificación de esta forma de proceder es la siguiente: frecuentemente el programador, una vez realizado el programa o en el curso de la escritura, tiene la necesidad de intercalar entre dos líneas de información una o varias que ha omitido por duda o por descuido. Si el número de las líneas fuese consecutivo y tuviese que efectuar una intercalación, necesitaría, a partir de la intercalada, modificar los números de línea siguientes hasta la última de la página.

Con la técnica de numerar de diez en diez se le permite intercalar hasta nueve líneas entre cada dos ya escritas sin necesidad de ninguna modificación subsiguiente.

Las cinco últimas líneas de las 25 que componen la hoja de codificación se utilizan para fines de intercalación. Por consiguiente, no se emplean en la escritura normal del programa, dejándolas de reserva para el caso de necesitar posibles intercalaciones.

En el listado de la compilación se reproduce en la impresora el programa fuente *independientemente del número de secuencia*. Sin embargo, si el número de secuencia no es creciente entre dos fichas consecutivas se indica esta anomalía mediante una anotación al margen y que corresponde a la ficha, teóricamente desordenada. Esta anotación, que suele ser una S, sirve para indicarle al programador un posible error a la colocación de las fichas que componen el programa fuente.

Indicador de continuación: Se utiliza, empleándose la columna número 7, para indicar que un *literal no numérico* que ha sobrepasado la capacidad de una línea se continúa en la siguiente. Como la longitud máxima permitida, en un literal no numérico, es de 120 posiciones, se da con frecuencia el caso de necesitarse más de una línea para escribirlo. Las normas de sintaxis son las siguientes:

- El literal no numérico puede escribirse a partir de cualquiera de las columnas 12 a 72.
- Para continuar su escritura en una línea distinta a la indicada se anota un guión en la columna 7 de la línea siguiente y una comilla en la columna anterior a la que se desea continuar escribiendo el literal interrumpido.

Ejemplo: Si se desea escribir el siguiente literal no numérico: COLOQUE EN LA IMPRESORA EL PAPEL DE EDICION DE FACTURAS. Se procederá de la siguiente forma:

Página	Línea	7	8	12	15	45	50	55	60	65	70
	0 1 0										
	0 2 0										
	0 3 0										
00 1	0 4 0										
00 1	0 5 0	—				'	C	O	L	O	Q
00 1	0 6 0					'	P	A	P	E	L
	0 7 0					'	S	'			
	0 8 0										
	0 9 0										

Sentencias e inscripciones del programa: Las columnas 8 a 72 se utilizan para contener las inscripciones y sentencias que componen el programa propiamente dicho. Los distintos ele-

mentos que componen la sintaxis del programa deben ser situados a partir de dos márgenes que comienzan en las columnas 8 y 12 y que reciben, respectivamente, los nombres de márgenes A y B.

El margen A se emplea para alinear las partes de mayor entidad del programa. En la columna 8 se deben comenzar a escribir con carácter obligatorio los siguientes elementos:

- Los nombres de división.
- Los nombres de sección.
- Los nombres de párrafo.
- Las inscripciones de nivel FD y RD.

Los elementos que pueden escribirse alineados a partir del margen, pero sin carácter obligatorio, son los siguientes:

- Los números de nivel, aunque los números de nivel aconsejables son los 01, 77 y 88

El margen B limita, obligatoriamente, la escritura de los siguientes elementos:

- Las sentencias.
- La continuación de los literales no numéricos.

Para los restantes elementos no mencionados explícitamente con anterioridad es aconsejable mantener el margen B como margen inicial de su escritura.

En la separata del final del capítulo se muestra un ejemplo de un programa en el que se han subrayado los elementos que obligatoriamente han de escribirse en los márgenes indicados, así como los márgenes recomendados para los elementos que no tienen indicación obligatoria.

página	línea	margen A	margen B
001	010		IDENTIFICATION DIVISION.
001	020		PROGRAM-ID. EJEMPLO1.
001	030		DATE-WRITTEN. FEB-75.
001	040		DATE-COMPILED. FEB 1975.
001	050		REMARKS. ESTE PROGRAMA, COMO EJEMPLO DEL CAPITULO III,
001	051		SE UTILIZA EXCLUSIVAMENTE PARA MOSTRAR AL
001	052		LECTOR EL REFLEJO DE LA SINTAXIS COBOL EN UNA
001	053		HOJA DE CODIFICACION, ES DECIR, LA UBICACION
001	054		DE LOS ELEMENTOS DEL LENGUAJE SOBRE EL MODELO STANDARD
001	060		ENVIRONMENT DIVISION.
001	070		CONFIGURATION SECTION.
001	080		SOURCE-COMPUTER. UNIVAC-9300.
001	090		OBJECT-COMPUTER. UNIVAC-9300.
001	100		INPUT-OUTPUT SECTION.
001	110		FILE-CONTROL.
001	120		SELECT FICH-ENT ASSIGN TO TAPE 1.
001	130		SELECT FICH-SAL ASSIGN TO TAPE 2. → nombre de dispositivo
001	140		DATA DIVISION.
001	150		FILE SECTION.
001	160	FD	FICH-ENT → nombre de fichero
001	170		RECORDING F
001	180		BLOCK CONTAINS 10 RECORDS
001	190		RECORD CONTAINS 85 CHARACTERS
001	200		LABEL RECORD STANDARD
001	210		DATA RECORD ENT.
001	220	01	ENT.
001	230		02 FILLER PICTURE X(6).
001	240		02 CENTRO.
001	250		03 CEN-1 PICTURE X. → nombre de dato
002	010		88 CENTRAL VALUE '1'.
002	020		88 REGIONAL VALUE '2'.
002	030		03 FILLER PICTURE XXX. → nombre de condición
002	040		02 CEN-2 PICTURE XX.
002	050		02 FILLER PICTURE X(73).
002	060	FD	FICH-SAL
002	070		RECORDING F
002	080		BLOCK CONTAINS 10 RECORDS
002	090		RECORD CONTAINS 85 CHARACTERS
002	100		LABEL RECORD STANDARD
002	110		DATA RECORD SAL.
002	120	01	SAL.
002	130		02 FILLER PICTURE X(85).
002	140		WORKING-STORAGE SECTION.
002	150	77	WREGENT PICTURE S9(7) VALUE ZERO.
002	160	77	WREGSAL PICTURE S9(7) VALUE ZERO.
002	170		PROCEDURE DIVISION.
002	180		APERTURA.
002	190		OPEN INPUT FICH-ENT OUTPUT FICH-SAL.
002	200		LEER SECTION.
002	210		INICIO.
002	220		READ FICH-ENT AT END GO TO FIN-T → VERBO
002	230		IF CENTRO = '2104' DISPLAY ENT. → literal
002	240		ADD 1 TO WREGENT.
002	250		IF CEN-2 = '24' GO TO INICIO.
003	010		CAMBIO. → nombre de párrafo
003	020		IF REGIONAL AND CEN-2 NOT = ZERC NEXT SENTENCE ELSE → c
003	030		MOVE SPACE TO CEN-2.
003	040		WRITE SAL FROM ENT.
003	050		ADD 1 TO WREGSAL.
003	060		GO TO INICIO.
003	070		FIN-T.
003	080		DISPLAY 'REGISTROS LEIDOS' WREGENT UPON CONSOLE.
003	090		DISPLAY 'REGISTROS GRABADOS' WREGSAL UPON CONSOLE.
003	100		CLOSE FICH-ENT FICH-SAL.
003	110		STOP RUN.



CAPITULO IV

LAS DIVISIONES DE UN PROGRAMA :


IDENTIFICATION
Y ENVIRONMENT



C O N T E N I D O

IV-1. IDENTIFICATION
DIVISION

IV-2. ENVIRONMENT DIVISION



IV - 1

IDENTIFICATION DIVISION



IV-1. IDENTIFICATION DIVISION

IV-1.1. GENERALIDADES

Esta división, que debe estar siempre presente, es la primera de un programa COBOL. Su cometido, fundamentalmente, es el de identificar. Esta división comprende un párrafo obligatorio que contiene el nombre del programa dado por el programador. Este nombre es por el cual el sistema identifica al programa.

Además del objetivo principal de esta división, se pueden incluir otras informaciones de carácter optativo que permiten conocer otras particularidades específicas que el programa traductor no tiene en cuenta en la compilación, siempre y cuando no existan errores de sintaxis y que únicamente se reproduce en el listado del programa fuente.

Esta división contiene exclusivamente párrafos con nombres prefijados y reservados.

IV-1.2. CONTENIDO DE LA IDENTIFICATION DIVISION

Los párrafos que pueden contener esta división y su orden de escritura son los siguientes:

- PROGRAM-ID.
- AUTHOR.
- INSTALLATION.
- DATE-WRITTEN.
- DATE-COMPILED.
- SECURITY.
- REMARKS.

IV-1.2.1. Párrafo PROGRAM-ID

Se utiliza para contener el nombre dado al programa por el programador y por el cual va a ser reconocido el programa por el sistema. Es un párrafo obligatorio. Debe ir seguido, separado como mínimo de un espacio, del nombre del programa. Las reglas para formar este nombre varían con el tipo de sistema.

El cuadro IV-1 contiene las reglas de formación del nombre del programa para cada uno de los sistemas estudiados.

SISTEMA	REGLAS DE FORMACION	OBSERVACIONES
		El sistema tiene en cuenta únicamente los:
CDC Sistema 6000	<ul style="list-style-type: none"> • Se forma con 1 a 30 caracteres alfanuméricos • El primero ha de ser alfabético 	7 primeros caracteres
IBM Sistemas 360 y 370	<ul style="list-style-type: none"> • Se forma con 1 a 8 caracteres alfanuméricos • El primero ha de ser alfabético • El nombre ha de escribirse entre comillas 	Todos
UNIVAC Sistema 1108	<ul style="list-style-type: none"> • Se forma con 5 a 30 caracteres alfanuméricos • El primero ha de ser alfabético 	5 primeros caracteres
UNIDATA HW-B Sistema GE-100	<ul style="list-style-type: none"> • Se forma con 1 a 30 caracteres alfanuméricos • El primero ha de ser alfabético 	8 primeros caracteres
UNIDATA Siemens Sistema 4004	<ul style="list-style-type: none"> • Se forma con 1 a 8 caracteres alfanuméricos • El primero ha de ser alfabético • El nombre ha de escribirse entre comillas 	Todos
UNIDATA CII Sistema IRIS 50	<ul style="list-style-type: none"> • Se forma, al menos, con 8 caracteres • El primero ha de ser alfabético 	8 primeros caracteres
UNIVAC Sistemas 9200-9300	<ul style="list-style-type: none"> • Se forma con 6 a 30 caracteres alfanuméricos • El primero ha de ser alfabético 	5 primeros caracteres
UNIDATA HW-B Sistema GE-600	<ul style="list-style-type: none"> • Se forma con 1 a 6 caracteres alfanuméricos • El primero ha de ser alfabético 	Todos

CUADRO IV-1.—Reglas de formación de los nombres del programa

IV-1.2.2. Párrafo AUTHOR

Es un párrafo opcional que cuando se utiliza sirve para contener el nombre del programador que ha escrito el programa. Como es obvio, no existe ninguna limitación en su escritura, pudiéndose emplear la longitud y caracteres que se deseen e incluso los puntos y signos de puntuación necesarios.

IV-1.2.3. Párrafo INSTALLATION

Es un párrafo opcional que cuando se utiliza sirve para contener cualquier comentario que se desee incluir sobre la instalación o centro de proceso de datos para el cual se ha escrito el programa. Tiene importancia especial en aquellas empresas que posean varios centros de proceso de datos y para los que puede existir un mismo programa con diversas variantes en función de la instalación específica.

IV-1.2.4. Párrafo DATE-WRITTEN

Es un párrafo opcional que cuando se utiliza sirve para contener la información relativa a la fecha en que se ha escrito el programa fuente (fecha de comienzo o de final, o ambas).

IV-1.2.5. Párrafo DATE-COMPILED

También opcional. Su empleo, dentro de los opcionales, es el más frecuente; permite que el compilador inserte en el listado del programa fuente la fecha en que se ha realizado la compilación, la cual se toma del propio sistema. Cualquier comentario que se indique en este párrafo es reemplazado por la fecha indicada.

IV-1.2.6. SECURITY

Opcional. Raramente utilizado, salvo en aquellas instalaciones en que por circunstancias especiales el acceso a un programa queda reservado a determinadas personas autorizadas específicamente o cuando la utilización del programa esté condicionada a determinadas circunstancias. Todas estas referencias, personas y circunstancias se incluyen como un comentario dentro del párrafo.

IV-1.2.7. Párrafo REMARKS

Párrafo opcional. Está reservado, como su nombre indica, para recoger cualquier tipo de anotación o comentario que describa el objeto del programa. En este párrafo se incluye normalmente una breve descripción del contenido del programa.

00001 001010 IDENTIFICATION DIVISION.
00002 001020 PROGRAM-ID. EJEMPLO.
00003 001030 AUTHOR.
00004 001040 DATE-WRITTEN. 6-2-75.
00005 001050 DATE-COMPILED. MAR 10,1975.
00006 001060 REMARKS.

00007 ESTE PROGRAMA HA SIDO CONFECCIONADO ESPECIALMENTE
00008 PARA ILUSTRAR A MODO DE EJEMPLO, DIVERSOS CAPITULOS
00009 DEL TEXTO ASI PUES EL LECTOR PODRA OBSERVAR QUE AL
00010 FINAL DE LOS CAPITULOS IV-1, IV-2, V-3, V-4 Y V-5
00011 FIGURAN EN FORMA DE SEPARATA LAS DISTINTAS PARTES
00012 DEL PROGRAMA QUE SE INICIAN EN ESTE CAPITULO

00013
00014 TAMBIEN AL FINAL DE LOS CAPITULOS III Y V-7 SE
00015 INCLUIDO SEPARATAS QUE ILUSTRAN EL CONTENIDO DE
00016 RESPECTIVOS CAPITULOS ASI COMO EN EL APENDICE
00017 SE HACE ENTRE PROGRAMAS

00018
00019 ESTAS SEPARATAS CONTIENEN PROGRAMAS COMPLETOS
00020 QUE TIENEN RELACION; COMO SE PUEDE OBSERVAR, CON EL
00021 PROGRAMA

00022
00023 EN ESTE PROGRAMA SE HA BUSCADO, FUNDAMENTALMENTE
00024 UTILIZAR EL MAXIMO NUMERO DE INSTRUCCIONES COMO
00025 POR CONSECUENCIA ALGUNOS PROCEDIMIENTOS PODRIAN
00026 RESULTAR DE MAS BREVE REDACCION, EMPLEANDO OTROS
00027 TIPOS DE INSTRUCCIONES DISTINTAS A LAS UTILIZADAS

00028
00029 EN LA PARTE DERECHA DE CADA SEPARATA SE COMENTAN
00030 POR PUNTO EL SIGNIFICADO DE CADA UNA DE LAS INSTRUCCIONES
00031 CODIFICADAS

00032
00033 DE ESTA FORMA, PRETENDEMOS QUE EL LECTOR TENGA
00034 FACILIDAD DE LECTURA AL PRESENTARSE EN LA MISMA SEPARATA
00035 DEL PROGRAMA Y COMENTARIOS.

00036
00037 EL PROGRAMA QUE SE COMENTA EFECTUA LA DEPURACION DE
00038 LOS DATOS CONTENIDOS EN UN FICHERO DE FICHAS Y LOS
00039 DATOS SOMETIENDO CADA UNO DE LOS CAMPOS, A UN CONTROL
00040 DE VALIDEZ

00041
00042 LOS REGISTROS QUE NO CONTIENEN NINGUN CAMPO EFECTIVO
00043 SON GRABADOS EN UN FICHERO DE CINTA MAGNETICA Y
00044 LOS REGISTROS QUE CONTIENEN AL MENOS UN CAMPO EFECTIVO
00045 SON IMPRIMIDOS

00046
00047 SE EFECTUA PARA DETERMINADO CAMPO UNA CONSULTA EN
00048 UN FICHERO CON ORGANIZACION DIRECTA Y COMO CONSECUENCIA
00049 DE ESTA CONSULTA SE ACTUALIZA EL CONTENIDO DE DICHO FICHERO

00050
00051 LA COMPILACION DE ESTE PROGRAMA SE REALIZA CON LA
00052 VERSION ANSÍ COBOL, VER APENDICE CORRESPONDIENTE
00053 PARA ESTUDIAR LAS VARIACIONES EXISTENTES CON LA VERSION
00054 COBOL COBASYL.

00055
00056
00057
00058
00059
00060
00061
00062
00063

	00000100
	00000200
	00000300
	00000400
	00000500
	00000600
NTE	00000700
TULOS	00000800
E AL	00000900
VI	00001000
TES DEL	00001100
	00001200
	00001210
HAN	00001300
E SUS	00001400
DE EN-	00001500
	00001505
	00001510
QUE NO	00001600
STE	00001700
	00001800
	00001810
NTE,	00001900
OL Y	00002000
HABER	00002100
RAS	00002200
	00002300
	00002310
A PUNTO	00002400
S-	00002500
	00002600
	00002610
COMODA	00002700
A HOJA,	00002800
	00002900
	00002910
ON DE	00003000
ERFORA-	00003100
EQUEO	00003200
	00003300
	00003310
RO-	00003400
Y	00003500
ERRO-	00003600
	00003610
	00003620
A	00003700
-	00003800
ITENIDO	00003900
	00003910
	00004000
LA	00004100
ITE,	00004200
L CO-	00004300
	00004400
	00004500
	00004600
	00004700
	00004800
	00004900
	00005000
	00005100
	00005200

Esta división no necesita ningún comentario adicional.

No se ha incluido el párrafo SECURITY, que es opcional ya que el programa no requiere, por sus características, ninguna medida de seguridad especial.

En las posiciones correspondientes a las columnas 73 a 80, en las que se codifica la identificación del programa fuente, el traductor incluye una numeración secuencial que utiliza para posteriores actualizaciones, siempre y cuando dicho programa se incluya en una biblioteca de programas fuente.

La fecha MAR 10, 1975, la incluye el programa traductor en el momento de la compilación. Se aconseja la lectura detenida del párrafo REMARKS para comprender el objetivo del programa.

Agradecemos, en este punto, la colaboración del programador Sr. Díaz Carpintero en la puesta a punto de este programa ejemplo.

IV - 2

ENVIRONMENT DIVISION

IV-2. ENVIRONMENT DIVISION

IV-2.1. GENERALIDADES

La división de características físicas del programa describe, como su nombre ya indica, las características del ordenador que va a compilar el programa fuente y la del ordenador que va a ejecutar el programa objeto, que pueden ser diferentes. Describe, asimismo, todas las características de los dispositivos periféricos, elementos de entrada-salida que son necesarios para que se ejecute el programa objeto.

Las funciones encomendadas a esta división pueden descomponerse en las siguientes:

- a) Describir la configuración del sistema que va efectuar la compilación.
- b) Describir la configuración del sistema que ejecutará el programa objeto.
- c) Relacionar los nombres simbólicos de los dispositivos que intervienen en la ejecución del programa.
- d) Relacionar los nombres simbólicos de todos los ficheros de entrada-salida que deben utilizarse en la ejecución del programa objeto.
- e) Indicar todos los procedimientos automáticos de repetición, números de áreas de memoria para contener los registros de los ficheros de entrada y salida y las técnicas de acceso y creación de los referidos ficheros de entrada y salida.

Todas estas funciones de descripción de características se realizan en dos secciones con nombres reservados, la primera de las cuales es obligatoria.

La estructura general de la División de Características se muestra en el cuadro IV-2.

IV-2.2. SECCION DE CONFIGURACION

IV-2.2.1. Párrafo SOURCE-COMPUTER

Este párrafo, obligatorio, indica cuál es el modelo de ordenador utilizado para compilar el programa.

Contiene como cláusula obligatoria la inscripción correspondiente al número de modelo y como cláusula optativa la cláusula MEMORY

	SECCIONES	PARRAFOS	CLAUSULAS
ENVIRONMENT DIVISION	CONFIGURATION SECTION	SOURCE-COMPUTER	MEMORY SIZE y n.º de modelo
		OBJECT-COMPUTER	MEMORY SIZE y n.º de modelo
			SEGMENT-LIMIT
		SPECIAL-NAMES	CONSOLE
			CURRENCY
			DECIMAL-POINT
	Nombres-nemotécnicos		
	INPUT-OUTPUT SECTION	FILE-CONTROL	ASSIGN
			MULTIPLE REEL
			RESERVE
			FILE-LIMIT
			ACCESS
			ORGANIZATION
			KEY
			PROCESSING
			FILE-CODE
		TRACK-AREA	
		I-O-CONTROL	SAME
			RERUN
APPLY			
MULTIPE-FILE			

CUADRO IV-2

SIZE X CARACTERES, en donde X indica el tamaño de la memoria necesaria para poder compilar el programa.

Ejemplos:

— SOURCE-COMPUTER. 6600 (para CDC).

— SOURCE-COMPUTER. 100 MEMORY SIZE 32 K CARACTERES (para HONEYWELL-BULL, etc.).

IV-2.2.2. Párrafo OBJECT-COMPUTER

Este párrafo, también obligatorio, indica el ordenador para el que se genera el programa objeto.

Contiene como cláusula obligatoria la inscripción correspondiente al número del modelo, y como cláusulas optativas la cláusula MEMORY SIZE Y CARACTERES, en donde Y indica el tamaño de la memoria necesaria para poder ejecutar el programa objeto y la cláusula SEGMENT-LIMIT IS número-entero, en la que el número entero expresa un número de prioridad para los casos en que el programa se haya descompuesto en segmentos o partes cuyo conjunto forma el programa total.

IV-2.2.3. Párrafo SPECIAL-NAMES

Este párrafo, opcional, se utiliza para agrupar determinadas cláusulas especiales referentes a particularidades de los distintos elementos de entrada-salida.

Las cláusulas posibles son:

- **CONSOLE IS** nombre-dispositivo.
- **CURRENCY SIGN IS** literal-no numérico.
- **DECIMAL-POINT IS COMMA.**
- Nombre-dispositivo **IS** Nombre-dato

IV-2.2.3.1. Cláusula CONSOLE

FORMATO

CONSOLE IS nombre-dispositivo

OBJETO

La función de esta cláusula es la de proporcionar un nombre nemotécnico a la consola de mando del sistema. Tiene aplicación con los verbos ACCEPT y DISPLAY, que se estudian en el capítulo VI.

IV-2.2.3.2. Cláusula CURRENCY

FORMATO

CURRENCY SIGN IS literal-no-numérico

OBJETO

La función de esta cláusula es la de reemplazar el signo \$ de moneda americana, que se utiliza en la cláusula PICTURE de edición de importes (ver capítulo V) por el correspondiente al signo de moneda local del país utilizador del ordenador.

PARTICULARIDADES DE SU USO

El COBOL, al ser creado en Estados Unidos, surgió con caracteres o signos propios de dicho país. Sin embargo, al ser utilizado en otros países se puede reemplazar alguno de los caracteres especiales por otros específicos del usuario. Esta cláusula es un ejemplo de dichas posibilidades.

El literal no-numérico tiene que ser de un solo carácter de entre la lista de caracteres básicos del COBOL, con la excepción de los siguientes:

Ø	1	2	3	4	5	6	7	8	9	A	B
C	D	E	J	K	L	P	R	S	V	X	Z
+	—	*	,	.	;	()	'	∅		

IV-2.2.3.3. Cláusula DECIMAL-POINT

FORMATO

DECIMAL-POINT IS COMMA

OBJETO

En el estudio de los caracteres básicos, capítulo I, veíamos que el signo especial • se utiliza en la formación de literales numéricos como indicador de la coma decimal en los números no enteros, ya que en los países anglosajones éste es también el uso comercial habitual.

Con el empleo de esta cláusula se puede transformar, en la edición de resultados, el signo • por el , de uso corriente en países no anglosajones.

IV-2.2.3.4. Cláusula de nombres nemotécnicos

FORMATO

nombre-dispositivo **IS** nombre-dato

OBJETO

Esta cláusula permite dar un nombre nemotécnico (representado por un nombre de dato) a la consola o a un equipo periférico determinado. Los nombres nemotécnicos así definidos se emplean en la División de Procedimientos en los verbos particulares de entrada-salida ACCEPT y DISPLAY.

PARTICULARIDADES DE SU USO

El nombre de dispositivo utilizado no puede ser empleado en otra parte de la Environment Division.

En algunos compiladores el nombre CONSOLE no se considera como un nombre de dispositivo. En estos casos la cláusula CONSOLE IS nombre-dato permite asignar a la consola un nombre nemotécnico.

IV-2.3. SECCION DE ENTRADA-SALIDA

Esta sección no es obligatoria. Su nombre ha de figurarse cuando se utilice alguno de los párrafos que componen la sección.

Permite describir determinadas características de las entradas y salidas de información para el programa considerado.

Los dos párrafos que únicamente puede contener son:

- FILE-CONTROL
- I-O-CONTROL

IV-2.3.1. Párrafo FILE-CONTROL

Este párrafo se utiliza para asignar a cada fichero que se utilice en el programa el nombre del dispositivo que la corresponde dentro del equipo periférico, necesario para poder ejecutar el programa. Proporciona también información al programa traductor sobre la forma en que están organizados dichos ficheros.

Este párrafo está formado exclusivamente por sentencias denominadas SELECT, existiendo tantas como ficheros se traten en el programa.

Cada una de las sentencias SELECT está formada por la propia palabra reservada SELECT acompañada de una serie de cláusulas cuyo número varía con el tipo de ordenador empleado.

Como puede ocurrir que un fichero dentro de un programa tenga un uso optativo, la sentencia SELECT puede ir acompañada de la palabra OPTIONAL, que indicará esta circunstancia.

También puede suceder que dos ficheros tengan la misma organización y descripción dentro de la DATA DIVISION de un mismo programa. Esto se produce, por ejemplo, en el caso de la actualización de un fichero secuencial. En estas circunstancias se escribe en la sentencia SELECT la palabra reservada RENAMING.

En resumen, el contenido del párrafo FILE-CONTROL está formado por una o varias sentencias SELECT con el siguiente formato:

SELECT [OPTIONAL] nombre-fichero-1 [RENAMING nombre-fichero-2] [cláusulas optativas]
--

[Cualquiera de las figuradas en el cuadro IV-2.]

Cada una de las cláusulas que puede acompañar a la sentencia SELECT tiene aplicación en los distintos compiladores de los sistemas estudiados, según lo indicado en el cuadro IV-3.

IV-2.3.1.1. Cláusula ASSIGN

OBJETO

Se utiliza para asignar un dispositivo físico de la instalación al fichero.

FORMATO

El formato de esta cláusula obligatoria varía sensiblemente según el equipo para el que se programa. En términos generales el formato de esta cláusula es:

CLAUSULAS	CDC sistema 6000	IBM sistemas 360-370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS 50	UNIVAC sistemas 9200-9300	HW-B sistema GE 600
ASSIGN	*	*	*	*	*	*	*	*
MULTIPLE	*		*	*	*		*	
RESERVE	*	*	*		*		*	
FILE-LIMIT	*	*	*	*	*		*	
ACCESS	*	*	*	*	*		*	
ORGANIZATION	*	*	*		*		*	
KEY	*	*	*	*	*		*	
PROCESSING	*		*	*			*	
FILE-CODE				*				
TRACK-AREA		*						

(*) Aplicable en el sistema indicado.

CUADRO IV-3

ASSIGN TO nombre-dispositivo [opción-1] [opción 2]

**PARTICULARIDADES
DE SU USO**

- Como nombre de dispositivo se puede utilizar cualquiera de los indicados en la figura I-1 del capítulo I.
- Como opción-1 se puede utilizar alguna de las siguientes palabras reservadas:
 - **DIRECT-ACCESS** para dispositivos de acceso directo (disco magnético o tambor).
 - **UTILITY** para cualquier dispositivo rápido de acceso secuencial (disco magnético y banda magnética).
 - **UNIT-RECORD** para los dispositivos de registros unitarios (lectora-perforada de fichas e impresora).
- Como opción-2 se emplea el número de modelo del dispositivo físico o un número entero, variable del 0 al 15 más la palabra reservada **UNIT**.

IV-2.3.1.2. Cláusula *MULTIPLE-REEL*

Esta cláusula, opcional, tiene aplicación exclusivamente en ficheros soportados sobre bandas magnéticas cuando son necesarias varias bobinas para contener el fichero total. No tiene más valor que el de documentación para el sistema y no afecta en nada al código generado por el compilador.

IV-2.3.1.3. Cláusula *RESERVE*

OBJETO

La función de esta cláusula es la de reservar o no reservar, dependiendo de la opción elegida, áreas de entrada y/o salida adicionales al área de trabajo necesaria para almacenar el bloque o registro físico leído mediante una instrucción de lectura o el bloque que se va grabar mediante una instrucción de escritura.

Si se dispone de memoria central suficiente la velocidad del proceso aumentará, ya que mientras se realizan operaciones con los datos contenidos en un área de entrada y/o salida se pueden ir grabando o leyendo datos en un área o «buffer» adicional.

FORMATO

El formato de esta cláusula es:

```
[ RESERVE { NO / N.º-entero } ALTERNATE AREA[S] ]
```

PARTICULARIDADES DE SU USO

El número de áreas adicionales reservadas depende del número entero figurado en la cláusula.

Si figura la palabra NO no se reservará ningún área adicional.

Si figura número entero se reservan tantas áreas adicionales como indique el valor de número entero.

En caso de omisión de esta cláusula se reserva un área adicional además del área necesaria.

EJEMPLO

RESERVE 3 ALTERNATE AREAS. Reserva tres áreas adicionales de entrada o salida, en función del fichero al que haga referencia esta cláusula.

IV-2.3.1.4. Cláusula *FILE-LIMIT*

OBJETO

La función de esta cláusula es la de indicar el número de pistas que van a utilizarse en la creación de un fichero en disco magnético.

FORMATO

El formato de esta cláusula es:

```
[ FILE-LIMIT IS n.º-entero-1 [THRU n.º-entero-2] [n.º-entero-3 [THRU n.º-entero-4] ] . . . . ]
```


- La omisión de esta cláusula indica organización secuencial.
- En los equipos UNIVAC se utiliza la opción INDEXED tanto en el caso de organización directa como en el de organización secuencial-indexada.

IV-2.3.1.7. Cláusulas KEY

OBJETO

Se utiliza esta cláusula para indicar los campos del registro que se utilizan como identificadores del mismo en ficheros con acceso directo.

FORMATO

Los formatos de estas cláusulas son:

ACTUAL KEY IS nombre-dato-1
SIMBOLIC KEY IS nombre-dato-2
RECORD KEY IS nombre-dato-3

PARTICULARIDADES DE SU USO

Esta cláusula es obligatoria cuando se utiliza la opción RANDOM de la cláusula ACCESS.

El número de cláusulas KEY a utilizar depende del tipo de organización empleado en el fichero.

La cláusula ACTUAL especifica, a través de nombre de dato, el número de pista relativo de un fichero en el que tiene que buscarse o situarse un registro.

La cláusula RECORD se utiliza en ficheros con organización index-secuencial e indica, a través de nombre-dato-3, el campo del registro que contiene el identificativo del mismo.

La cláusula SIMBOLIC se utiliza para localizar, a través de nombre-dato-2, el registro que tenga el mismo identificativo cuando se trata de una búsqueda y para crear el identificativo del registro cuando se trata de escribirlo.

IV-2.3.1.8. Cláusula PROCESSING

OBJETO

Esta cláusula se utiliza para indicar el orden en que los registros van a ser procesados.

FORMATO

PROCESSING MODE IS SEQUENTIAL

PARTICULARIDADES DE SU USO

Esta cláusula se utiliza exclusivamente con carácter documental y no produce por consiguiente ningún efecto en el programa objeto.

IV-2.3.1.9. Cláusula FILE-CODE

OBJETO

La función de esta cláusula es la de preparar la etiqueta de identificación de un fichero cuando éste contenga etiquetas estándar.

c) Utilización para indicar *puntos de reanudación*.

Cuando la ejecución de un programa tiene una larga duración se puede, después de ejecutarse una parte, almacenar sobre un fichero el contenido y el estado de la memoria central, así como otros registros necesarios al sistema, con el fin de reiniciar posteriormente el tratamiento en este punto.

Esto se realiza normalmente cuando el programador piensa que puede haber algún riesgo o incidente durante el desarrollo del programa. Si llega a producirse este hecho y se han tomado las debidas precauciones mediante las correspondientes inscripciones en este párrafo, no será necesario volver a iniciar la ejecución del programa desde su principio, sino solamente a partir del punto en donde se ha realizado el almacenamiento de la información o punto de reanudación.

Estos puntos de reanudación se realizan, la mayoría de las veces, según el número de registros que haya que tratar en un determinado fichero.

Las posibles cláusulas que integran el párrafo I-O-CONTROL o control de entrada-salida son:

- Cláusula SAME.
- Cláusula RERUN.
- Cláusulas APPLY.
- Cláusula MULTIPLE-FILE.

Todas estas cláusulas no se utilizan en los diversos equipos estudiados, aunque las dos primeras son comunes a todos. En el cuadro IV-4 se recoge la utilización específica de cada cláusula en cada sistema.

EQUIPOS	CLAUSULAS			
	SAME	RERUN	APPLY	MULTIPLE-FILE
CDC Sistema 6000	*	*		*
IBM Sistemas 360 y 370	*	*	*	
UNIVAC Sistema 1108	*	*		
UNIDATA HW-B Sistema GE-100	*	*	*	
SIEMENS Sistema 4004	*	*	*	
UNIDATA-CII Sistema IRIS-50	*	*	*	
UNIVAC Sistemas 9200-9300	*	*	*	
UNIDATA HW-B SISTEMA GE-600	*	*	*	

(*) Aplicable en el sistema indicado.

CUADRO IV-4

IV-2.3.2.1. Cláusula SAME

OBJETO

La función de esta cláusula es la de especificar al programa traductor que reserve el mismo área para dos o más archivos (SAME significa mismo), precisamente los archivos cuyos nombres se incluyan en la cláusula.

FORMATO

El formato de esta cláusula es:

SAME AREA FOR nombre-fichero-1 [nombre-fichero-2]

PARTICULARIDADES DE SU USO

Los archivos que se nombran en esta cláusula no pueden abrirse simultáneamente mediante una declaración OPEN, ver epígrafe VI-2.3.1, sino que debe abrirse uno cada vez mediante declaraciones OPEN distintas.

Pueden utilizarse tantas cláusulas SAME como se deseen o necesiten, pero ningún nombre de archivo puede aparecer en más de una cláusula SAME.

IV-2.3.2.2. Cláusula RERUN

OBJETO

La función de esta cláusula es la de permitir el establecimiento de «puntos de control» o «puntos de reanudación» durante la ejecución de un programa con el fin de poder, en el caso de ocurrir una interrupción accidental o provocada, reiniciar la ejecución del mismo en los puntos establecidos.

Un punto de comprobación es una información del estado del ordenador en un punto dado en la ejecución del programa objeto. Contiene toda la información necesaria para volver a empezar el programa desde ese punto.

Esta cláusula tiene dos formatos posibles:

FORMATO

Formato 1.º:

[**RERUN** [**ON** { Nombre-dispositivo } { Nombre-fichero } **EVERY END OF** { REEL } { UNIT } **OF** nombre-fichero-2]...]

Formato 2.º:

[**RERUN ON** { Nombre-dispositivo } { Nombre-fichero-1 } **EVERY número-entero** { **RECORDS OF** nombre-fichero-2 } { **CLOCK-UNITS** }]

**PARTICULARIDADES
DE SU USO**

Dependiendo de los formatos elegidos, el establecimiento de los puntos de reanudación tiene lugar en distintos puntos o fases de ejecución del programa.

Con el formato 1.º, los puntos de reanudación se establecen físicamente en el dispositivo que se mencione, bien por su nombre de archivo o bien por su nombre-externo o de dispositivo, es decir, se almacena en el dispositivo mencionado la información relativa al estado del programa justamente en el momento o cada vez que se llegue al final físico de un volumen en el archivo que se indique en *nombre-de-archivo-2*.

Con el formato 2.º, los puntos de reanudación se establecen en función de la opción elegida.

Con la opción RECORDS OF, ocurrirá *cada vez* que se hayan procesado del archivo cuyo nombre es *nombre-de-archivo-2* tantos registros como se indique en *número-entero*.

Con la opción CLOCK-UNITS ocurrirá cada vez que transcurran tantas *unidades de tiempo* (normalmente minutos) como se indiquen en *número-entero*.

IV-2.3.2.3. *Cláusulas APPLY*

Las cláusulas de aplicación específica o cláusulas APPLY se utilizan en función de las circunstancias especiales que concurren en determinados momentos de la ejecución de un programa.

Existe una gran variedad de cláusulas APPLY, las cuales difieren bastante entre los traductores que las utilizan. Las más generalizadas son:

[**APPLY FIXED LENTH ON** nombre-fichero-1
[nombre-fichero-2]...]

Esta cláusula, utilizada por HW-B, sirve para indicar que el fichero o ficheros a los cuales hace referencia contiene exclusivamente registros de longitud fija (equivalente por tanto a la cláusula RECORDING MODE IS F que no se emplea en los compiladores de estos equipos).

[**APPLY** } **RANDOM**
 } **SEQUENTIAL** } **FORMAT ON** nombre-fichero-1
 } **INDEXED** }
[nombre-fichero-2]...]

Esta cláusula de HW-B se emplea para indicar la organización física de un fichero en disco (equivale a la cláusula ORGANIZATION, ya

estudiada, que emplean otras firmas). Si se omite el sistema supone que la organización es RANDOM, es decir, aleatoria.

APPLY nombre-condición TO FORM-OVERFLOW ON nombre-fichero

Esta cláusula que emplean IBM y SIEMENS es muy utilizada, ya que facilita enormemente el control de salto de página en la impresora de forma automática, a través del canal 12 de la cinta de carro o cinta de control de la impresora, liberando así al programador de la preocupación de llevar la cuenta en el programa del número de líneas impresas; de esta forma únicamente deberá proporcionar en dicho programa, un nombre de condición que exprese la capacidad excedida, es decir, cuando se llena la página. En consecuencia, el programador solamente necesitará preguntar en el programa, antes de imprimir una línea, cuál es el valor del nombre de condición; por ejemplo, si el nombre de condición establecido en la condición de llenado de página fuese PAGINA-LLENA, además de escribir la cláusula APPLY en la forma:

APPLY PAGINA-LLENA TO FORM-OVERFLOW
ON IMPRESORA

deberá incluir en la PROCEDURE DIVISION antes de escribir un verbo de escritura una condición tal como:

IF PAGINA-LLENA WRITE LINEA AFTER ADVANCING 2 LINES
(ver epígrafe VI-2.3.3)

**[APPLY RESTRICTED SEARCH OF n.º-entero TRACKS
ON nombre-fichero]**

Esta cláusula, utilizada por IBM y SIEMENS, se aplica sólo a los archivos con organización directa. Se emplea para limitar el número de pistas que intervienen en la búsqueda de un registro que se va a localizar o en la búsqueda de espacio libre para un registro que se va a grabar.

IV-2.3.2.4. Cláusula *MULTIPLE-FILE*

OBJETO

Se utiliza cuando un carrete de banda magnética contiene más de un fichero; es equivalente a la opción REEL de la cláusula FOR MULTIPLE REEL, que en el caso de CDC en la FILE-CONTROL, solamente tiene efectos de documentación.

FORMATO

El formato de esta cláusula es:

MULTIPLE FILE TAPE CONTAINS nombre-fichero-1
[**POSITION** n.º-entero-1] [nombre-fichero-2
[**POSITION** n.º-entero-2] ...]

Un ejemplo del uso de esta cláusula sería:

MULTIPLE FILE TAPE CONTAINS FICHERO-A FICHERO-B

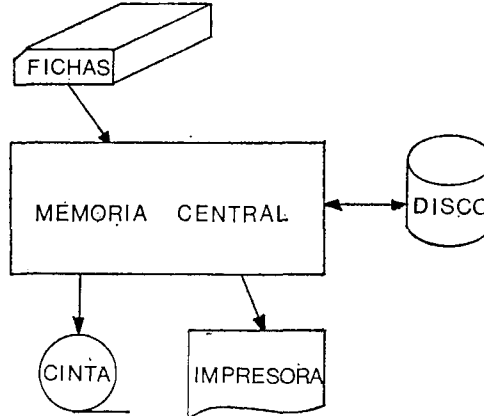
que indicaría que la cinta a la que se haga referencia en la cláusula **SELECT** de la **FILE-CONTROL** contiene dos ficheros que son el **FICHERO-A** y el **FICHERO-B**.

00064	001080	ENVIRONMENT DIVISION.	
00065	001090	CONFIGURATION SECTION.	
00066	001100	SOURCE-COMPUTER. IBM-370-158.	
00067	001110	OBJECT-COMPUTER. IBM-370-158.	
00068			
00069			
00070	001120	SPECIAL-NAMES.	
00071	001125	CO1 IS LINNEA1	
00072	001130	DECIMAL-POINT IS COMMA.	
00073			
00074			
00075	001140	INPUT-OUTPUT SECTION.	
00076			
00077	001150	FILE-CONTROL.	
00078	001160	SELECT FICHAS	ASSIGN TO UT-S-SYS004.
00079			
00080			
00081			
00082	001170	SELECT DISCO	ASSIGN TO DA-I-SYS005
00083	001180	ACCESS IS RANDOM	
00084	001190	NOMINAL KEY IS CLAVE-ID	
00085	001200	RECORD KEY IS CLAVE-DISCO.	
00086			
00087			
00088			
00089	001210	SELECT IMPRESORA	ASSIGN TO UT-S-SYS006.
00090			
00091			
00092			
00093	001220	SELECT CINTA	ASSIGN TO UT-S-SYS007
00094	001225	FOR MULTIPLE REEL	
00095	001230	RESERVE NO ALTERNATE AREA.	
00096			
00097			
00098			
00099			
00100			
00101			
00102			
00103			
00104			
00105			
00106			
00107			
00108			
00109			
00110			
00111			
00112			
00113			
00114			
00115			
00116			
00117			
00118			
00119			
00120			
00121			
00122			
00123			
00124			
00125			
00126			

00006400
 00006500
 00006600
 00006700
 00006800
 00006900
 00007000
 00007100
 00007200
 00007300
 00007400
 00007500
 00007600
 00007700
 00007800
 00007900
 00008000
 00008100
 00008200
 00008400
 00008500
 00008600
 00008700
 00008800
 00008900
 00009000
 00009100
 00009200
 00009300
 00009400
 00009500
 00009600
 00009700
 00009800
 00009900
 00010000
 00010100
 00010200
 00010300
 00010400
 00010500
 00010600
 00010700
 00010800
 00010900
 00011000
 00011100
 00011200
 00011300
 00011400
 00011500
 00011600
 00011700
 00011800
 00011900
 00012000
 00012100
 00012200
 00012300
 00012400
 00012500
 00012510
 00012610

Se indica que tanto la compilación como la ejecución del programa se realiza con un ordenador IBM, sistema 370, modelo 158.

En este párrafo de nombres especiales se indica que:
 — al canal 01 se le atribuye el nombre simbólico LINNEA1.
 — la coma decimal sustituye en la edición al punto decimal anglosajón.



La configuración necesaria para ejecutar el programa es la indicada en la figura u organigrama de proceso.

El fichero en fichas, denominado con el nombre de fichero FICHAS, y que contiene los datos o registros que van a ser sometidos a una depuración se asigna al nombre de dispositivo SYS 004. Sus características son: UT, utility (utilidad) como clase de dispositivo.

S, secuencial como modo de acceso.

El fichero con disco, denominado con el nombre de fichero DISCO, contiene información para ser consultada con cada registro del fichero FICHAS y ser actualizado en caso necesario; se le asigna el nombre de dispositivo SYS005. Sus características son:

DA, direct access (acceso directo) como clase de dispositivo.

I, secuencial con índices o indexado como modo de acceso.

El fichero sobre impresora, denominado con el nombre de fichero IMPRESORA, se genera para contener los registros del fichero en fichas que contienen, al menos, algún error en cualquiera de sus campos. Sus características son:

UT, utility (utilidad) como clase de dispositivo.

S, secuencial como modo de acceso.

El fichero en banda negativa, denominado con el nombre CINTA, se genera para contener los registros del fichero en fichas que no contienen ningún error en ninguno de sus campos. Sus características son:

UT, utility (utilidad) como clase de dispositivo.

S, secuencial como modo de acceso.

Se indica, también, para este fichero, que es multivolumen y que no se reserva nada más que un área de entrada para albergar los registros del fichero.

Se indica para el fichero DISCO que la clave simbólica o nominal se encuentra en el campo de nombre de dato CLAVE-ID de la WORKING-STORAGE SECTION y la clave de registro en el campo de nombre de dato CLAVE-DISCO del registro.

CAPITULO V

LAS DIVISIONES DE UN PROGRAMA :

DATA DIVISION

C O N T E N I D O

- V-1. ORGANIZACION DE LOS DATOS
- V-2. ORGANIZACION DE LA DATA DIVISION
- V-3. SECCION DE FICHEROS
- V-4. SECCION DE TRABAJO O MANIOBRA
- V-5. SECCION DE CONSTANTES
- V-6. SECCION COMUN DE ALMACENAMIENTO O SECCION DE UNION
- V-7. SECCION DE INFORMES

V - 1

ORGANIZACION DE LOS DATOS

V-1. ORGANIZACION DE LOS DATOS

V-1.1. GENERALIDADES

Sabemos que tanto los datos que alimentan un programa y que van a ser elaborados por el ordenador hasta convertirlos en resultados como las instrucciones que componen el programa de actuación, comparten la memoria central durante la ejecución del programa. Por otra parte, suponemos que el lector conoce las nociones básicas de organización de datos en ficheros.

La primera cuestión que puede planteárenos es la siguiente: ¿Cómo puede distinguirse dentro de la memoria interna lo que es dato de lo que es instrucción? En principio no se puede realizar esta distinción por la simple lectura de los mismos, ya que ambos, físicamente, están constituidos por un conjunto de BITS energizados o no; para poder realizar dicha discriminación es necesario que el programador facilite en su programa cierta información sobre los datos, de forma tal que el ordenador conozca la ubicación de los mismos en memoria a través de las direcciones de almacenamiento interno.

La DATA DIVISION se utiliza, pues, para la descripción completa de los ficheros, de sus registros lógicos y de las zonas de maniobra o de trabajo.

Veamos un ejemplo muy simple que aclare los conceptos anteriores. Supongamos que deseamos, en una parte del programa, realizar la operación de multiplicar dos datos representados, respectivamente, por los nombres de datos CANTIDAD y PRECIO, a fin de obtener el resultado representado por el nombre de datos IMPORTE-TOTAL.

Una de las formas de realizarlos sería mediante una sentencia que utilice el verbo multiplicar.

MULTIPLY CANTIDAD BY PRECIO GIVING IMPORTE-TOTAL.

Con esta sentencia indicamos en el programa que se multiplique el valor del dato CANTIDAD por el valor del dato PRECIO y que se almacene el resultado del producto en una parte que denominamos IMPORTE-TOTAL.

Para que pueda realizarse todo lo anterior es necesario que se conozca:

- Que CANTIDAD y PRECIO son nombres de datos.
- Que IMPORTE-TOTAL es el resultado de una operación y que se desea almacenar en alguna parte de la memoria.
- En qué partes de la memoria se encuentran los valores que se van a multiplicar y en qué posición se va a almacenar el producto, a fin de poder acceder a ella cuando se desee conocer el resultado.

Para resolver esta cuestión, el programador DECLARA en la División de Datos del programa que CANTIDAD y PRECIO son DATOS cuyos valores están almacenados en un determinado soporte externo organizado en forma de fichero, así como sus características: longitud, utilización, clase, etc., y RESERVA una zona de trabajo o de maniobra en la memoria para que

se albergue el producto representado por IMPORTE-TOTAL. En cambio, no se preocupa en absoluto del posicionamiento que tendrá en la memoria, ya que esta labor la realiza el programa traductor de forma automática, en la fase de compilación, estableciendo una correspondencia entre cada nombre de dato y la dirección que va a ocupar en memoria. De esta forma, cada vez que los necesite en las diferentes partes del programa bastará con que los referencie con su nombre simbólico, con lo que el programa objeto interpretará no el nombre simbólico, sino la dirección real en la cual estará el dato o resultado simbolizado por dicho nombre.

La misión, pues, del programador consiste en exponer de una forma ordenada y estructurada la naturaleza de todos y cada uno de los datos que va a utilizar en el programa (ficheros, tablas, constantes, zonas de maniobra o de trabajo, etc.).

V-1.2. ESTRUCTURACION DE LOS DATOS EN COBOL

En un programa, la estructuración de los datos del problema está basada en la noción de *nivel*. En efecto, a cada uno de los datos se les asocia un valor comprendido entre 01 y 49, ambos inclusive, que se denomina indicador de nivel o número de nivel.

V-1.2.1. Los números de nivel

Los números de nivel se asocian a los nombres de datos en función de la categoría de éstos. Bajo el punto de vista de organización, los datos se clasifican en:

- Datos o campos elementales.
- Datos o campos de grupos o compuestos.
- Datos o campos independientes.

Un dato elemental es un dato que no puede subdividirse o que no es necesario subdividir para ser tratado por el programa.

Un dato de grupo es un dato que necesita ser subdividido en datos elementales, o en sentido contrario, es un conjunto de varios datos elementales o de varios datos de grupo.

Veamos la estructura (1) de un registro lógico que contenga información acerca de un empleado en una empresa.

JOSE	FERNANDEZ	ABOGADO	GENERALISIMO, 470 MADRID	2123456
------	-----------	---------	-----------------------------	---------

Este registro contiene información sobre las siguientes características del empleado:

- Nombre.
- Apellido.
- Profesión.
- Dirección.
- Número de teléfono.

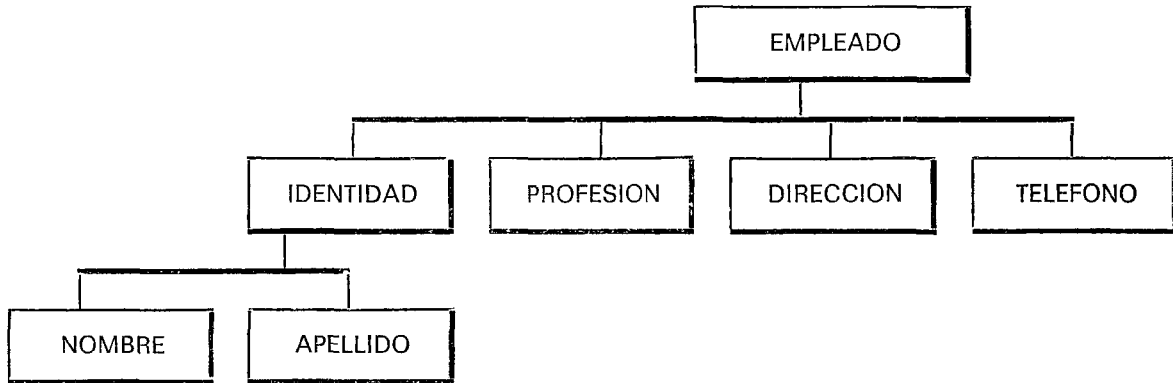
Cada una de estas informaciones pueden considerarse como datos elementales del registro, ya que no pueden subdividirse en otros más elementales. Sin embargo, los dos primeros datos podrían agruparse en un dato más amplio denominado Identidad, que sería en estas circunstancias un dato de grupo, de forma tal, que cada vez que se accede a él proporciona a su vez información sobre los datos elementales que lo componen.

(1) Las posiciones ocupadas por los datos en los registros o en la memoria central se denominan campos.

Cada uno de los valores de los distintos datos que componen el registro lógico ocupará una zona de la memoria de distinta longitud en función de las características definidas por cada uno de los datos.

A fin de indicar al ordenador qué espacio deberá reservar en la memoria para todos los elementos que componen el registro lógico, se podrían agrupar todos los datos elementales o de grupo que lo componen en otro grupo de mayor jerarquía que se podría denominar EMPLEADO.

Esquemáticamente podríamos representar de forma gráfica lo expuesto en el ejemplo anterior:



Vemos que existen jerárquicamente en el ejemplo expuesto, hasta tres niveles de información:

- Un primer nivel compuesto por un dato de grupo.
- Un segundo nivel compuesto por un dato de grupo y tres datos elementales.
- Un tercer nivel compuesto por dos datos elementales.

A los datos que componen cada uno de los niveles se les asigna un indicador de nivel, independientemente que se trate de un dato de grupo o de un dato elemental.

Al nivel más elevado que representa al registro lógico se le asigna el nivel 01, y a los distintos niveles un número de nivel cuyo valor deberá ser más elevado cuanto menor sea el nivel jerárquico.

Así, en el ejemplo expuesto, se podrían dar los siguientes valores:

EMPLEADO				NIVEL 01
IDENTIDAD	PROFESION	DIRECCION	TELEFONO	NIVEL 02
NOMBRE	APELLIDOS			NIVEL 03

V-1.2.2. Normas para la utilización de los números de nivel en datos elementales y de grupo

Las normas a seguir en la utilización de los números de nivel son las siguientes:

1) Para identificar cualquier campo de datos compuesto o elemental se deben utilizar cualquiera de los números de nivel 01 a 49, teniendo presente que:

- No es obligatorio que los números de nivel utilizados sean correlativos.
- Un mismo número de nivel puede servir tanto para identificar un campo compuesto como un campo elemental.
- Para números de nivel inferior a 10, debe colocarse el dígito cero a la izquierda del dígito significativo.

2) La identificación de la subordinación de un campo elemental a un campo compuesto está en función del número de nivel atribuido a cada uno de ellos, siendo el campo subordinado el que tiene un número de nivel mayor, debiendo tenerse presente que el ordenador reconoce que un campo de número de nivel X está subordinado a campo de grupo de nivel Y, cuando entre los números de nivel X e Y se cumplen todas y cada una de las siguientes condiciones:

- a) El número de nivel X está descrito posteriormente a Y en la secuencia de escritura del programa.
- b) El número de nivel X es mayor en valor numérico que Y.
- c) Entre las inscripciones de X e Y no existe ningún otro número de nivel inferior a Y.

Ejemplo: Supongamos un registro compuesto de los siguientes campos:

VENTAS, NUMERO-CREDITO, TIPO, ORDEN, TIPO-VENTA, TRANSACION, CLIENTE, CODIGO, NOMBRE, VENTA-EFFECTUADA, IMPORTE, FECHA.

Si asignamos los siguientes números de nivel:

NOMBRE	N.º DE NIVEL ASIGNADO
VENTAS	01
NUMERO-CREDITO	02
TIPO	03
NUMERO-ORDEN	03
TIPO-VENTA	02
TRANSACION	02
CLIENTE	03
CODIGO	04
NOMBRE	04
VENTA-EFFECTUADA	03
IMPORTE	04
FECHA	04

Podemos deducir que:

VENTAS: Es un dato de grupo, representativo del registro lógico, integrado por:

- NUMERO-CREDITO
- TIPO-VENTA
- TRANSACION

NUMERO DE CREDITO: Es un dato de grupo integrado por:

- TIPO
- NUMERO-ORDEN

TIPO-VENTA: Es un dato elemental.

TRANSACION: Es un dato de grupo compuesto de:

- CLIENTE
- VENTA-EFECTUADA

TIPO: Es un dato elemental.

NUMERO-ORDEN: Es un dato elemental.

CLIENTE: Es un dato de grupo compuesto de:

- CODIGO
- NOMBRE

VENTA-EFECTUADA: Es un dato de grupo compuesto de:

- IMPORTE
- FECHA

CODIGO, NOMBRE, IMPORTE y FECHA: Son datos elementales.

También podemos deducir observando la definición efectuada, en sentido contrario que, por ejemplo:

CODIGO es un dato elemental que no está subordinado al campo de NUMERO-CREDITO, ya que aunque se cumplen las reglas 2a y 2b no se cumple la regla 2c.

Se recomienda al lector que con los datos expuestos represente esquemáticamente el registro definido en el ejemplo.

De esta forma se podrá observar que la organización de la información a tratar no es otra cosa que una representación arborescente. El grupo forma un árbol con diversas ramificaciones. Y por consiguiente, podemos deducir que en la representación arborescente de un grupo, **un elemento** está siempre situado en la extremidad de una ramificación de la cual no parte ninguna otra ramificación.

V-1.2.3. Números de nivel especiales

Además de los números de nivel ya citados se pueden emplear otros tres números de nivel especiales, que son 66, 77 y 88. Estos números tienen una aplicación muy específica.

V-1.2.3.1. Número de nivel 66 o número de reagrupamiento

En primer lugar, es importante indicar que este número de nivel no puede utilizarse en todos los traductores. En el caso de poderse utilizar, su objetivo es dar nombre de grupo a uno o varios elementos consecutivos, a los que ya se ha asignado número de nivel. El motivo fundamental de esta forma de proceder es el de modificar la jerarquía de los números de nivel existentes.

En el ejemplo expuesto en el epígrafe V-1.2.2 se puede asignar, después de la definición efectuada, uno o varios números de nivel 66 de la siguiente forma:

Nombre que se va a dar al grupo	La palabra reserva RENAMES
Primer elemento del conjunto a reagrupar	La palabra reservada
THRU	Ultimo elemento del conjunto a reagrupar;

Ejemplos:

					N.º NIVEL
AGRUPACION-1	RENAMES	NUM.-CREDITO	THRU	TIPO-VENTA	66
AGRUPACION-2	RENAMES	CLIENTE	THRU	IMPORTE	66

En los ejemplos anteriores, el número de nivel 66 permite crear un grupo que denominamos AGRUPACION-1, que contiene a los grupos NUMERO-CREDITO y TIPO-VENTA, y el grupo que denominamos AGRUPACION-2, que contiene al grupo CLIENTE completo y al elemento IMPORTE.

También como aplicación muy específica se puede utilizar para dar dos nombres distintos a un mismo elemento. En este caso basta eliminar la parte final del formato anteriormente indicado (desde THRU).

Tomando como base el ejemplo ya utilizado.

CANTIDAD RENAMES IMPORTE, con número de nivel 66, permite emplear los nombres CANTIDAD e IMPORTE para el mismo elemento del registro.

V-1.2.3.2. Número de nivel 77 o número de nivel independiente

A veces es necesario utilizar datos o zonas de trabajo que no están compuestos de varias partes o que no forman parte de un conjunto mayor. Estos datos no entran, pues, en la jerarquía ya mencionada de grupos y elementos. Estos datos se denominan *datos independientes*.

Para definirlos en el programa se emplea el número de nivel 77.

V-1.2.3.3. Número de nivel 88 o número de nivel de los nombres de condición

Este número de nivel se utiliza para asignar nivel a cada uno de los nombres dados a los valores de un nombre de datos.

Así, por ejemplo, si el nombre de datos COLOR tiene cinco valores, del 1 al 5, a los que se ha asignado los nombres AMARILLO, VERDE, AZUL, ROJO, NEGRO, respectivamente, es necesario asignar a cada uno de ellos el número de nivel 88 detrás de la inscripción correspondiente al nombre de datos, independientemente del número de nivel que posea.

	NUMERO DE NIVEL
COLOR	03
AMARILLO	88
VERDE	88
AZUL	88
ROJO	88
NEGRO	88

V-1.2.3.4. *Indicadores de nivel de los ficheros.*

Antes de describir las características de los registros que componen un fichero se debe efectuar la descripción de las características del mismo, utilizándose como indicador el número especial FD (iniciales de FILE DESCRIPTION, descripción de ficheros) o el número de nivel RD (iniciales de REPORT DESCRIPTION) cuando el fichero es la impresora y se describe, además, en la sección de informes.

V-1.2.4. **Reglas de escritura de los números de nivel en la hoja de codificación**

a) Los números de nivel deben escribirse sobre la hoja de codificación a partir de la columna 8 y hasta la columna 11 inclusive. Algunos compiladores admiten que se describan los números de nivel, de forma tal que puedan escribirse sobrepasando el margen B. En otros compiladores, por el contrario, está impuesta su escritura a partir de la columna 12.

b) Para la descripción de un fichero las letras FD y RD de los indicadores de nivel FD y RD, respectivamente, deben figurarse obligatoriamente a partir del margen A.

c) A continuación del indicador de nivel de cada nombre se debe escribir, dejando al menos un espacio libre el nombre del dato correspondiente.

d) Es recomendable, aunque no obligatorio, alinear los nombres de datos asociados a los números de nivel del mismo valor y decalar hacia la derecha los números de nivel de los grupos contenidos en otros grupos y la de los datos elementales en relación a los grupos a los que pertenecen.

e) Los números de nivel 77 no pueden utilizarse en la descripción de ficheros, ya que están reservados exclusivamente en la descripción de zonas de trabajo o maniobra.

f) Los números de nivel 77 deben ser los primeros a describir en la zona de trabajo o maniobra.

g) Los números de nivel 77 pueden describirse a partir de la columna 8.

• *Ejemplo de escritura de los ejemplos expuestos en los párrafos anteriores.*

Página	Línea	Cont	A	B																	
1	3	4	6	7	8	12	15	20	25	30	35	40	45								
	010					[ZONA DE DESCRIPCION DE FICHEROS]															
	020					FD	FICHERO-	VENTAS	~~~~~												
	030					01	VENTAS.														
	040					02	NUM-CREDITO.														
	050					03	TIPO	~~~~~													
	060					03	NUM-ORDEN	~~~~~													
	070					02	TIPO-VENTA	~~~~~													
	080					02	TRANSACCION.														
	090					03	CLIENTE.														
	100					04	CODIGO	~~~~~													
	110					04	NOMBRE	~~~~~													
	120					03	VENTA-EFECTUADA.														
	130					04	IMPORTE	~~~~~													
	140					04	FECHA	~~~~~													
	150					66	AGRUPACION-1	RENAMES	NUM-CREDITO												
	160							THRU	TIPO-VENTA.												
	170					66	AGRUPACION-2	RENAMES	CLIENTE	THRU	IMP										
	180					[ZONA DE TRABAJO O MANIOBRA]															
	190					77	CONTADOR	~~~~~													
	200					01	COLOR	~~~~~													
						88	AMARILLO	~~~~~													
						88	AZUL	~~~~~													
						~~~~~															

V - 2

ORGANIZACION DE LA DATA  
DIVISION





## V-2. ORGANIZACION DE LA DATA DIVISION

### V-2.1. GENERALIDADES

La división de datos está estructurada en *secciones* y éstas en inscripciones o descripciones de datos.

El número de secciones varía con cada uno de los constructores.

Aunque el número máximo de secciones en que puede descomponerse la división de datos es de seis, ninguno de los fabricantes las utiliza todas, pues el caso de máximo fraccionamiento es de 5. La explicación de esta circunstancia es que en determinados compiladores, algunas de las secciones que se desglosan en otros, están en él agrupadas.

Sin embargo existen siempre para todos los compiladores dos secciones obligatorias y comunes.

- FILE SECTION (Sección de ficheros).
- WORKING-STORAGE SECTION (Sección de almacenamiento de trabajo).

Las cuatro secciones restantes se denominan:

- COMMON-STORAGE SECTION (Sección común de almacenamiento).
- CONSTANT SECTION (Sección de constantes).
- LINKAGE SECTION (Sección de unión).
- REPORT SECTION (Sección de informes).

Cada una de estas secciones puede formar parte de los compiladores más actuales existentes en el mercado de ordenadores.

La inexistencia de alguna sección puede obligar a la existencia de otra, como ocurre con la COMMON-STORAGE SECTION y la LINKAGE SECTION.

Cuando una sección no está incluida en un compilador, los datos que en ella deberán definirse se incluyen en otras de las secciones.

El número de las secciones que se emplean en cada uno de los compiladores actuales en el mercado, así como su orden de escritura dentro de la DATA DIVISION, se detalla en el epígrafe siguiente.

### V-2.2. ORDEN Y NUMERO DE SECCIONES DE LA DATA DIVISION EN LOS PRINCIPALES COMPILADORES

El orden en que deben aparecer cuando se necesitan las diversas secciones, así como las secciones que pueden contener cada uno de los compiladores de los sistemas figurados, se describe en el cuadro siguiente:

SECCION	CDC (Serie 6000)	IBM (Series 360 y 370)	UNIVAC (Serie 1108)	UNIDATA-CII (Serie Iris 50)	UNIDATA-HWB (Serie GE 600)	UNIDATA-SIEMENS (Serie 4004)	UNIVAC (Series 9200-9300)	UNIDATA-HWB (Serie 100)
FILE SECTION	1. ^a	1. ^a	1. ^a	1. ^a	1. ^a	1. ^a	1. ^a	1. ^a
WORKING-STORAGE SECTION	3. ^a	2. ^a	2. ^a	2. ^a	2. ^a	2. ^a	2. ^a	2. ^a
CONSTANT SECTION	4. ^a		3. ^a					
COMMON-STORAGE SECTION	2. ^a		4. ^a					
LINKAGE SECTION		3. ^a		3. ^a		3. ^a	3. ^a	3. ^a
REPORT SECTION	5. ^a	4. ^a			3. ^a	4. ^a		4. ^a

### V-2.3. OBJETIVO DE CADA UNA DE LAS SECCIONES DE LA DATA DIVISION

- FILE SECTION: Se emplea siempre que se utilicen ficheros en el programa; sirve para definir las características de los mismos y las de cada uno de los registros y datos que lo componen. Puede observarse que siempre figura en primer lugar en todos los compiladores.
- WORKING-STORAGE SECTION: Se utiliza para definir las zonas de trabajo o de maniobra en la memoria central.
- COMMON-STORAGE SECTION: Se utiliza para definir dentro de la memoria central las zonas que se van a utilizar para enlazar un programa principal en otros programas subordinados escritos en COBOL o en otros lenguajes.
- LINKAGE SECTION: Tiene la misma utilización que la COMMON-STORAGE SECTION, aunque con una organización interna distinta.
- CONSTANT SECTION: Se utiliza para albergar los datos de un problema que mantienen un valor constante, a lo largo del desarrollo del mismo. Normalmente está formado únicamente por datos con números de nivel 77. Los compiladores que no poseen esta sección utilizan la WORKING-STORAGE SECTION para los mismos menesteres.
- REPORT SECTION: Esta sección se utiliza frecuentemente en los compiladores con la versión ANSI (1); como su nombre indica, se emplea fundamentalmente en programas en los que se desea obtener resultados impresos. Puede observarse en el cuadro anterior que, cuando existe, figura siempre en último lugar.

(1) Recientemente, la American National Standards Institute (ANSI), formalmente denominada USA Standard Institute (USASI), fue responsabilizada en establecer un American National Standard COBOL, denominado abreviadamente ANSI COBOL. Reservamos un apéndice de esta obra para comentar las particularidades de esta versión.

V - 3

SECCION DE FICHEROS



### V-3. SECCION DE FICHEROS

#### V-3.1. GENERALIDADES

Esta sección está reservada para la descripción de todos los ficheros utilizados en el programa. Suministra información sobre los registros físicos y lógicos del fichero.

Como es sabido, los ficheros contienen además de los propios registros de datos, otros registros especiales denominados ETIQUETAS. La descripción de la naturaleza de las mismas se realiza también en esta sección.

Todos los ficheros que se hayan definido en la ENVIRONMENT DIVISION a través de las cláusulas SELECT del párrafo FILE-CONTROL deberán ser descritos en esta sección.

Tomemos como ejemplo una descripción de la ENVIRONMENT DIVISION de un programa escrito para el sistema UNIVAC 1108: FILE-CONTROL.

```
SELECT FICHERO-A ASSIGN TO CARD-READER-EIGHTY  
SELECT IMPRESORA ASSIGN TO PRINTER.
```

Para cada SELECT descrita será necesario efectuar dentro de la FILE SECTION una descripción del fichero mencionado; en el ejemplo FICHERO-A e IMPRESORA.

La FILE SECTION describe las áreas de entrada y salida utilizadas en el programa. Un área de entrada es el almacenamiento reservado para un fichero que se va a leer o fichero de entrada. Análogamente, un área de salida es un almacenamiento reservado para un fichero que se va a crear o fichero de salida. Los dispositivos a utilizar y los nombres de los ficheros se asignan en la ENVIRONMENT DIVISION en una cláusula SELECT.

En la FILE SECTION se describe cada fichero con una inscripción de número de nivel FD. Cada inscripción FD estará seguida del nombre del fichero y un determinado número de cláusulas.

Tanto el nombre de la sección como el indicador de nivel FD deben codificarse en el margen A de la hoja de codificación.

#### V-3.2. ESTRUCTURA DE LA DESCRIPCION DE UN FICHERO

El formato general de la descripción de cada fichero en la FILE SECTION es:

A

B

FD

Nombre-fichero

[ **RECORDING MODE IS** {opción-1} ][ **BLOCK CONTAINS** número-entero {opción-2} ][ **RECORD CONTAINS** [ número-entero-1 **TO** ] número-entero-2 CHARACTERS ][ **LABEL RECORD** [S] [ ARE ] {opción-3} ][ **DATA RECORD** [S] [ ARE ] nombre-registro-1 [ nombre-registro-2 ] . . . . . ]

Descripción de los registros lógicos del fichero

## V-3.3. CLAUSULA RECORDING MODE

FORMATO

[ **RECORDING MODE IS** {opción-1} ]

OBJETO

La misión de esta cláusula varía sustancialmente en función del tipo de compilador usado. Por consiguiente, la **opción 1** delimita el uso de la misma. Las posibles opciones a elegir, dependiendo de los requisitos, son las siguientes:

F; V; U, BLANK, SIGN, XS3, BINARY, BCD, DECIMAL

OPCIONES

En el cuadro siguiente se resume la correspondencia entre las opciones a elegir para cada una de las distintas compilaciones.

OPCION	CDC (Serie 6000)	IBM (Series 360 y 370)	UNIVAC (Serie 1108)	UNIDATA-CII (Serie IRIS 50)	UNIDATA-HWB (Serie GE 600)	UNIDATA-SIEMENS (Serie 4004)	UNIVAC (Series 9200-9300)	UNIDATA-HWB (Serie 150)
F		*		*		*	*	
U		*		*		*		
V		*		*		*	*	
BLANK			*					
SIGN			*					
XS3			*					
BINARY	*				*			*
BCD					*			*
DECIMAL	*							

* Opción válida para el sistema indicado.

OPCIONES	COMETIDO
F	Indicar que los registros de un fichero son de longitud fija.
U	Indicar que los registros de un fichero son de longitud indefinida.
V	Indicar que los registros de un fichero son de longitud variable.
BLANK	Indicar que los blancos no significativos de las zonas numéricas de los ficheros de entrada son automáticamente sustituidos por ceros en el área de entrada.
SIGN	Indicar que los campos numéricos de los registros de los ficheros de entrada que tienen atribuido signo, éste está superpuesto al carácter extremo derecho y que será transferido al carácter extremo izquierdo del campo correspondiente al área de entrada.
XS3	Se utiliza exclusivamente en soportes en cinta magnética con un formato especial.
BINARY	Cuando la información se registra codificada en el sistema binario.
BCD o DECIMAL	Cuando la información se registra en decimal codificada en binario.

Para las opciones **BINARY** y **BCD o DECIMAL** y cuando el soporte es una cinta magnética, se puede añadir a la opción las palabras reservadas **HIGH** o **LOW**, dependiendo de que la densidad de grabación sea alta o baja, es decir, de que exista mayor o menor número de bits por pulgada de cinta magnética grabada o leída.

### V-3.4. CLAUSULA BLOCK CONTAINS

**FORMATO**

**BLOCK CONTAINS** número-entero {opción-2}

**OBJETO**

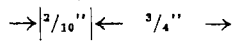
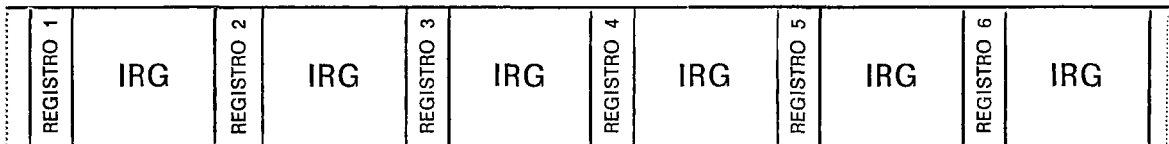
Esta cláusula se incluye solamente en la inscripción FD cuando los ficheros son cintas o discos magnéticos.

Las cintas o discos magnéticos tienen normalmente bloqueados los registros para aumentar el rendimiento de las operaciones de entrada-salida.

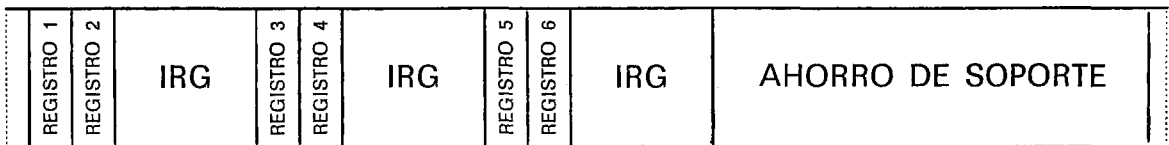
Agrupar físicamente los registros de un fichero en BLOQUES de información tiene dos objetivos fundamentales:

- Aprovechar al máximo los carretes de cinta magnética o las pistas de los discos magnéticos.
- Optimizar los tiempos de lectura y grabación empleados.

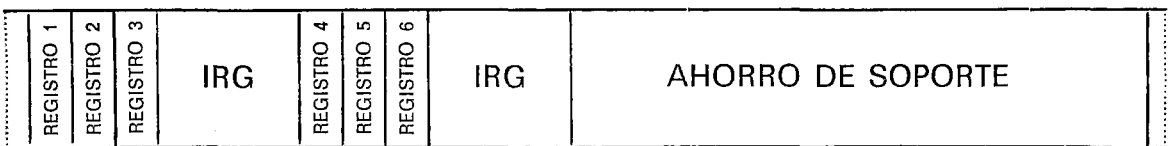
Para entender con mayor claridad estos dos objetivos, obsérvese la figura siguiente.



BLOQUES DE UN REGISTRO



BLOQUES DE DOS REGISTROS



BLOQUES DE TRES REGISTROS



La figura muestra en esquema y a escala aproximada la imagen que tendría un mismo fichero que contuviese seis registros de 160 caracteres cada uno, bloqueados, en tres supuestos distintos.

1. Agrupación de registros de uno en uno o factor de bloqueo = 1.
2. Agrupación de registros de dos en dos o factor de bloqueo = 2.
3. Agrupación de registros de tres en tres o factor de bloqueo = 3.

Teniendo en cuenta que la longitud de un IRG (Inter Record Gap o espacio entre bloques) es de  $\frac{3}{4}$  de pulgada y suponiendo una unidad de grabación de 800 bytes por pulgada, tendremos:

LONGITUD OCUPADA en el primer supuesto:  
 $6 (0,2 + 0,75) = 5,7''$ .

LONGITUD OCUPADA en el segundo supuesto:  
 $6 \times 0,2 + 3 \times 0,75 = 3,45''$ .

LONGITUD OCUPADA en el tercer supuesto:  
 $6 \times 0,2 + 2 \times 0,75 = 2,7''$ .

En términos relativos, en el tercer supuesto existe un ahorro del 52 por 100 de espacio de soporte con relación al primero, o, expresado de otra forma, la misma información puede almacenarse en la mitad de espacio.

A medida que aumenta el factor de bloqueo aumenta el aprovechamiento de espacio. En contrapartida, la memoria interna necesaria para almacenar la información leída aumentará, ya que ésta se almacena por bloque de información y no por registros lógicos individuales.

El otro objetivo está íntimamente relacionado con el primero, ya que al aumentar el bloqueo disminuye el tiempo de lectura o de grabación. A igualdad de registros en un fichero, el número de interrupciones en la lectura coincide con el número de IRG, ya que los dispositivos de entrada o salida leen o graban entre cada dos IRG consecutivos. Como a mayor factor de bloqueo existen menor número de IRG, habrá menor número de interrupciones. Estas interrupciones son las que motivan la existencia de los IRG con objeto de que la cinta frene y vuelva a arrancar con la suficiente holgura como para no perder información, invirtiendo en las operaciones de frenado y arranque tiempos infinitamente mayores que los empleados en leer o grabar, sobre todo con bloques pequeños.

#### OPCIONES

La opción que acompaña al formato de esta cláusula sirve para indicar si el agrupamiento de registros se expresa en número de caracteres o en número de registros lógicos.

En el primer caso se emplea la palabra reservada de CHARACTERS, y en la segunda, RECORDS.

#### V-3.5. CLAUSULA RECORD CONTAINS

#### FORMATO

[**RECORD CONTAINS** [número-entero-1 **TO**] número-entero-2 **CHARACTERS**]

La cláusula RECORD CONTAINS indica el tamaño de cada registro. Un fichero que se va a imprimir deberá tener la siguiente descripción:

RECORD CONTAINS 132 CHARACTERS

En este caso, número-entero-2 tiene el valor 132 y no es necesario emplear la parte optativa de la cláusula.

Un fichero en fichas tendrá la siguiente inscripción:

RECORD CONTAINS 80 CHARACTERS

Es decir, número-entero-2 tendrá el valor 80 y tampoco será necesario utilizar la parte optativa.

**OPCIONES**

Sin embargo, para cintas o discos deberá emplearse la parte opcional (número-entero-1 TO) cuando el tamaño del registro sea variable; *número-entero-1* se utiliza para indicar el tamaño del registro de menor longitud y *número-entero-2* para el de mayor longitud.

Esta cláusula, aun siendo opcional, es de especial interés, ya que permite obtener un chequeo del tamaño del registro entre el valor dado en la descripción efectuada y el valor obtenido como suma de los campos componentes del registro en la descripción de estos últimos.

V-3.6. CLAUSULA LABEL RECORD

**FORMATO**

**LABEL RECORD** [S] [  $\begin{matrix} ARE \\ IS \end{matrix}$  ] {opción-3}

**OBJETO**

Las cintas y los discos no pueden ser leídos o grabados de la misma forma que se leen o perforan los ficheros en fichas, debido a que la información contenida en los ficheros magnéticos no es reconocible por parte de los operadores, tan directamente como pueden serlo los ficheros en fichas. Por ello es necesario crear en los ficheros sobre soportes magnéticos, una información adicional a la contenida en los registros de datos que sirve como elemento identificador. Los registros de etiquetas, o simplemente etiquetas, son unos registros especiales situados al principio y al final de cada fichero magnético.

Estos registros se crean cuando se generan ficheros de salida o se leen cuando se trata de un fichero de entrada.

Existen tres posibilidades:

- Omitir esta identificación.
- Crear etiquetas estandarizadas de formato común para todos los ficheros y generadas por el propio sistema.
- Crear etiquetas con formato especial para ficheros determinados.

**OPCIONES**

Cada una de estas opciones se recoge en las cláusulas LABEL RECORD, con la inclusión de una de las opciones:

- OMITTED, para omisión de etiquetas.
- STANDARD, para etiquetas tipo estándar.
- *Nombre-de-dato*, para etiquetas con formato especial.

La elección de una de las opciones dependen del nivel de seguridad en el tratamiento del fichero al que se aplique la tiqueta.

Por ejemplo, en ficheros de trabajo, que se crean y destruyen en un proceso determinado, no será necesario dotarle de etiquetas. En cambio, será necesario crear etiquetas, incluso especiales, en aquellos ficheros en los que, por motivos de seguridad o confidencialidad, su acceso quede restringido únicamente a los programadores que tengan conocimiento perfecto del contenido de la etiqueta.

V-3.7. CLAUSULA DATA RECORD

**FORMATO**

**DATA RECORD** [S] [ ARE ] nombre-dato-1  
                                  [ IS ] [nombre-dato-2]...

**OBJETO**

Esta cláusula permite dar uno o varios nombres a los registros lógicos de un fichero. Cuando solamente esté presente nombre-dato-1, que representa al nombre del registro, todos los registros lógicos del fichero tendrán el mismo nombre, aunque esto no signifique que todos sean de igual longitud.

Cuando existan varios nombre-dato-1 significará que todos los registros lógicos del fichero no contienen el mismo tipo de información.

Para cada uno de los nombre-dato que se figuren en la cláusula será necesario efectuar una descripción de registro a nivel 01.

V-3.8. INSCRIPCIONES DE DESCRIPCION DE REGISTROS

Como se ha visto en la cláusula DATA RECORD, descrita anteriormente, para cada tipo de registro contenido en un fichero se atribuye un nombre asociado al mismo; existirán, por tanto, tantos nombres de registros como tipos distintos existan.

Para cada tipo es necesario efectuar una descripción de los elementos que lo componen.

La descripción de un registro se realiza mediante una inscripción 01 y tantas inscripciones de nivel 02 a 49 como sean necesarias.

Cada una de las inscripciones, independientemente del número de nivel que posean, se componen de tres partes:

1.ª	NUMERO DE NIVEL
2.ª	NOMBRE-DATO (nombre dado al registro o nombre dado a los elementos que lo componen) o la palabra reservada FILLER.
3.ª	UNA O VARIAS CLAUSULAS DE DESCRIPCION.

Existen dos categorías entre las cláusulas de descripción:

**GENERALES:** Son las relativas a la descripción de los registros lógicos de los ficheros de entrada o de salida no impresos.

Aportan la información de las características de los datos que van a ser tratados en el programa.

**DE EDICIÓN:** Son las relativas a la descripción de los registros lógicos de los ficheros de salida impresos.

Contienen la información de las características de los resultados que van a ser obtenidos con el programa.

#### V-3.8.1. Cláusulas generales de las inscripciones de descripción de registros

Las cláusulas de tipo general que pueden utilizarse en una inscripción de un registro son las contenidas en el cuadro siguiente.

CLAUSULA	FORMATO
REDEFINES	[ <u>REDEFINES</u> nombre-dato ]
SIZE	[ <u>SIZE</u> IS número-entero [opción] ]
CLASS	[ [ <u>CLASS</u> IS] {opción} ]
USAGE	[ [ <u>USAGE</u> IS] {opción} ]
SIGNED	[ <u>SIGNED</u> ]
POINT	[ <u>POINT</u> LOCATION IS {opción} número-entero PLACES ]
PICTURE	[ <u>PICTURE</u> IS {opción = serie caracteres} ]
OCCURS	[ <u>OCCURS</u> [número-entero-1 TO] número-entero-2 TIMES [DEPENDING ON nombre-dato] ]
VALUE	Ver formato en el epígrafe correspondiente a esta cláusula.

Cada una de las cláusulas contenidas en el cuadro anterior se utiliza aisladamente o en conjunción con otras, para describir una característica determinada del registro o de los campos que lo componen.



- No puede utilizarse en una inscripción con número de nivel especial.
- Cuando se utilice para redefinir un grupo, esta cláusula debe estar colocada inmediatamente detrás del último elemento del grupo considerado.

#### V-3.8.1.2. Cláusula *SIZE* o de tamaño

[ **SIZE** IS número-entero [opción] ]

**OBJETO**

Esta cláusula se utiliza para especificar el número de caracteres o tamaño, que tienen un campo considerado sobre su soporte externo y consecuentemente la dimensión de la zona de memoria en el área de entrada o de salida que lo contenga.

**NÚMERO ENTERO:** Indica el dimensionamiento. Por consiguiente, debe tratarse de un valor entero distinto de cero y sin signo.

**OPCIONES**

Las dos posibles opciones son:

DIGITS  
CHARACTERS

La primera se utiliza cuando el contenido del campo sea numérico, y la segunda cuando el campo considerado deba tener una información alfanumérica.

**PARTICULARIDADES DE SU USO**

- El punto decimal de un dato no entero no debe contabilizarse en la descripción de su tamaño, ya que no se encuentra almacenado junto al dato, ni en el soporte externo ni en la memoria central.
- En el caso de que el campo tenga signo, solamente se considera éste a efectos de tamaño cuando no esté superpuesto al último carácter.
- Esta cláusula no se utiliza en todos los compiladores.
- En algunos compiladores esta cláusula se denomina MAP (1).

#### V-3.8.1.3. Cláusula *CLASS* o de naturaleza del campo

[ [ **CLASS** IS ] {opción} ]

**OBJETO**

Esta cláusula se utiliza para definir la naturaleza de los caracteres que componen un campo.

**OPCIONES**

Las opciones, una de las cuales debe elegirse obligatoriamente en la cláusula, son tres:

(1) En el compilador de los sistemas 9200II/9300II de UNIVAC.

**ALPHABETIC:** Cuando el campo definido contiene únicamente caracteres alfabéticos o espacios.

**NUMERIC:** Cuando el campo definido contiene exclusivamente caracteres numéricos. Sin embargo, cuando el campo está dotado de signo se admite como primer carácter del campo el signo + o -, o como último carácter uno no numérico (2).

**ALPHANUMERIC o AN:** Cuando el campo definido puede contener cualquier tipo de caracteres numéricos, alfabéticos o especiales.

**PARTICULARIDADES  
DE SU USO**

- Esta cláusula puede utilizarse a cualquier nivel, incluso simultáneamente, en la descripción de un campo de grupo y de los campos elementales que lo forman, siempre y cuando las descripciones correspondientes no se contradigan.
- La tercera opción, en su forma abreviada AN, no es utilizable en todos los compiladores.
- Esta cláusula no existe en todos los compiladores.
- Cuando se está ejecutando un programa, las informaciones enviadas por los distintos campos deben corresponder a su naturaleza de forma tal que en la mayoría de los compiladores, cuando esto no ocurre, se emite un mensaje de error.
- Cuando sea necesario enviar a la misma zona información de naturaleza diferente debe hacerse uso de la cláusula REDEFINES.
- La palabra CLASS puede, si se desea, eliminarse de la escritura de la cláusula.

V-3.8.1.4. *Cláusula USAGE o de utilización*

**FORMATO**

[ **USAGE IS** ] {opción} ]

**OBJETO**

Esta cláusula permite definir el uso o utilización que se va a hacer de la información asociada a esta cláusula durante la ejecución del programa.

Como ya indicamos anteriormente, el programador de COBOL no tiene que preocuparse, si no lo desea, de la forma en que se almacenan los datos en la memoria interna. Sin embargo muchas veces es conveniente y necesario (3) decidir la forma más idónea en que debe realizarse dicho almacenamiento, ya que el aprovechamiento de memoria está en relación directa con la mayor o menor utilización de bits para expresar un dato. En el supuesto de ordenadores que empleen octetos como unidad básica, los caracteres numéricos pueden representarse de diversas formas, cuyo consumo de bytes variará con la opción elegida.

(2) Siempre que la combinación signo-carácter dé una configuración de dígito existente entre los caracteres representables en la máquina.

(3) Cuando existen problemas de tamaño de memoria disponible y en todas las circunstancias para optimizar tiempos de proceso.

**OPCIONES**

Las opciones que pueden incluirse en esta cláusula son las siguientes:

DISPLAY

DISPLAY-n

COMPUTATIONAL

COMPUTATIONAL-n

El empleo de cada una de estas opciones varía sensiblemente con cada uno de los sistemas existentes en el mercado.

En el anexo C se detallan para cada uno de los sistemas indicados las características peculiares de las distintas opciones de esta cláusula.

En términos más generales, la utilización de las diversas opciones de esta cláusula es la siguiente:

**DISPLAY:** Se emplea para indicar que la representación del elemento definido en la memoria central se realiza en caracteres y que normalmente no interviene en operaciones aritméticas.

**COMPUTATIONAL:** Se emplea para indicar que los elementos descritos van a utilizarse fundamentalmente en cálculos aritméticos.

**PARTICULARIDADES DE SU USO**

- Cuando se omite la escritura de esta cláusula se supone implícita la opción **DISPLAY**.
- No existen todas las opciones mencionadas en los distintos compiladores.
- Si se especifica la opción **COMPUTATIONAL**, el campo descrito ha de ser obligatoriamente numérico, y por consiguiente puede omitirse la cláusula **CLASS**, o en el caso de utilizarse debe figurar con la opción **NUMERIC**.

V-3.8.1.5. *Cláusula SIGNED o indicadora de signo*

**FORMATO**

[ **SIGNED** ]

**OBJETO**

Se utiliza exclusivamente para indicar que el campo que describe contendrá un valor numérico con signo.

**OPCIONES**

Esta cláusula no comporta el uso de ninguna opción.

**PARTICULARIDADES DE SU USO**

- No puede utilizarse más que en la descripción de campos elementales.
- Cuando se utiliza esta cláusula puede omitirse el uso de la cláusula **CLASS**, ya que presupone la opción **NUMERIC** de la misma.
- En los campos elementales numéricos que no empleen en su descripción esta cláusula, el programa presupone, en el curso de su ejecución, que se trata de un campo positivo.
- Esta cláusula no existe en todos los compiladores.



V-3.8.1.6. Cláusula POINT o indicadora de punto decimal

**FORMATO**

[**POINT** LOCATION IS {opción} número-entero PLACES]

**OBJETO**

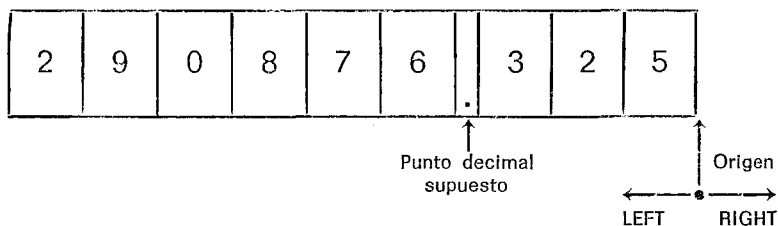
Esta cláusula se utiliza para indicar la ubicación del punto decimal supuesto en un campo numérico descrito en caracteres.

Ya hemos indicado anteriormente que los puntos decimales de los números decimales no se almacenan ni en los soportes externos ni en la memoria central. Sin embargo, por razones obvias, es totalmente necesario conocer el emplazamiento en donde debiera estar situado, con objeto de poder conocer el verdadero valor numérico de los resultados, en los que intervienen dichos campos como elementos de cálculo.

**OPCIONES**

Las opciones que pueden elegirse en el formato de esta cláusula son:

- LEFT (izquierda).
- RIGHT (derecha).



La opción LEFT indica la existencia del punto decimal en el campo al que se refiere la cláusula.

La opción RIGHT, en lugar de indicar la ubicación del punto decimal, sirve para suponer adicionados, sin estar almacenados, tantos ceros como se indique con el valor de número-entero.

**EJEMPLO**

Si se incluye la cláusula, POINT LOCATION IS RIGHT 2 PLACES, entre las inscripciones de un campo y el contenido del mismo es 156 el programa a efectos de cálculo, tratará el dato con valor 15600.

**PARTICULARIDADES DE SU USO**

- La omisión de esta cláusula implica que el valor numérico es un valor entero.
- La opción LEFT es equivalente a dividir el valor numérico por  $10^n$ .
- La opción RIGHT es equivalente a multiplicar el valor por  $10^n$ , siendo en este caso y en el anterior n el valor del número-entero indicado en la cláusula.

V-3.8.1.7. Cláusula PICTURE o modelo

**FORMATO**

[**PICTURE** IS {opción = serie de caracteres} ]

**OBJETO**

La cláusula PICTURE tiene una función múltiple. En este epígrafe vamos a estudiar exclusivamente su enfoque bajo el punto de vista de

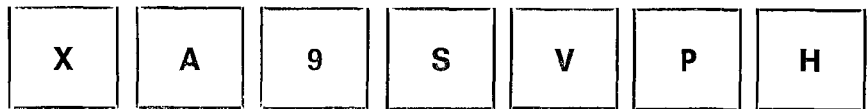
cláusula general o descriptiva de ficheros de entrada. En epígrafes posteriores se estudiarán las distintas posibilidades de esta cláusula.

Con esta limitación la función de la cláusula PICTURE engloba las funciones de áreas de las cláusulas anteriormente estudiadas (CLASS, SIZE, POINT LOCATION y SIGNED).

**OPCIONES**

No existe una lista específica de opciones entre las que se pueda elegir la más adecuada para cada caso. Como se observa en el formato, la opción consiste en establecer una serie o secuencia de determinados caracteres, cada uno de los cuales indica una determinada característica del dato descrito y de la zona de memoria reservada para albergarlo.

Los caracteres que pueden emplearse dentro de la secuencia son los siguientes:



Las reglas que rigen la elección de cada uno de estos caracteres dentro de la secuencia son:

**NUMERO DE CARACTERES**

- El número de caracteres elegidos indicará el tamaño del campo descrito, con excepción de los caracteres V, P y S, es equivalente por consiguiente al número-entero de la cláusula SIZE.

X	→	Cada carácter X, que se elija, indica que la posición ocupada por este carácter describe un carácter alfanumérico. Equivale a la cláusula CLASS IS ALPHANUMERIC.
A	→	Cada carácter A elegido indica que la posición ocupada por dicho carácter describe un carácter alfabético. Equivale a la cláusula CLASS IS ALPHABETIC.
9	→	Cada carácter 9 elegido indica que la posición ocupada por dicho carácter describe un carácter numérico. Equivale a la cláusula CLASS IS NUMERIC.

**EJEMPLOS**

- PICTURE IS XXXXX será equivalente a las cláusulas:  
SIZE 5 CHARACTERS  
CLASS ALPHANUMERIC
- PICTURE IS AAAA será equivalente a las cláusulas:  
SIZE 4 CHARACTERS  
CLASS ALPHABETIC
- PICTURE IS 999999 será equivalente a las cláusulas:  
SIZE 6 DIGITS  
CLASS NUMERIC

**MEZCLA  
DE CARACTERES**

- Pueden elegirse en la secuencia de caracteres, cualesquiera de ellos mezclados entre sí.

PICTURE IS XXAA999 indicará que el campo descrito contiene siete caracteres, de los cuales los dos primeros son alfanuméricos, los dos siguientes alfabéticos y los tres últimos numéricos. Esta cláusula es equivalente a:

SIZE 7 CHARACTERS  
CLASS ALPHANUMERIC

**COMPATIBILIDADES**

- Los caracteres X y A solamente son compatibles entre sí y con el carácter 9 dentro de la misma secuencia.
- El carácter 9 puede compartirse con los caracteres S, V y P.

**S** →

El carácter S indica la presencia de un signo para el elemento descrito. Este carácter supone la existencia del signo superpuesto al carácter numérico extremo derecho, del campo. Equivale a la cláusula SIGNED.

**V** →

El carácter V indica el emplazamiento de un punto decimal supuesto, dentro del campo descrito. Es equivalente a la cláusula POINT LOCATION IS LEFT número-entero PLACES.

**P** →

El carácter P indica la existencia, supuesta y solamente a efectos de cálculos, de un número de ceros igual al número de caracteres P, a la derecha del campo descrito por la cláusula. Es equivalente a la cláusula POINT LOCATION IS RIGHT número-entero PLACES.

**EJEMPLOS**

a) PICTURE S9999

es equivalente a las cláusulas

SIZE 4 DIGITS  
CLASS NUMERIC  
SIGNED

b) PICTURE 999V99

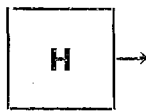
es equivalente a las cláusulas

SIZE 5 DIGITS  
CLASS NUMERIC  
POINT LOCATION IS LEFT 2 PLACES

c) PICTURE 999PPP

es equivalente a las cláusulas

SIZE 3 DIGITS  
CLASS NUMERIC  
POINT LOCATION IS RIGHT 3 PLACES



El carácter H tiene la misma función que la opción COMPUTATIONAL-3 de la cláusula USAGE. Solamente puede utilizarse, pues, para describir campos numéricos e indicar que el almacenamiento de los caracteres del campo descrito se realiza en formato decimal empaquetado (4).

**EJEMPLOS**

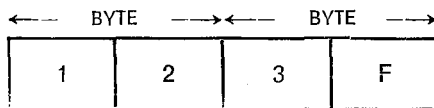
PICTURE H 999

es equivalente a las cláusulas

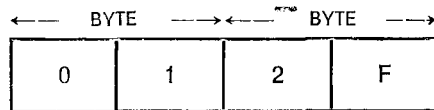
SIZE 3 DIGITS

USAGE COMPUTATIONAL-3

El almacenamiento sería en caso de que el campo descrito tuviese el contenido 123.



PICTURE H99, con contenido 12, tendría como almacenamiento:



El número de BYTES es igual a  $(\frac{\text{tamaño}}{2} + 1)$  bytes en caso de SIZE par y  $(\frac{\text{tamaño} + 1}{2})$  bytes en caso de SIZE impar.

- Los caracteres V y P no pueden figurar en la misma secuencia de caracteres.
- El número máximo de caracteres permitidos en una secuencia es de 30.
- Una serie de caracteres idénticos puede sustituirse por el número de ellos encerrado entre paréntesis.

XXXXX es equivalente a X(5)

999 es equivalente a 9(3)

De esta forma puede eliminarse la limitación anterior de 30 caracteres, ya que 31 veces, que no puede ser así descrito, puede describirse como X(31).

- Algunos compiladores admiten la palabra PIC en lugar de PICTURE.

(4) Decimal empaquetado significa almacenamiento de dígitos en medios octetos.

V-3.8.1.8. *Cláusula OCCURS o de repetición de un campo*

**FORMATO**

**OCCURS** [número-entero-1 **TO**!] número-entero-2  
**TIMES** [**DEPENDING** ON nombre-dato]

**OBJETO**

La cláusula OCCURS permite evitar la repetición de la descripción de varios campos elementales o de grupo, consecutivos e idénticos.

Quando varios campos consecutivos tienen las mismas e idénticas características, puede evitarse el repetir la descripción de cada uno de ellos, especificando las características del primero de ellos e incluyendo la cláusula OCCURS, en la que número-entero-2 indicará el número de repeticiones.

**EJEMPLO**

Supongamos un registro semanal de compras diarias:

CLIENTE	IMPORTE-COMPRAS					
	LUNES	MARTES	MIERC.	JUEVES	VIERNES	SABADO

Las características de los campos de IMPORTE-COMPRAS ... deben de tener, lógicamente, las mismas características; por ejemplo, tres enteros y dos decimales.

La descripción sería, utilizando cláusulas PICTURE:

- 02 CLIENTE PICTURE XXXX
- 02 COMPRAS-LUNES PICTURE 999V99
- 02 COMPRAS-MARTES PICTURE 999V99
- 02 COMPRAS-MIERCOLES PICTURE 999V99
- 02 COMPRAS-JUEVES PICTURE 999V99
- 02 COMPRAS-VIERNES PICTURE 999V99
- 02 COMPRAS-SABADO PICTURE 999V99

Utilizando la cláusula OCCURS quedaría la descripción reducida a

- 02 CLIENTE PICTURE XXXX
- 02 COMPRAS-DIA PICTURE 999V99 OCCURS 6 TIMES

El único inconveniente, como contrapartida, es la necesidad de tener que establecer un indicador que permita, cuando se traten estos datos en el programa, conocer a cuál de los seis se hace referencia.

Este problema lo trataremos en el estudio de subíndices.

**OPCIONES**

La única opción existente en esta cláusula es la opción DEPENDING, la cual permite describir campos elementales, o de grupo, de un registro de longitud variable. Los límites máximo y mínimo del factor de repetición viene dado por número-entero-2 y número-entero-1, respectivamente. El número de repeticiones del campo descrito en cada uno de los registros específicos del fichero se obtiene por el valor que nombre-dato tenga en el propio registro y que deberá, como es lógico, ser definido previamente.

**PARTICULARIDADES  
DE SU USO**

- En la descripción de un registro lógico pueden existir todas las inscripciones con cláusulas OCCURS que sean necesarias, con las siguientes limitaciones:
  - Si las inscripciones tienen el mismo número de nivel, no existe ninguna limitación en el número de cláusulas OCCURS.
  - Si las inscripciones de cláusulas OCCURS tienen números de nivel subordinados a otros campos, es decir, si se trata de cláusulas OCCURS en datos elementales o de grupo pertenecientes a otros grupos, se permite un número máximo de tres cláusulas OCCURS subordinados entre sí. El estudio de estas particularidades se verá con detalle en el epígrafe referente a subíndices.
- Esta cláusula no puede utilizarse en inscripciones con números de nivel 01 o números de nivel especiales.
- Cuando se emplea la opción **DEPENDING**, nombre-dato debe tener un contenido numérico entero positivo con un valor igual o menor a número-entero-2.

V-3.8.1.9. *Cláusula VALUE o de valor de los nombres de condición*

**FORMATO**

**VALUE** [**S**] [^{ARE}_{IS}] literal-1 [**THRU** literal-2] [literal-3  
[**THRU** literal-4] ... ]

**OBJETO**

La cláusula **VALUE**, como cláusula general de descripción de registros de ficheros, se utiliza para indicar los valores específicos de un nombre de dato, al que se atribuyen nombres de condición.

**OPCIONES**

Cuando el nombre de condición o los nombres de condición agrupan a varios valores de un campo, deberá utilizarse la opción u opciones **THRU** de la cláusula.

**EJEMPLOS**

Si el campo **COLOR** de un registro puede tener los valores 1, 2 y 3 y se desea atribuir un nombre de condición a cada uno de estos valores, se utilizarán tres cláusulas **VALUE**, además de la inscripción correspondiente a la definición del campo propiamente dicho.

02 **COLOR PICTURE 9**

88 **BLANCO VALUE 1S 1** (asigna al valor **1**, el nombre **BLANCO**)

88 **AZUL VALUE 1S 2** (asigna al valor **2**, el nombre **AZUL**)

88 **NEGRO VALUE 1S 3** (asigna al valor **3**, el nombre **NEGRO**)

En cambio, si se trata de un campo con muchos valores, a los que agrupadamente se desea asignar un nombre, deberá utilizarse la opción **THRU**.

Si el campo **MATERIAL** tiene los valores 1 a 300 y desea asignar a los valores 1 a 50 y 150 a 200 el nombre **TORNILLOS**, a los valores 51 a 100 y 201 a 250 el nombre de **TUERCAS** y a los valores 101 a 150 y 251 a 300 el nombre **CLAVOS**, se puede codificar esta descripción de la siguiente forma:

02 MATERIAL PICTURE 999

88 TORNILLOS VALUES ARE 001 THRU 050 151 THRU 200

88 TUERCAS VALUES ARE 051 THRU 100 201 THRU 250

88 CLAVOS VALUES ARE 101 THRU 150 251 THRU 300

**PARTICULARIDADES  
DE SU USO**

- Es obligatorio que el literal utilizado en la cláusula esté en concordancia con el tipo de elemento al cual está asociado.
- La opción THRU no existe en todos los compiladores.

**V-3.8.2. Cláusulas de edición de las inscripciones de descripción de registros**

Cualquiera de las cláusulas de tipo general estudiadas en los epígrafes anteriores puede utilizarse como cláusula de descripción para los ficheros de salida. Sin embargo, existen unas cláusulas específicas que únicamente pueden ser utilizadas en ficheros de salida denominadas cláusulas de edición.

Como cualquier soporte físico puede ser utilizado como fichero de salida estas cláusulas son aplicables a ficheros en cinta, disco, ficha perforada, etc.; sin embargo, se utilizan casi exclusivamente en ficheros de salida impresos, es decir, cuando el soporte físico es la impresora de líneas. La característica común de estas cláusulas es la de permitir que los resultados aparezcan sobre la impresora de forma clara y legible, añadiendo, suprimiendo o sustituyendo caracteres especiales que faciliten la lectura de aquéllos.

Las cláusulas específicas de edición son las contenidas en el cuadro siguiente.

CLAUSULA	FORMATO
BLANK	[ <u>BLANK</u> WHEN <u>ZERO</u> ]
JUSTIFIED	[ <u>JUSTIFIED</u> {opción} ]
ZERO	[ <u>ZERO SUPRESS</u> [ <u>LEAVING</u> número-entero PLACES] ]
CHECK	[ <u>CHECK PROTECT</u> [ <u>LEAVING</u> número-entero PLACES] ]
FLOAT	[ <u>FLOAT DOLLAR SIGN</u> [ <u>LEAVING</u> número-entero PLACES] ]
PICTURE	[ <u>PICTURE IS</u> {opción = serie caracteres} ]

La utilización de las cláusulas de edición está reservada exclusivamente para inscripciones de campos elementales. En la edición de un elemento, la transformación del contenido de la zona de memoria de

trabajo, denominada en estos casos zona emisora, tiene lugar en la transferencia a la zona de salida, denominada zona receptora.



La transformación de los datos de la zona emisora a la receptora se realiza en todos los casos de la siguiente forma:

- Transformación en caracteres, si es necesario (por ejemplo, cuando en la zona emisora esté almacenado el dato en forma binaria pura).
- Alineación de puntos decimales (supuestos a reales).
- Relleno de ceros si la zona receptora es de mayor tamaño que la emisora.
- Truncamiento de caracteres si la zona receptora es menor que la emisora.
- Inclusión de determinados caracteres, eliminación de caracteres no deseados (ceros no significativos, etc.), sustitución de caracteres.

#### V-3.8.2.1. Cláusula *BLANK* o "espacios"

**FORMATO**

[ **BLANK** WHEN **ZERO** ]

**OBJETO**

Transforma los caracteres CERO de la zona emisora en blancos en la zona receptora cuando todos los caracteres de la zona emisora son ceros.

**OPCIONES**

No tiene.

**PARTICULARIDADES DE SU USO**

Solamente puede aplicarse a elementos numéricos.

**EJEMPLO**

En la inscripción

02 TOTAL SIZE 4 NUMERIC BLANK WHEN ZERO

el valor impreso del campo TOTAL será espacios si el contenido de dicho campo tiene valor cero.

#### V-3.8.2.2. Cláusula *JUSTIFIED* o de alineación

**FORMATO**

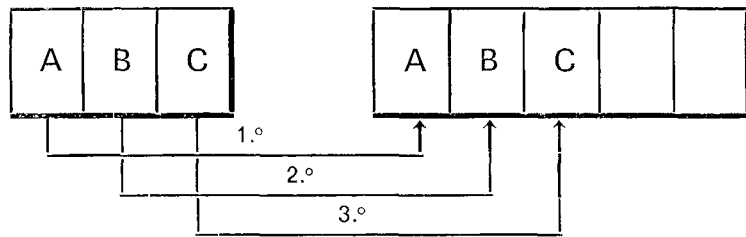
[ **JUSTIFIED** {opción} ]

**OBJETO**

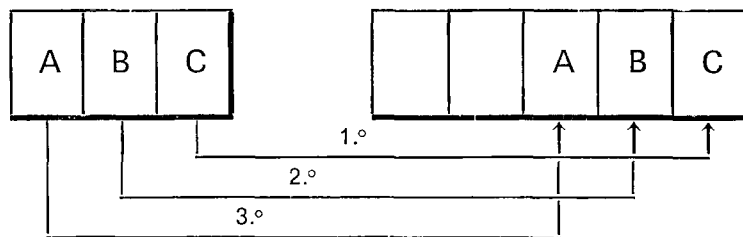
Esta cláusula se utiliza para variar algunas de las reglas de actuación de la transferencia de datos de la zona emisora a la zona receptora señaladas en el epígrafe V-3.8.2.



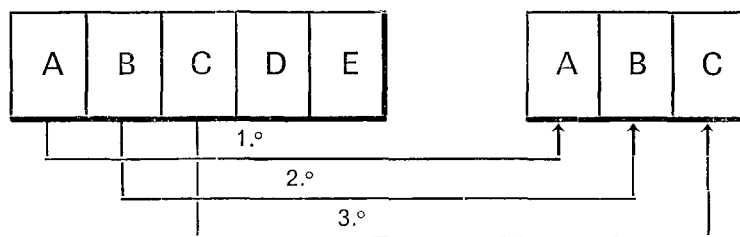
En esta transferencia, los caracteres no numéricos pasan de una zona a otra de izquierda a derecha del elemento.



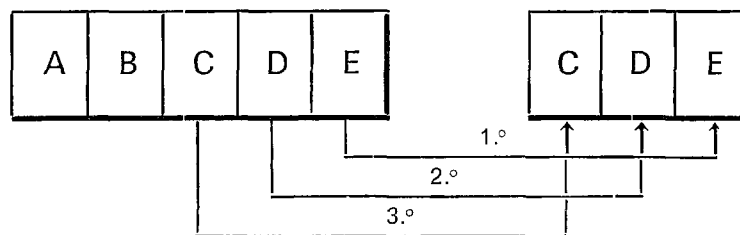
Utilizándose la cláusula JUSTIFIED, con la opción RIGHT, la transferencia tiene lugar de derecha a izquierda.



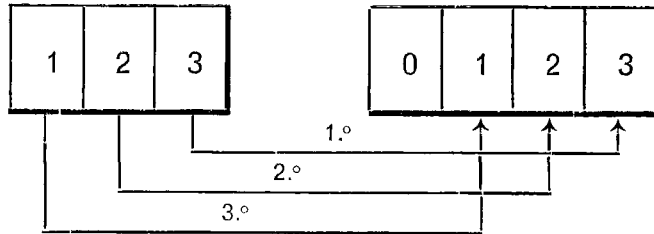
En el caso de existir truncamiento, debido a que la zona receptora sea de menor tamaño que la emisora, sin la cláusula JUSTIFIED RIGHT el truncamiento tiene lugar en los últimos caracteres, considerados de izquierda a derecha del campo.



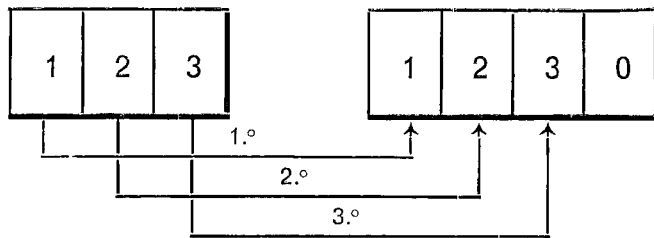
Con la cláusula JUSTIFIED RIGHT el truncamiento tiene lugar en sentido contrario.



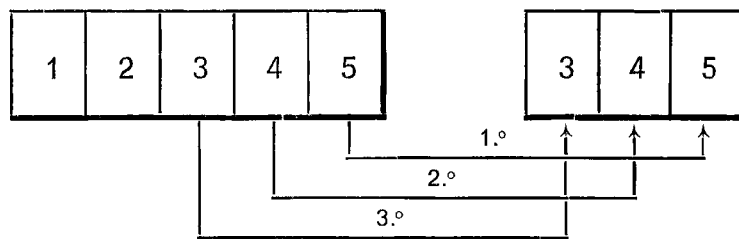
Cuando la transferencia se realiza para caracteres numéricos, éstos pasan de una zona a otra, de derecha a izquierda, del elemento (con relleno automático de ceros si la zona receptora es de mayor longitud que la emisora).



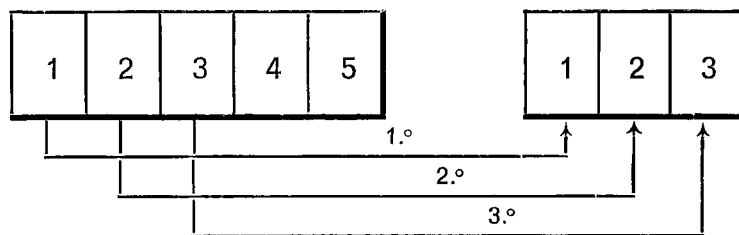
Utilizándose la cláusula **JUSTIFIED**, con la opción **LEFT**, la transferencia tiene lugar de izquierda a derecha.



En el caso de existir truncamiento, debido a que la zona receptora sea de menor tamaño que la emisora, sin la de cláusula **JUSTIFIED LEFT**, el truncamiento tiene lugar en los últimos caracteres, considerados de derecha a izquierda del campo.



Con la cláusula **JUSTIFIED LEFT**, el truncamiento tiene lugar en sentido contrario.



**OPCIONES**

**PARTICULARIDADES DE SU USO**

**EJEMPLOS**

LEFT y RIGHT, ya comentadas.

La opción LEFT tiene aplicación en campos definidos con CLASS numérica y la opción RIGHT con CLASS ALPHABETIC o ALPHA-NUMERIC y nunca en sentido contrario.

Descripción de la zona emisora:

02 E PICTURE X(5)

Descripciones de las zonas receptoras:

02 R11 PICTURE X(3) JUSTIFIED RIGHT

02 R12 PICTURE X(3)

02 R21 PICTURE X(6) JUSTIFIED LEFT

02 R22 PICTURE X(6)

02 R31 PICTURE X(5) JUSTIFIED RIGHT

02 R32 PICTURE X(5)

Si suponemos que el contenido de la zona emisora es 3A5BC y se envía la información de la zona emisora a las seis zonas receptoras indicadas, la impresión de dichas zonas daría:

R11	5	B	C			
R12	3	A	5			
R21	3	A	5	B	C	
R22	3	A	5	B	C	
R31	3	A	5	B	C	) Cuando la zona emisora y la zona receptora son de igual tamaño, la cláusula JUSTIFIED no produce ningún efecto.
R32	3	A	5	B	C	

No produce ningún efecto (es equivalente a no haber utilizado cláusula JUSTIFIED)

V-3.8.2.3. Cláusula ZERO o de supresión de ceros no significativos

**FORMATO**

**ZERO SUPRESS [ LEAVING número-entero PLACES ]**

**OBJETO**

Suprime, en la transferencia de la zona emisora a receptora, todos los ceros no significativos de un campo numérico.

**OPCIONES**

La única opción es la indicada en el formato. Evita la eliminación de todos los ceros no significativos, sino únicamente los que se deseen, ya que se respetan en la transferencia los ceros que quedan dentro de la zona indicada por número-entero; el valor indicado en

número-entero establece el número de dígitos, contados de derecha a izquierda, a partir de la posición del punto decimal a los que no les afecta la cláusula.

**PARTICULARIDADES DE SU USO**

Esta cláusula solamente puede utilizarse con campos numéricos con contenido en caracteres, es decir, con CLASS NUMERIC. No existe en todos los compiladores.

**EJEMPLOS**

Descripción de la zona emisora:

02 A SIZE 5 NUMERIC

Descripción de las zonas receptoras:

02 B SIZE 5 ZERO SUPRESS

02 C SIZE 5 ZERO SUPRESS LEAVING 3 PLACES

Suponiendo que el contenido de la zona emisora sea 00027, el contenido de la zona receptora después de la transferencia será:

B					2	7
---	--	--	--	--	---	---

C				0	2	7
---	--	--	--	---	---	---

V-3.8.2.4. Cláusula CHECK o de protección de cheques

**FORMATO**

**CHECK PROTECT [ LEAVING número-entero PLACES ]**

**OBJETO**

Sustituye por asteriscos todos los ceros no significativos de un campo numérico, en la transferencia de la zona emisora a la repetidora.

**OPCIONES**

Tiene la misma opción que la cláusula ZERO con las mismas particularidades, referidas a la sustitución de parte de los ceros por asteriscos.

**PARTICULARIDADES DE SU USO**

Esta cláusula solamente puede utilizarse en campos numéricos con contenido en caracteres. Utilizada en unión de la cláusula BLANK, y cuando ésta tiene efecto, puede dar lugar a sustituir todos los caracteres por asteriscos o todos los caracteres por espacios (5).

**EJEMPLOS**

Descripción de la zona emisora:

02 A SIZE 6 NUMERIC

Descripción de zonas receptoras:

02 B SIZE 6 CHECK PROTECT

02 C SIZE 6 CHECK PROTECT LEAVING 3 PLACES

Suponiendo que el contenido de la zona emisora sea 00027, el contenido de las zonas receptoras, después de la transferencia, será:

B	*	*	*	*	2	7
---	---	---	---	---	---	---

C	*	*	*	0	2	7
---	---	---	---	---	---	---

(5) Depende de los compiladores utilizados.

V-3.8.2.5. Cláusula *FLOAT* o de moneda

**FORMATO**

[ **FLOAT** DOLLAR SIGN [ **LEAVING** número-entero PLACES ] ]

**OBJETO**

Esta cláusula permite reemplazar, en la transferencia de la zona emisora a la receptora, los ceros no significativos por espacios, salvo el cero más extremo derecho, el cual se reemplaza por el signo \$.

**OPCIONES**

Tiene las mismas opciones que las ya estudiadas en las cláusulas ZERO y CHECK.

**PARTICULARIDADES DE SU USO**

Solamente puede utilizarse en campos numéricos con contenido en caracteres. Es necesario prever un tamaño mayor en la zona receptora para poder albergar el signo \$, en los casos que sea necesario. Cuando se utilice la opción CURRENCY SIGN del párrafo SPECIAL-NAMES, se sustituye el carácter \$ por el carácter elegido en la opción.

**EJEMPLOS**

Descripción de la zona emisora:

02 A SIZE 5 NUMERIC

Descripción de las zonas receptoras:

02 B SIZE 5 FLOAT DOLLAR SIGN

02 C SIZE 5 FLOAT DOLLAR SIGN LEAVING 3 PLACES

Suponiendo que el contenido de la zona emisora sea 00027, el contenido de las zonas receptoras, después de la transferencia, será:

B			\$	2	7
---	--	--	----	---	---

C		\$	Ø	2	7
---	--	----	---	---	---

V-3.8.2.6. Cláusula *PICTURE* o modelo

**FORMATO**

[ **PICTURE** IS {opción = serie caracteres} ]

**OBJETO**

Además de las funciones que desarrolla esta cláusula con las opciones estudiadas en la descripción de registros lógicos de ficheros de entrada, existen opciones especiales a efectos de edición de resultados. Esta cláusula reúne las características estudiadas en las cláusulas BLANK, ZERO, CHECK y FLOAT e incluye una nueva serie de funciones que dependen de la serie de caracteres elegidos.

**OPCIONES**

Existen dos tipos de opciones básicas:

- Caracteres de inserción.
- Caracteres de sustitución.

**CARACTERES DE INSERCION**

**B** **∅** que intervienen en la cláusula con los caracteres X y A de la cláusula PICTURE general.

**\$** **+** **—** **,** **B** **∅** **CR** **DB** que intervienen en las cláusula con los caracteres 9VP de la cláusula PICTURE general.

CARACTERES DE SUSTITUCION			
Cadena de Z,	ZZZZ	.....	o Z flotante
Cadena de *,	****	.....	o * flotante
Cadena de \$,	\$\$\$\$	.....	o \$ flotante
Cadena de +,	++++	.....	o + flotante
Cadena de —,	-----	.....	o — flotante

CARACTER	FUNCION	EJEMPLOS		
		ZONA EMISORA	PICTURE	ZONA RECEPTORA
<b>B</b>	Cada carácter B permite insertar un espacio en la zona receptora.	3   A   B   C	XXBXX	3   A     B   C
<b>∅</b>	Cada carácter ∅ permite insertar un carácter ∅ en la zona receptora.	3   A   B   C	XX0XX	3   A   ∅   B   C
<b>\$</b>	Un carácter \$ colocado en primer lugar, a la izquierda, de la cláusula PICTURE, permite insertar el signo \$.	3   5   6   7	\$9999	\$   3   5   6   7
<b>+</b>	Este carácter, colocado en primer o último lugar de una cláusula PICTURE, permite insertar un signo + o — en el mismo lugar que ocupa el carácter. Cuando el campo a editar sea positivo, nulo o sin signo, se inserta el carácter +; si el campo a editar es negativo, se inserta el carácter —.	3   5   6   7	+9999	+   3   5   6   7
		3   5   6   7	+9999	—   3   5   6   7
		3   5   6   7	+9999	+   3   5   6   7
		3   5   6   7	9999+	3   5   6   7   +
		3   5   6   7	9999+	3   5   6   7   —
		3   5   6   7	9999+	3   5   6   7   +
		3   5   6   7	9999+	3   5   6   7   +

CARACTER	FUNCION	EJEMPLOS																																																																																																																																																																																																																																																																									
		ZONA EMISORA	PICTURE	ZONA RECEPTORA																																																																																																																																																																																																																																																																							
-	Este carácter, colocado en primer o último lugar de una cláusula PICTURE, permite insertar un signo — o un espacio en el mismo lugar que ocupa el carácter. Cuando el campo a editar sea positivo, nulo o sin signo, se inserta un espacio. Si el campo a editar es negativo, se inserta el carácter Ø.	<table border="1"> <tr><td></td><td></td><td></td><td style="text-align: center;">+</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td></td><td></td><td style="text-align: center;">-</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td></td><td></td><td style="text-align: center;">+</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td></td><td></td><td style="text-align: center;">-</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td></td><td></td><td style="text-align: center;">+</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td></td><td></td><td style="text-align: center;">-</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td></tr> </table>				+	3	5	6	7				-	3	5	6	7				+	3	5	6	7				-	3	5	6	7				+	3	5	6	7				-	3	5	6	7	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																																																																																																								
			+																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
			-																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
			+																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
			-																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
			+																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
			-																																																																																																																																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
.	Este carácter, que puede estar colocado en cualquier lugar de la cláusula PICTURE, permite insertar un carácter . en el lugar en que se encuentra el carácter V de la cláusula PICTURE de la zona emisora o en el lugar ocupado por el carácter a insertar cuando no existe el carácter V.	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>									3	5	6	7																																																																																																					<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
,	Este carácter, que puede estar colocado en cualquier lugar de la cláusula PICTURE, permite insertar un carácter en el mismo lugar en que se encuentre en la cláusula.	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td>5</td><td>6</td><td>7</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>									3	5	6	7																																																																																																					<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																																								
3	5	6	7																																																																																																																																																																																																																																																																								
<b>CR</b> <small>CRédito signo bancario</small>	Estos dos caracteres, que solamente pueden figurar en el extremo derecho de la cláusula PICTURE, permiten insertar los mismos caracteres en la zona receptora cuando el contenido de la zona emisora sea negativo. Si el contenido es positivo, nulo o sin signo, se insertan dos espacios.	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																									<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																																
<b>DB</b> <small>DéBito signo bancario</small>	Estos dos caracteres, que solamente pueden figurar en el extremo derecho de la cláusula PICTURE, permiten insertar los mismos caracteres en la zona receptora cuando el contenido de la zona emisora sea negativo. Si el contenido es positivo, nulo, o sin signo, se insertan dos espacios.	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																																																																								

CARACTER	FUNCION	EJEMPLOS		
		ZONA EMISORA	PICTURE	ZONA RECEPTORA
<div style="border: 1px solid black; padding: 5px; width: fit-content;">ZZZZ...</div> <p>Z flotante</p>	Una serie de caracteres Z en cadena, consecutivos o separados por los códigos de inserción ., B y 0, permiten sustituir los ceros no significativos, colocados en el mismo lugar que en la cláusula de la zona emisora, por espacios en la zona receptora.	0   0   0   0   3   5 0   0   0   0   3   5 0   0   0   9   3   5	ZZZZ99 ZZZ99 ZZZ.999	3   5       0   3   5       .   5   3   5
<div style="border: 1px solid black; padding: 5px; width: fit-content;">***...</div> <p>* flotante</p>	Esta cadena de caracteres tiene las mismas reglas de actuación que la serie de caracteres ZZZ... con la particularidad de sustituir los ceros no significativos por *.	0   0   0   3   5 0   0   0   3   5 0   0   9   3   5 ↑ punto decimal supuesto	***99 **999 **.*9	*   *   *   3   5 *   *   0   3   5 *   *   .   9   3   5
<div style="border: 1px solid black; padding: 5px; width: fit-content;">\$\$\$...</div> <p>\$ flotante</p>	Esta serie de caracteres permite sustituir los ceros no significativos de la zona emisora que ocupen el mismo lugar que los caracteres \$ de la cláusula, por espacios en la zona receptora, excepto el ocupado por el último cero no significativo, contando de izquierda a derecha de la zona, que es sustituido por un signo \$.	0   0   0   2   7 0   0   0   2   7 0   0   0   1   2 ↑ punto decimal supuesto	\$\$\$99 \$\$999 \$\$9.9CR	\$   2   7     \$   0   2   7   \$   1   .   2   C   R
<div style="border: 1px solid black; padding: 5px; width: fit-content;">+++...</div> <p>+ flotante</p>	Esta serie de caracteres permite sustituir los ceros no significativos de la zona emisora que ocupen el mismo lugar que los caracteres + de la cláusula, por espacios en la zona receptora, excepto el ocupado por el último cero no significativo, contando de izquierda a derecha de la zona, el cual es sustituido por un signo + si el valor de la zona emisora es positivo, nulo o sin signo y por un signo - si el valor es negativo.	0   0   0   3   5   6 0   0   0   3   5   6 0   0   0   3   5   6 0   0   0   3   5   6	+++99 +++99 +++99 +++99	-   3   5   6 +   3   5   6 +   3   5   6 +   3   5   6
<div style="border: 1px solid black; padding: 5px; width: fit-content;">---...</div> <p>-- flotante</p>	Tiene el mismo efecto que el + flotante, salvo que se edita un espacio, en lugar del signo + cuando el valor de la zona emisora es positivo, nulo o sin signo.	0   0   0   3   5   6 0   0   0   3   5   6 0   0   0   3   5   6	---99 ---99 ---99	-   3   5   6       3   5   6       3   5   6



**PARTICULARIDADES  
DE SU USO**

En la escritura de una cláusula PICTURE de edición se deben tener en cuenta las siguientes particularidades:

- a) Solamente se puede utilizar un carácter de sustitución a la vez, es decir, uno entre los siguientes: Z \$ * + -.
- b) No se puede utilizar al mismo tiempo el carácter de sustitución + flotante o - flotante y los caracteres de inserción + • -.
- c) El signo + como sustitución o inserción no puede utilizarse conjuntamente con el signo - en la misma cláusula.
- d) El carácter • no puede escribirse nunca como carácter extremo derecho.
- e) No puede aparecer a la izquierda de un carácter de sustitución ningún carácter 9.
- f) Los caracteres V CR y DB no pueden aparecer más que una sola vez en el contenido de una cláusula.
- g) Para que un carácter + o - sea considerado como carácter de sustitución, debe formarse como mínimo una cadena de dos de ellos, consecutivos o separados por caracteres de inserción.
- h) Un carácter \$ y un carácter + o - no pueden encontrarse en la misma hilera, solamente se permite en el caso de que el carácter + o el - se encuentren en la posición extrema derecha.

**V-3.9. UTILIZACION DE LA PALABRA RESERVADA FILLER EN LAS INSCRIPCIONES DE DESCRIPCION DE REGISTROS**

Las inscripciones de descripción de registro definen las características de los campos del mismo. En un problema determinado puede ocurrir que no se tenga necesidad de utilizar todos los campos que forman el registro del fichero; sin embargo, el contenido total del registro pasará del soporte externo a la zona de entrada de la memoria central ocupando un espacio igual a la longitud total del registro; no se pueden eliminar en la transferencia de información del soporte a la memoria central, aquellos campos que no intervengan posteriormente en el proceso.

En estas circunstancias parece inútil dar nombre de dato a los campos que no van a ser utilizados. El COBOL prevé esta circunstancia y utiliza la palabra reservada FILLER en las inscripciones de aquellos campos de los que no se desea efectuar una descripción específica.

Un fichero de clientes contiene registros con el siguiente formato:

**EJEMPLO**

TIPO	NOMBRE	NUM. CLIENTE	SALDO

Cualquier programa que utilice este registro para obtener un listado de direcciones de clientes no necesitará, como es lógico, el campo SALDO. La descripción del registro en estas circunstancias sería:

02 TIPO PICTURE 9  
02 NOMBRE PICTURES A(20)  
02 NUMERO-CLIENTE PICTURE 9(5)  
02 FILLER PICTURE X(6)

00127	002010	DATA DIVISION.	
00128	002020	FILE SECTION.	
00129			
00130			
00131	002040	FD FICHAS	
00132	002050	RECORDING F	
00133	002070	RECORD CONTAINS 80 CHARACTERS	
00134	002080	LABEL RECORD OMITTED	
00135	002090	DATA RECORD FICHA-E.	
00136	002100	01 FICHA PIC X(80).	
00137			
00138			
00139	003010	FD DISCO	
00140	003020	RECORDING MODE IS F	
00141	003040	RECORD CONTAINS 11 CHARACTERS	
00142	003050	LABEL RECORD STANDARD	
00143	003060	DATA RECORD DIS.	
00144	003070	01 DIS.	
00145	003075	02 FILLER PIC X.	
00146	003080	02 CLAVE-DISCO PIC X(4).	
00147	003090	02 COEFICIENTE-D PIC S9(3)V999	COMP-3.
00148	003100	02 CODIGO PIC X.	
00149	003110	02 TIPO-DISCO PIC X.	
00150	003120	88 SOLTEROS VALUE 1.	
00151	003130	88 CASADOS VALUE 2.	
00152	003140	88 VIUDOS VALUE 3.	
00153			
00154			
00155			
00156	004010	FD IMPRESORA	
00157	004020	RECORDING F	
00158	004030	RECORD 133	
00159	004040	LABEL RECORD OMITTED	
00160	004050	DATA RECORD IMPRE.	
00161	004070	01 IMPRE PIC X(133) SYNC.	
00162			
00163			
00164	005010	FD CINTA	
00165	005020	RECORDING F	
00166	005030	RECORD 12	
00167	005040	BLOCK 30 RECORDS	
00168	005050	LABEL RECORD STANDARD	
00169	005060	DATA RECORD CIN.	
00170	005070	01 CIN PIC X(12).	
00171			
00172			
00173			
00174			
00175			
00176			
00177			
00178			
00179			
00180			
00181			
00182			
00183			
00184			
00185			
00186			
00187			
00188			
00189			

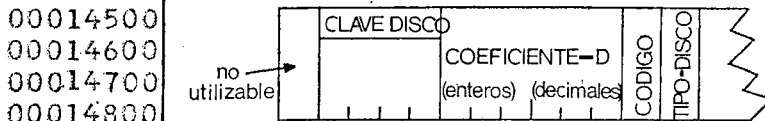
00012700 | Es necesario definir en la sección de ficheros cuatro niveles FD, uno  
00012800 | para cada fichero, tal como figuran en las líneas de referencia 002040,  
003010, 004010 y 005010.

00012900 |  
00013000 | El fichero FICHAS tiene las siguientes características:  
00013100 | — Registros de longitud fija.  
00013200 | — Longitud de registro de 80 caracteres.  
00013400 | — Bloques de un registro (al estar omitida la cláusula BLOCK CON-  
00013500 | TAINS).  
00013600 | — No contiene etiquetas.

00013700 | Se atribuye el nombre FICHA-E como nombre de registro.  
00013800 | No se hace la descripción de los campos en que está desglosado el  
00013900 | registro. Esta descripción se realiza en la WORKING-STORAGE SECTION.

00014000 | El fichero DISCO tiene las siguientes características:  
00014100 | — Registro de longitud fija.  
00014200 | — Longitud de registro de 11 caracteres.  
00014300 | — Bloques de un registro.  
00014300 | — Etiquetas tipo estándar.

00014400 | Se atribuye el nombre DIS como nombre de registro cuyo formato es:



00014900 | La primera posición del registro no es utilizada en el programa.  
00015000 | El campo COEFICIENTE-D se almacena en formato empaquetado (dos  
00015100 | dígitos por BYTE).  
00015200 | Al campo TIPO-DISCO, con tres valores posibles, se le asignan los  
00015300 | nombres de condición: SOLTEROS para el valor 1, CASADOS para el  
00015400 | valor 2, VIUDOS para el valor 3.

00015500 |  
00015600 |

00015700 | El fichero IMPRESORA tiene las siguientes características:  
00015800 | — Registros de longitud fija.  
00015900 | — Longitud de registro de 133 caracteres (el primer carácter como  
00016000 | carácter de control).  
00016100 | — Sin etiquetas.

00016200 | Se atribuye el nombre IMPRE como nombre de registro y cuya definición  
00016300 | de campos se realiza en la WORKING-STORAGE SECTION.

00016400 |

00016500 | El fichero CINTA tiene las siguientes características:

00016600 |  
00016700 | — Registros de longitud fija.  
00016800 | — Longitud de registro de 12 caracteres.  
00016900 | — Factor de blocaje = 30 (30 registros lógicos en un registro físico).

00017000 | Se asigna el nombre CIN a los registros del fichero. No tiene desglose  
00017100 | de campos elementales y, por consiguiente, el propio registro es el  
00017200 | único campo del registro.

00017300 |  
00017400 |  
00017500 |  
00017600 |  
00017700 |  
00017800 |  
00017900 |  
00018000 |  
00018100 |  
00018200 |  
00018300 |  
00018400 |  
00018500 |  
00018600 |  
00018700 |  
00018800 |  
00018810 |  
00018820 |

V - 4

SECCION DE TRABAJO  
O MANIOBRA



#### V-4. SECCION DE TRABAJO O MANIOBRA

##### V-4.1. GENERALIDADES

Durante la ejecución del programa los resultados intermedios y otras informaciones necesitan estar almacenados antes de ser procesados o transferidos fuera de la memoria. Las áreas de almacenamiento, también denominadas *áreas de maniobra*, mantienen, pues, temporal o totalmente, datos iniciales y resultados que necesitan ser tratados en el desarrollo del proceso.

Las inscripciones que normalmente se mantienen en este área de maniobra son tablas de correspondencia, índices, contadores, etc.

##### V-4.2. INSCRIPCIONES EN LA SECCION DE TRABAJO

Las distintas cláusulas que se pueden utilizar en las inscripciones de las zonas de almacenamiento se han estudiado en los epígrafes anteriores V-3, y todas son de aplicación en esta sección.

Sin embargo existen dos cláusulas prácticamente específicas de la WORKING-STORAGE SECTION:

- Cláusula SYNCHRONIZED.
- Cláusula VALUE.

La cláusula VALUE, estudiada como aplicación a la descripción de nombre de condición y la cláusula OCCURS, adquiere en esta sección relevancia particular para describir tablas de correspondencia.

También la cláusula REDEFINES tiene en esta sección una utilización particular diferente a la función realizada como cláusula de descripción de registros en la FILE SECTION.

##### V-4.2.1. Cláusula SYNCHRONIZED o de sincronización

[ **SYNCHRONIZED** {opción} ]

#### OBJETO

Esta cláusula se emplea fundamentalmente en ordenadores que utilizan la noción de "palabra de máquina de longitud fija" como unidad de almacenamiento en lugar de octetos. Consecuentemente, debe ocurrir con esta forma de almacenamiento, que determinados campos serán

de longitud menor que los espacios de memoria asignados para almacenarlos. Esto indica que pueden existir "trozos" de palabras no ocupadas que se rellenan con caracteres no significativos, tales como ceros o blancos, según el sistema de que se trate. Para cada constructor una palabra máquina contiene un número fijo de caracteres. Si el carácter es directamente direccionable, cada carácter tiene su propia dirección. Si el ordenador tiene un direccionamiento por palabra, un carácter en una palabra de memoria, queda localizado por una dirección de palabra y un número de esta palabra.

Cuando se asocia a la cláusula SYNCHRONIZED, el campo elemental se coloca al principio o al fin de una nueva palabra máquina. Se encuentra así solo en una palabra máquina.

### OPCIONES

Las dos opciones posibles son LEFT y RIGHT.

Si se elige la opción LEFT, el primer carácter del elemento asociado ocupa la posición más a la izquierda de la siguiente palabra máquina.

Se comprende fácilmente que la parte derecha de la palabra precedente queda desocupada. En general, si el campo no ocupa un número entero de palabras máquina, la parte derecha de la última palabra máquina queda desocupada.

Si se elige la opción RIGHT, el último carácter del campo asociado ocupa la parte extrema derecha de una nueva palabra máquina. Contrariamente a la opción LEFT, si el campo no ocupa un número entero de palabras máquina, la parte siguiente de la primera palabra máquina ocupada por este campo queda desocupada.

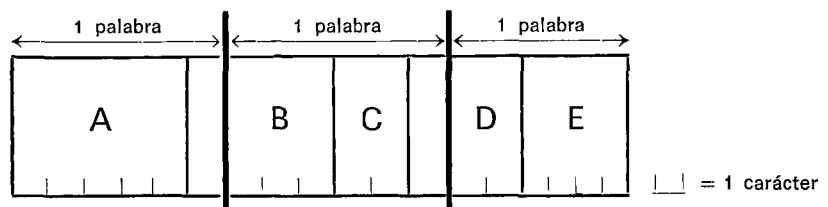
### EJEMPLOS

Supongamos un ordenador cuyas palabras máquina contengan seis caracteres.

Un registro tiene la siguiente descripción:

- Ø1 R1.
- Ø2 A PICTURE X(5).
- Ø2 B PICTURE X(4).
- Ø2 C PICTURE XX.
- Ø2 D PICTURE X(3).
- Ø2 E PICTURE X(4).

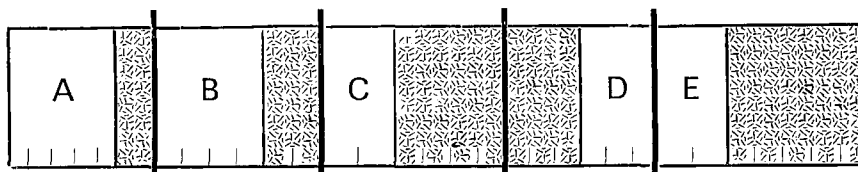
La longitud total del registro R1 será de 18 caracteres, es decir, tres palabras máquina.





Si efectuamos la descripción de la siguiente forma:

- Ø1 R1.
- Ø2 A PICTURE X(5).
- Ø2 B PICTURE X(4) SYNCHRONIZED LEFT.
- Ø2 C PICTURE XX.
- Ø2 D PICTURE X(3) SYNCHRONIZED RIGHT.
- Ø2 E PICTURE X(4).



Aunque como se ve, la longitud total es ahora de cinco palabras máquina, el acceso a los elementos sincronizados se realiza de esta forma mucho más rápidamente.

#### V-4.2.2. Cláusula VALUE o de valor inicial

**[ VALUE[S] IS literal ]**

##### OBJETO

La función específica de este elemento en la WORKING-STORAGE SECTION (ya se estudió su función en la FILE SECTION), es la de indicar el contenido inicial del campo elemental al que hace referencia la cláusula antes de comenzarse a ejecutar el programa.

##### EJEMPLO

Si en un programa se tuviese que realizar el cálculo de la longitud de una circunferencia, se podría utilizar esta cláusula en la WORKING-SECTION creando un campo independiente a nivel 77:

```
77 PI PICTURE 9V9999 VALUE IS 3.1416.
```

##### PARTICULARIDADES DE SU USO

Se puede utilizar a cualquier nivel, ya sea de grupo elemental o especial.

Es necesario que el tipo de literal esté en concordancia con la clase del campo, al cual está asociado.

Se pueden utilizar las constantes figurativas en el lugar de los literales.

#### V-4.2.3. Cláusula OCCURS

**[ OCCURS número-entero TIMES [ INDEXED BY nombre-dato ] ]**

##### OBJETO

Ya se ha visto en la FILE SECTION que la cláusula OCCURS se utiliza para indicar la repetición de campos que tengan idénticos formatos. El uso específico de esta cláusula en la WORKING-STORAGE SECTION, estriba en la definición de tablas.

**TABLAS:** Una tabla es una serie de campos almacenados en la memoria central dispuestos en orden lógico significativo y consecutivo.

La estructura de una tabla se especifica por medio de la cláusula OCCURS.

La cláusula OCCURS especifica el número de veces que se repite un campo de datos. Un campo descrito por la cláusula OCCURS se llama elemento de la tabla. El nombre y la descripción de elemento de la tabla se aplica a cada repetición o aparición del campo de datos.

El COBOL permite tablas de tres dimensiones. Cada dimensión se debe definir por una cláusula OCCURS, con nivel jerárquico diferente. Para definir una tabla de una dimensión hay que escribir una cláusula OCCURS como parte de la descripción de datos del campo repetido.

La definición de una tabla de una dimensión dentro de cada aparición de un elemento de la tabla da como resultado una tabla de dos dimensiones. Esto se efectúa escribiendo una segunda cláusula OCCURS como parte de la descripción de un campo de datos subordinado (es decir, que tenga mayor número de nivel) a otro campo que tenga escrita, a su vez, una cláusula OCCURS.

Para comprender mejor todo lo anteriormente expuesto, veamos un ejemplo:

Supongamos una TABLA DE POBLACION que contenga los datos de población de las provincias españolas.

PROVINCIA	HABITANTES
ALAVA	204.323
ALBACETE	355.026
ALICANTE	920.105
ALMERIA	375.004
⋮	⋮
ZARAGOZA	760.186

Esta tabla podría almacenarse en la memoria efectuando la inscripción de 50 campos independientes, con nivel 77:

```

77 ALAVA PICTURE 9(6)
77 ALBACETE PICTURE 9(6)
    ⋮
77 ZARAGOZA PICTURE 9(6)

```

Sin embargo, resulta más breve considerar la lista de provincias como un conjunto organizado en tabla, pudiéndose almacenar en memoria de la siguiente forma:

- Ø1 TABLA-POBLACION (Como conjunto)
- Ø2 PROVINCIA OCCURS 5Ø TIMES PICTURE 9(7)

(Se debe indicar PICTURE 9(7), ya que el contenido de alguna provincia puede ser 7 a/'s, como Madrid, Barcelona, etc.)

Obsérvese que la cláusula OCCURS no puede ser usada a nivel 01, ya que siempre estará subordinada al conjunto. La tabla indicada en la WORKING-STORAGE SECTION define 50 posiciones de siete caracteres.

Se reservan, pues, 350 posiciones de la memoria central.

La tabla anterior es una muestra clara de una tabla de una dimensión.

Supongamos ahora que la tabla de población contiene los habitantes de las provincias españolas en los tres últimos años. Se representaría gráficamente según muestra la figura.

PROVINCIA	HABITANTES		
	1971	1972	1973
ALAVA	204.323	212.650	228.475
ALBACETE	355.026	370.026	396.001
ALICANTE	920.105	938.205	942.306
⋮	⋮	⋮	⋮
ZARAGOZA	760.186	782.189	793.225

Sin la cláusula OCCURS, la reserva, en memoria central, para almacenar esta tabla debería hacerse:

Recorriendo la tabla de izquierda a derecha y de arriba hacia abajo.	}	77 ALAVA-71 PICTURE 9(6)
		77 ALAVA-72 PICTURE 9(6)
		77 ALAVA-73 PICTURE 9(6)
		77 ALBACETE-71 PICTURE 9(6)
		⋮
		77 ZARAGOZA PICTURE 9(6)
Recorriendo la tabla de arriba hacia abajo y de izquierda a derecha.	}	77 ALAVA-71 PICTURE 9(6)
		77 ALBACETE-71 PICTURE 9(6)
		⋮
		77 ZARAGOZA-71 PICTURE 9(6)
		77 ALAVA-72 PICTURE 9(6)
		77 ZARAGOZA-73 PICTURE 9(6)

Utilizando en cambio la cláusula OCCURS, la reserva de espacio en memoria se haría de la siguiente forma:

- Ø1 TABLA-POBLACION.
- Ø2 PROVINCIA OCCURS 5Ø TIMES.
- Ø3 ELEMENTO OCCURS 3 TIMES PICTURE 9(7).

o también:

- Ø1 TABLA-POBLACION.
- Ø2 ANNO OCCURS 3 TIMES.
- Ø3 ELEMENTO OCCURS 5Ø TIMES PICTURE 9(7).

Como puede ya deducirse, la tabla anterior es una tabla de dos dimensiones.

Por último, si deseamos tener almacenada una tabla de la población de las provincias españolas en los tres últimos años separando la población activa de la pasiva, la tabla sería:

PROVINCIA	HABITANTES					
	1971		1972		1973	
	Act.	Pas.	Act.	Pas.	Act.	Pas.
ALAVA	95.007	109.316	99.100	113.550	110.210	117.865
ALBACETE	126.198	228.828	130.205	239.821	134.220	251.781
ZARAGOZA						

La reserva sin cláusulas OCCURS se haría:

- 77 ALAVA-71-ACT PICTURE 9(6)
- 77 ALAVA-71-PAS PICTURE 9(6)
- 77 ALAVA-72 ACT PICTURE 9(6)
- 77 ALAVA-72-PAS PICTURE 9(6)
- 77 ALAVA-73-ACT PICTURE 9(6)
- 77 ALAVA-73-PAS PICTURE 9(6)
- ⋮
- 77 ZARAGOZA-73-PAS PICTURE 9(6)

Es decir, un total de 300 inscripciones.

En cambio, utilizando la cláusula OCCURS, la reserva de espacio se efectuaría:

- Ø1 TABLA-POBLACION.
- Ø2 PROVINCIA OCCURS 5Ø TIMES.
- Ø3 ANNO OCCURS 3 TIMES.
- Ø4 ELEMENTO OCCURS 2 TIMES PICTURE 9(7).

Esta última tabla es una tabla de tres dimensiones.

#### OPCIONES

La opción INDEXED BY nombre-dato, que no es utilizable en todos los compiladores, se emplea para indicar el dato que va a servir como subíndice en el manejo de la tabla. En los compiladores que no utilizan esta opción de la cláusula, los subíndices se definen como campos elementales de la WORKING-STORAGE SECTION.

#### V-4.2.4. Cláusula REDEFINES o de redefinición

[ **REDEFINES** nombre-dato ]

#### OBJETO

Todo lo expuesto acerca de esta cláusula en la descripción de los registros lógicos de los ficheros de entrada permanece válido para su uso en la WORKING-STORAGE SECTION; pero además, en esta sección, al estar permitido la utilización de esta cláusula a nivel 01, se puede efectuar un ahorro, a veces considerable, de memoria, ya que la misma zona de trabajo puede utilizarse con fines diferentes a lo largo del desarrollo del programa.

Esta cláusula, en unión de las cláusulas OCCURS y VALUE, permite almacenar en memoria el contenido de los valores de una tabla.

En primer lugar se reserva la zona de trabajo que va a albergar la tabla, el contenido numérico total de la misma, utilizándose una o varias cláusulas VALUE, normalmente a gusto o conveniencia del programador.

Después se efectúa la estructuración de la zona de memoria ya definida, volviendo a hacer referencia a ella, es decir, redefiniéndola con la cláusula REDEFINES. Lo que podríamos llamar reticulación o rayado de la tabla, se realiza con la cláusula OCCURS.

#### EJEMPLO

Utilizando la misma tabla, de una dimensión, empleada en el epígrafe anterior, la codificación total en la WORKING-STORAGE SECTION, se realizará de la siguiente forma:

- Ø1 TABLA-POBLACION.
- Ø2 FILLER PICTURE X(21) VALUE '020432303550260920105'.
- Ø2 FILLER PICTURE X(21) VALUE '037500402037980687599'.
- Ø2 FILLER PICTURE X(21) VALUE '038756503227580343456'.
- ⋮
- Ø1 ESTRUCTURA-TABLA REDEFINES TABLA-POBLACION.
- Ø2 ELEMENTO PICTURE 9(7) OCCURS 5Ø TIMES.

Con el primer grupo de inscripciones se reservan 350 posiciones consecutivas de memoria que se inicializan con el valor:

020432303550260920105037500402037910687599 ....., hasta 350 cifras.

Con el segundo grupo de inscripciones se indica que todo el conjunto anteriormente definido está compuesto de 50 elementos de siete posiciones cada uno.

El acceso individual a cada una de las posiciones se realiza utilizando el nombre de índice, en este caso ELEMENTO acompañado del subíndice correspondiente. Por ejemplo, para acceder a la población de Alicante se indicará el nombre ELEMENTO (3) o ELEMENTO (nombre-dato), siendo nombre-dato el índice que, en este caso deberá tener planeado el valor 3.

Obsérvese que la reserva de memoria se puede realizar de cualquier otra forma hasta completar las 350. Por comodidad el programador suele realizarlo en trozos múltiples de los elementos de la tabla. En este caso podría haberse hecho de 50 en 50 o de 70 en 70, etc., incluso, en trozos de distinta longitud.

También debe destacarse el empleo de la palabra reservada FILLER, ya que no tiene ningún objeto el dar nombre a cada uno de los trozos, sino al conjunto total que es TABLA-POBLACION. Cada uno de sus elementos se conocerá con el nombre genérico de ELEMENTO acompañado de su correspondiente subíndice.

00190	006010	WORKING-STORAGE SECTION.			
00191	006020	77 FICHAS-LEIDA	PIC	S9(7)	VALUE ZERO
00192	006030	77 RGTOS-ESCRIT	PIC	S9(7)	VALUE ZERO
00193	006040	77 TOTAL-COE	PIC	S9(3)V999	VALUE ZERO
00194	006050	77 INFLACION	PIC	S9V9999	VALUE +1,01
00195	006060	77 INDICE-APLICA	PIC	S999V999	VALUE ZERO
00196	006070	77 X1	PIC	S99	VALUE +1
00197	006080	77 HOJA	PIC	S999	VALUE ZERO.
00198	006090	77 CLAVE-ID	PIC	X(4)	VALUE ZERO.
00199	006100	77 SW1	PIC	9	VALUE ZERO.
00200	006200	77 CONTADORLI	PIC	S99	VALUE ZERO
00201	007010	01 FILLER.			
00202	007020	02 FICHA-FECHA.			
00203	007030	03 FILLER	PIC	XX	VALUE ZERO.
00204	007040	03 MES-FECHA	PIC	99	VALUE ZERO.
00205	008010	02 WORKAFICHA.			
00206	008020	03 PARADISCO.			
00207	008030	04 CLAVEFICHA.			
00208	008032	05 1CLA	PIC	9	VALUE ZERO.
00209	008034	05 2CLA	PIC	9	VALUE ZERO.
00210	008036	05 3CLA	PIC	9	VALUE ZERO.
00211	008038	05 4CLA	PIC	9	VALUE ZERO.
00212	008040	04 COEFICI-FICHA	PIC	X(6)	VALUE ZERO.
00213	008050	04 PTSCOEFI REDEFINES COEFICI-FICHA	PIC	9(3)	
00214	008060	04 CODIGO	PIC	X	VALUE SPACE.
00215	008070	04 TIPO-FICHA	PIC	X	VALUE ZERO.
00216	008080	04 TIPO REDEFINES TIPO-FICHA	PIC	9.	
00217	008090	03 FILLER	PIC	X(68)	VALUE SPACE.
00218	009010	02 T-COEFICIENTE.			
00219	009020	03 FILLER	PIC	X(18)	VALUE '00311522013314082
00220	009040	02 FILLER REDEFINES T-COEFICIENTE.			
00221	009050	03 C-TABLA	PIC	S9V9	OCCURS 9.
00222	009080	02 TABLAMES.			
00223	009090	03 FILLER	PIC	X(24)	VALUE '01ENERO 02FE
00224	009100	03 FILLER	PIC	X(24)	VALUE '03MARZO 04AB
00225	009110	03 FILLER	PIC	X(24)	VALUE '05MAYO 06JU
00226	009120	03 FILLER	PIC	X(24)	VALUE '07JULIO 08AG
00227	009130	03 FILLER	PIC	X(24)	VALUE '09SEPTIEMBRE10OC
00228	009140	03 FILLER	PIC	X(24)	VALUE '11NOVIEMBRE 12DI
00229	009210	02 FILLER REDEFINES TABLAMES.			
00230	009220	03 CASOS OCCURS 12.			
00231	009225	04 MESTA	PIC	99.	
00232	009230	04 NOMTA	PIC	X(10).	
00233	010010	02 LINEA1.			
00234	010020	03 FILLER	PIC	X(40)	VALUE SPACE.
00235	010030	03 FILLER	PIC	X(26)	VALUE 'LISTADO DE FICHAS
00236	010035	03 FILLER	PIC	X(17)	VALUE SPACES.
00237	010050	03 MESI	PIC	X(40)	VALUE SPACE.
00238	010060	03 FILLER	PIC	X(5)	VALUE 'PAG. '.
00239	010070	03 LHOJA	PIC	----	
00240	010075	03 FILLER	PIC	X	VALUE '-'. .
00241	010080	02 LINEA2	PIC	X(133)	VALUE ALL '-'. .
00242	010110	02 LINEA3.			
00243	010120	03 FILLER	PIC	X(48)	VALUE SPACE.
00244	010130	03 FILLER	PIC	X(36)	VALUE 'CLAVE C
00245	010140-	' CODIGO TIPO'.			
00246	010170	02 LINEA4.			
00247	010180	03 FILLER	PIC	X(49)	VALUE SPACE.
00248	010190	03 CLAVEI	PIC	X(10)	VALUE ZERO.
00249	010210	03 COEFI	PIC	X(6)	VALUE SPACE.
00250	010220	03 FILLER	PIC	X(9)	VALUE SPACE.
00251	010230	03 CODIGI	PIC	X(8)	VALUE SPACE.
00252	010250.	03 TIPOI	PIC	X	VALUE SPACE.

00019000  
 COMP-3.00019100  
 COMP-3.00019200  
 COMP-3.00019300  
 36. 00019400  
 COMP-3.00019500  
 COMP. 00019600  
 00019700  
 00019800  
 00019900  
 COMP-3.00019910

Se establecen, en el área o memoria de trabajo, las siguientes áreas o campos de nivel independiente:  
 FICHAS-LEIDA: contador de fichas procesadas o número de registros tratados del fichero FICHAS.  
 RGOTOS-ESCRIT: contador de registros grabados en el fichero CINTA (número de registros correctos del fichero FICHAS).  
 TOTAL-COE: campo descrito para almacenar el resultado de un proceso aritmético con los campos del registro FICHA.  
 INFLACION: constante con nombre con valor 1,0136.  
 INDICE-APLICA: campo descrito para almacenar el resultado de un proceso aritmético.  
 X1: nombre de subíndice, utilizado para manejar las tablas de valores empleadas en el programa (ver líneas 009010 a 009050 y 009080 a 009230).  
 HOJA: campo para albergar la numeración de páginas del fichero IMPRESORA.  
 CLAVE-ID: nominal KEY del fichero DISCO; SW1, switch utilizado en el programa.  
 CONTADOR-LI: contador de líneas impresas.

00020000  
 00020100  
 00020200  
 00020300

FICHA-FECHA es el nombre de una zona de memoria reservada para almacenar la fecha del mes de tratamiento, que se introducirá a través de una ficha por la unidad de entrada lógica (lectora de fichas).

00020400  
 00020500  
 00020600  
 00020610  
 00020620  
 00020630  
 00020640

WORKAFICHA es el nombre de un área de 80 posiciones en donde se va a almacenar el contenido de cada uno de los registros definidos en la FILE SECTION para el fichero FICHAS. Como el tratamiento de los mismos se realiza en esta sección, se efectúa aquí su descripción de campos. El formato de este área es:

CLAVEFICHA				COEFICI - FICHA ó	CODIGO	TIPO-FICHA ó TIPO	ZONA NO UTILIZADA YA QUE EL REGISTRO DE ENTRADA CONTIENE SOLO 12 caracteres de los 80 disponibles
10A	20A	30A	40A	PTS COEFI			

V999.

00020700  
 00020800  
 00020900  
 00021000  
 00021100  
 00021200

A las posiciones 5 a 10 de la zona se le atribuyen dos nombres por medio de la cláusula redefines: COEFI-FICHA con tipo alfanumérico y con valor inicial cero y PTS COEFI con tipo numérico (tres enteros y tres decimales).  
 A la posición 12 de la zona se le atribuyen también dos nombres de dato: TIPO-FICHA, alfanumérica, con valor inicial cero, y TIPO, numérico sin valor inicial.  
 Las 68 últimas posiciones no son utilizadas, ya que se corresponden a las 68 últimas columnas de la ficha que no contienen información.

91.

00021300  
 00021400  
 00021500  
 00021600  
 00021800

Se almacena una tabla, en este área, de una sola entrada, con nueve elementos. Con la inscripción de la línea 009020 se reserva el área necesaria de 18 posiciones y se almacenan los valores de los nueve elementos. Con la línea 009050 se estructura el área y se da nombre genérico a los elementos que la componen (el campo X1, de nivel 77, se utiliza como nombre de índice de dicho nombre genérico).

00	31	15	22	01	33	14	08	29
----	----	----	----	----	----	----	----	----

Cada uno de los elementos tiene el nombre C-TABLA (X1).

ENERO  
 FEBRIL  
 MARZO  
 ABRIL  
 MAYO  
 JUNIO  
 JULIO  
 AGOSTO  
 SEPTIEMBRE  
 OCTUBRE  
 NOVIEMBRE

00021900  
 00022000  
 00022100  
 00022200  
 00022300  
 00022400  
 00023100  
 00023200  
 00023210  
 00023300

Se almacena esta tabla, también de una entrada, para albergar los literales de los nombres de los meses del año que se utilizarán en el programa para imprimirse sobre la impresora, en función de la fecha introducida por la unidad de entrada lógica (ver líneas 007010 a 007040).  
 Con las inscripciones de las líneas 009090 a 009140 se reservan las 144 posiciones de memoria necesarias y se almacenan los literales correspondientes.  
 Con las inscripciones de las líneas 009220 a 009230 se estructura el área y se da nombre genérico a los elementos que la componen.

01ENERO	02FEBRERO	03MARZO	04ABRIL
---------	-----------	---------	---------

ERRONEAS

00023400  
 00023500  
 00023600  
 00023610  
 00023700  
 00023800  
 00023900  
 00023910

LINEA1 es un área de 133 posiciones que almacena el contenido de una línea de cabecera del fichero IMPRESORA.  
 MES I L HOJA L  
 LISTADO DE FICHAS ERRONEAS contenido variable PAG (contenido variable)  
 El campo LHOJA se describe con PICTURE _ _ _ , ya que va a contener como máximo tres dígitos y de esta forma, en la edición, el número de la página estará siempre enmarcado entre dos guiones.

1	41	67	84	124	129	133
---	----	----	----	-----	-----	-----

DEFICIENTE

00024000  
 00024200  
 00024300  
 00024400  
 00024500

LINEA2, es un área de 133 posiciones en la que se almacenan, exclusivamente, guiones.  
 LINEA3, es un área de 84 posiciones en la que se almacena el contenido de una línea de títulos del fichero IMPRESORA (siempre de contenido Invariable).  
 CLAVE COEFICIENTE CODIGO TIPO

1	49
---	----

00024600  
 00024700  
 00024800  
 00024900  
 00025000  
 00025100  
 00025200

LINEA4, es un área de 83 posiciones, en la que se almacena el contenido de una línea de detalle (registro lógico del fichero IMPRESORA), cuyo formato es:  
 CLAVE I COEFI CODIGO TIPO I

1	50	60	66	75	83
---	----	----	----	----	----



V - 5

SECCION DE CONSTANTES



## V-5. SECCION DE CONSTANTES

### V-5.1. GENERALIDADES

El concepto de constante con nombre se estudió en el capítulo de MORFOLOGIA. Una constante con nombre es un campo al que se le ha asignado un nombre y cuyo valor no cambia durante el curso del programa. La sección de Constantes contiene las inscripciones que describen las constantes con nombre y especifican sus valores.

La sección de Constantes se organiza exactamente de la misma forma que la de Almacenamiento de Trabajo. Se comienza con el encabezamiento CONSTANT SECTION y está formada exclusivamente por inscripciones de Descripción de Registro. Análogamente a la sección de Almacenamiento de trabajo, las inscripciones son de dos tipos: independientes y de grupo.

### V-5.2. INSCRIPCIONES DE CONSTANTES INDEPENDIENTES

Las inscripciones de constantes independientes describen un solo campo o dato que no esté subdividido y que no es, en sí mismo, una subdivisión de ningún otro campo. Se le asigna siempre el número de nivel 77. Cada campo independiente ha de describirse en una inscripción de Descripción de registro separada, compuesto de las siguientes partes:

- Número de nivel 77.
- Nombre de dato.
- Cláusula CLASS o PICTURE.
- Cláusula SIZE (en el caso de no usarse la cláusula PICTURE).
- Cláusula VALUE.

Tanto la cláusula OCCURS como REDEFINES carecen de significado en una inscripción de constante independiente. Al escribir la sección de constantes, las inscripciones independientes preceden a las de grupo.

### V-5.3. INSCRIPCIONES DE CONSTANTES DE GRUPO O AGRUPADAS

Las inscripciones de constantes agrupadas constan de dos o más constantes agrupadas para formar un registro. Estas inscripciones se emplean, a menudo, para describir una serie de constantes que han de almacenarse para su uso en una tabla. Las inscripciones con las que se escribe la descripción de un grupo de constantes, en la sección de Constantes, son exactamente iguales, en formato, a las que se utilizan para describir un registro en la sección de ficheros. Cualquier cláusula que pueda utilizarse en una inscripción de Descripción de Registro puede emplearse también para describir un campo en la sección de Constantes.

#### V-5.4. VALORES

Para una constante puede especificarse cualquier valor, ya sea numérico o no numérico, con las siguientes limitaciones:

- El tamaño del literal de la cláusula VALUE no puede exceder el SIZE o PICTURE del campo. Si el tamaño del literal es menor, se adoptan las reglas normales de alineación, o la almacenación a la derecha en el caso de utilizarse la cláusula JUSTIFIED RIGHT.
- El valor especificado no puede contradecir a la CLASS del campo.

#### V-5.5. UTILIZACION DE LA CONSTANT SECTION

En los compiladores que admiten las secciones WORKING-STORAGE SECTION y CONSTANT SECTION, el uso de la CONSTANT SECTION queda limitado a todas las inscripciones que poseen cláusula VALUE. Por consiguiente, las tablas de datos con valores fijos se establecen en esta sección describiéndose en la WORKING-STORAGE SECTION, las tablas que se generan en el desarrollo del programa.

En los compiladores que no utilicen esta sección, todas las inscripciones de la misma se realizan a la WORKING-STORAGE SECTION.

V - 6

SECCION COMUN DE  
ALMACENAMIENTO  
Y SECCION DE UNION



## V-6. SECCION DE ALMACENAMIENTO COMUN Y SECCION DE UNION

### V-6.1. GENERALIDADES

Estas dos secciones, que no figuran en todos los compiladores, se excluyen mutuamente. Su función es la de permitir el uso de subprogramas.

Un subprograma es un programa COBOL que comprende las cuatro divisiones habituales y que se compila separadamente con la ayuda de una ficha de control especial.

El estudio de los subprogramas se realiza en parte en la **PROCEDURE DIVISION** (ver verbo **ENTER**) y en parte en el apéndice A para cada uno de los distintos compiladores.

La **LINKAGE SECTION** (sección de Unión) o la **COMMON-STORAGE SECTION** (sección de Almacenamiento común) describen los datos que se toman de otro programa o las áreas de registros de etiquetas especificadas por el usuario.

Las anotaciones de descripción de registros efectuados en la sección de Unión proporcionan los nombres de datos con los que se pueden hacer referencia a áreas de datos reservados en memoria por otros programas. Estas anotaciones de la sección de Unión no reservan áreas en la memoria interna, ya que los datos se supone que existen en otra parte cualquiera de la memoria.

La sección de Unión es necesaria en cualquier programa en el que aparezca una cláusula **LABEL RECORDS**, con una opción nombre-dato o una declaración **ENTRY** con una opción **USING** (ver verbo **ENTER**). La sección de Unión sirve como un mecanismo para hacer referencia a datos entre un programa y el área de registros de etiquetas de usuario o entre un programa principal (que hace una llamada a una subrutina) y el programa subordinado (subrutina llamada por el principal). En el último caso, la sección de Unión está presente siempre en el programa llamado por el principal.

Se puede utilizar cualquier cláusula de descripción de registros para describir partidas en la sección de Unión, con tal de que se cumplan las siguientes condiciones:

1. Se satisfagan las reglas indicadas en la sección de Almacenamiento de trabajo y/o en la sección de Constantes relativas a almacenamiento contiguo y no-contiguo.
2. No se especifique la cláusula **VALUE** en otros niveles distintos del 88.
3. No se puede hacer referencia a los nombres de datos definidos en la sección Unión (exceptuando las áreas de registros de etiquetas definidas por el usuario) hasta que no se haya ejecutado la declaración **ENTRY** que contenga el nombre de dato.





V - 7

SECCION DE INFORMES



## V-7. SECCION DE INFORMES

### V-7.1. GENERALIDADES SOBRE IMPRESION DE INFORMES AUTOMATICOS (1)

La rutina de impresión de informes permite al programador especificar el formato de un fichero de salida, cuando éste ha de dar como resultado un informe impreso.

Cuando se utiliza esta rutina, las cabeceras o los pies de las páginas se describen una vez en la DATA DIVISION y se imprimen automáticamente al pasar el programa; los contadores de líneas impresas y de páginas escritas se incrementan automáticamente; también automáticamente cambian las páginas y se imprimen cabeceras y pies de página, se suman los datos de líneas y detalle que se desean acumular y se imprimen los totales. En otras palabras, una vez que el programador ha procesado en la PROCEDURE DIVISION, los datos de los ficheros de entrada, la rutina de impresión de informes introduce los datos en una cinta magnética (o cualquier otro soporte provisional), en el formato especificado, para impresiones off-line (2). También es posible imprimir el informe on-line (3).

Otro aspecto útil de la rutina de impresión de informe es el de poder generar varios informes dentro de un programa.

Se puede añadir a cada renglón de salida un código de identificación de un solo carácter que indica el informe al que pertenece.

De esta manera, la salida de varios informes puede realizarse en un solo fichero de salida, del que luego se seleccionará el informe que se desee, para su impresión off-line, por medio del código de identificación.

Las principales especificaciones de la rutina de impresión de informes se dan en la DATA DIVISION. Cuando se utiliza la rutina es necesaria una especificación en la FILE SECTION para listar los nombres de los informes que hayan de producirse, y se precisa una REPORT SECTION al final de la DATA DIVISION para definir esos informes.

En un programa que utilice la impresión de informes se leen registros y se procesan datos antes de llegar al informe. El informe se pro-

---

(1) Se incluye el estudio de esta sección en esta parte de la obra por motivos de estructuración. Sin embargo, se recomienda al lector efectuar, en principio, un estudio somero y volver a un estudio detallado una vez efectuado el de la PROCEDURE DIVISION.

(2) Fuera de línea o en tiempo diferido.

(3) En línea o al mismo tiempo que se produce.

duce por ejecución de las órdenes INITIATE, GENERATE y TERMINATE en la PROCEDURE DIVISION.

La orden INITIATE inicializa todos los contadores asociados a la rutina de impresión de informes, la GENERATE es empleada cada vez que se ha de producir una porción detallada del informe y la TERMINATE para finalizar el informe.

La rutina permite el proceso adicional de los datos por medio de la instrucción declarativa USE BEFORE REPORTING incluida en la porción de DECLARATIVES de la PROCEDURE DIVISION.

Cuando se emplee la rutina de impresión de informes hay que hacer una especificación en la FILE SECTION para listar los nombres de los informes que hayan de producirse e incluir la REPORT SECTION en la DATA DIVISION.

El nombre de cada uno de los informes que hayan de ser producidos por la rutina debe aparecer en una cláusula adicional denominada REPORT (S) en la FILE SECTION; en una descripción de fichero referente al fichero en que haya de producirse el informe.

El formato de la cláusula es:

<b>REPORT</b> [S] $\left[ \begin{array}{c} \text{IS} \\ \text{ARE} \end{array} \right]$ nombre-dato-1 nombre-dato-2...
------------------------------------------------------------------------------------------------------------------------

La presencia de dos o más nombre-dato en esta cláusula indica que el fichero contiene más de un informe. Estos informes pueden ser de diferentes tamaños, formatos, etc., y el orden en que aparezcan sus nombres en la cláusula carece de importancia.

## V-7.2. ESTRUCTURA DE LA REPORT SECTION

La REPORT SECTION contiene especificaciones del formato físico de cada página del informe y refleja la organización lógica del mismo. Se compone de tres distintos tipos de especificaciones que se describen a continuación: especificaciones de descripción de informe, especificaciones de descripción de grupo de informe y especificaciones de descripción de elemento de informe.

Las especificaciones de esta sección estipulan:

1. El máximo número de líneas que pueden aparecer en una página.
2. El contenido de las cabeceras y los pies de páginas, cuándo son necesarios y dónde han de aparecer.
3. El formato de los datos, la fuente de los datos y dónde han de aparecer.
4. Los datos que hayan de ser sumados y dónde y cuándo han de aparecer los totales.

### V-7.3. ESTRUCTURA DEL INFORME

La REPORT SECTION debe contener al menos una inscripción de descripción de informe (Report Description), señalada por un indicador de nivel especial denominado RD. Esta inscripción ha de ser la primera en la descripción de un informe; contiene información aplicable al conjunto del informe. Las descripciones de grupo de informe van a continuación del RD, clasificadas por tipo y definidas por medio de los números de nivel 01 y 02. Un grupo de informe se define por una inscripción a nivel 01 seguida por una o varias a nivel 02. La inscripción a nivel 01 indica el tipo de grupo de informe que se describe. Las descripciones a nivel 02 describen la información que se ha de imprimir; por ejemplo, al comenzar una nueva página se ha de imprimir un grupo de informe que constituya una cabecera. Las inscripciones de grupo de informe consisten en algunas cláusulas descritas en la DATA DIVISION y algunas otras propias de la REPORT SECTION.

### V-7.4. INSCRIPCIONES DE DESCRIPCION DE INFORMES

Una RD (Report Description) ha de ser la primera especificación de la descripción de un informe. Las especificaciones subsiguientes hasta la próxima RD o hasta el encabezamiento de la PROCEDURE DIVISION se consideran como parte del informe designado por la inscripción de descripción de informe. El formato general de esta descripción es:

RD nombre-dato [cláusula CODE] [cláusula CONTROL]  
[cláusula PAGE]

#### V-7.4.1. Cláusula CODE

[WITH **CODE** nombre-dato]

#### OBJETO

La cláusula CODE se utiliza para especificar un carácter de identificación que se añade al comienzo de cada línea escrita. La identificación tiene sentido cuando en un fichero se da salida a más de un informe.

#### PARTICULARIDADES DE SU USO

Nombre-dato es un nombre-nemotécnico; ha de ir asociado a un carácter simple en el párrafo SPECIAL-NAMES de la ENVIRONMENT DIVISION. Esta cláusula no se debe especificar si el informe se va a imprimir on-line.

#### V-7.4.2. Cláusula CONTROL

[**CONTROL**[S] [ IS  
ARE ] {opción} ]

**OBJETO**

La cláusula CONTROL especifica los controles que han de aplicarse al informe y los niveles de dichos controles. Un control es un campo de datos que se comprueba cada vez que se ha de imprimir un grupo de informe de detalle. Si la comprobación indica que el valor del campo de datos ha cambiado se dice que se ha producido una ruptura de control y se toman medidas especiales antes de imprimir la línea de detalle.

**OPCIONES**

Existen tres posibles opciones en la cláusula CONTROL:

<b>FINAL</b>
Nombre de datos...
<b>FINAL</b> nombre de datos...

La opción FINAL es el control de más nivel. Los niveles de los restantes controles quedan indicados por el orden en que estén escritos (FINAL es una excepción de la regla, según la cual los controles son campos de datos).

Cuando se comprueban los controles, el de más alto nivel que se ha especificado se comprueba primero, a continuación el de nivel inmediatamente inferior, etc. Cuando se produce una ruptura de control a un nivel determinado, dicha ruptura implica la de los niveles inferiores. Una ruptura de control para FINAL sólo se produce al comienzo y al final de un informe (esto es, antes de que se imprima la primera línea de detalle y después de que se imprima la última). Las medidas a tomar como resultado de una ruptura de control dependen de lo definido por el programador. Puede definir un grupo de cabecera y/o de pie, o bien ninguno de los dos, para cada control.

Las cabeceras y pies que se hayan definido se imprimen antes que la línea de detalle originalmente referenciable. Se imprimen en el orden siguiente: pie de menor nivel, pie de nivel inmediatamente superior, etc., hasta, e inclusive, el pie de nivel en que se haya producido la ruptura de control; a continuación, la cabecera de ese nivel, después la de nivel inmediatamente inferior, etc.; luego se imprime la línea de detalle. Si durante la impresión de cabeceras o pies se detecta un final de formulario, se saltará a la página siguiente y se imprimirá un pie y/o cabecera de página si se han especificado los correspondientes grupos de informe.

V-7.4.3. Cláusula PAGE

<b>PAGE</b> LIMIT[S] [ ^{IS} ARE ] núm.-entero LINE[S] [opción-1] [opción-2] [opción-3] [opción-4]
---------------------------------------------------------------------------------------------------------------

**OBJETO**

La cláusula PAGE se utiliza para describir el formato de una página de informe.

## OPCIONES

El límite del informe en número de líneas viene dado por el valor de número-entero-1.

Con la opción-1, que se especifica como: *HEADING* número-entero se indicará a través del valor de número-entero la primera línea en la que se pueda imprimir cualquier cosa. Ningún grupo de informe puede comenzar antes de la línea especificada por esta opción.

Con la opción-2, que se especifica como: *FIRST DETAIL* número-entero se indica a través de número-entero la primera línea en que se puede imprimir un grupo de detalle, una cabecera o un pie. Ningún grupo de detalle o de control puede comenzar antes de la línea especificada por número-entero. Si las cabeceras de informe o de página se extienden más allá de la línea especificada por número-entero, el primer grupo de detalle o de control de informe va a continuación de la última línea de cabecera.

Con la opción-3, que se especifica como: *LAST DETAIL* número-entero se indica a través de número-entero la última línea en que se puede imprimir un detalle o una cabecera de control.

Con la opción-4, que se especifica como *FOOTING* número-entero se indica la última línea en que se puede imprimir un pie de control. Los pies de página siguen a la línea FOOTING pero no sobrepasan la línea LIMITE.

Si se omite la cláusula PAGE, se presuponen, en algunos compiladores, los valores siguientes de número-entero:

- Para opción 1 = 1.
- Para opción 2 = 1.
- Para opción 3 = 48.
- Para opción 4 = 48.
- Para el límite = 52.

Si se especifica la cláusula PAGE, pero se omite uno o varios de los números-enteros de las opciones, se suponen los siguientes valores (ha de especificarse el límite):

- Número-entero, opción 1 = 1 o el valor especificado.
- Número-entero, opción 2 = número-entero opción 1 o el valor especificado.
- Número-entero, opción 4 = límite o el valor especificado.
- Número-entero, opción 3 = número-entero opción 4 o el valor especificado.

### V-7.5. INSCRIPCIONES DE GRUPO DE INFORME Y DE ELEMENTO DE INFORME

Un grupo de informe es una línea o grupo de líneas de un informe y se imprime bajo determinadas condiciones; por ejemplo, una cabecera de página se imprime al comienzo de cada página, y un pie de informe, al final del informe.

Un grupo de informe se describe a nivel 01 y va seguido por una o varias descripciones a nivel 02. La descripción a nivel 01 indica el tipo de grupo de informe de que se trata. La descripción a nivel 02 indica la información que ha de ser impresa cuando las condiciones pertinentes señalan que se debe dar salida a un grupo de informe.

Las cláusulas que componen las inscripciones a nivel 01 y 02 son:

CLAUSULA	NIVEL	FORMATO
TYPE	01	<u>TYPE</u> [ opción-1 {opción-2} ]
LINE	01 y 02	[ <u>LINE</u> {opción} ]
NEXT GROUP	01	[ <u>NEXT GROUP</u> {opción} ]
COLUMN	02	[ <u>COLUMN</u> número-entero ]
GROUP	02	[ <u>GROUP</u> INDICATE ]
SOURCE	02	[ <u>SOURCE</u> nombre-datos ]
SUM	02	[ <u>SUM</u> {opción-1} [ <u>RESET ON</u> {opción-2} ] ]

Además de las cláusulas BLANK, JUSTIFIED, PICTURE y VALUE que se han estudiado en la FILE SECTION y que tienen utilización en la REPORT SECTION.

#### V-7.5.1. Cláusula TYPE

TYPE { opción-1 {opción-2} }

**OBJETO**

La cláusula TYPE especifica el tipo de grupo de informe a describir en función de la opción-1 elegida.

**OPCIONES**

Las opciones que indica el tipo de informe son:

OPCION-1	ABREVIATURA
REPORT HEADING	RH
PAGE HEADING	PH
CONTROL HEADING	CH*
DETAIL	DE
CONTROL FOOTING	CF*
PAGE FOOTING	PF
REPORT FOOTING	RF

(*) Admite la opción-2.



Con la opción *REPORT HEADING* el grupo de informe se imprime sólo una vez, al comienzo del informe. Se imprime al ejecutarse la primera orden *GENERATE* referente al informe. Para un informe solamente se puede definir un grupo de tipo *REPORT HEADING*.

Con la opción *PAGE HEADING* se imprime el grupo de informe al comienzo de cada página. Para un informe solamente se puede definir un grupo de tipo *PAGE HEADING*.

Con la opción *CONTROL HEADING* el grupo de informe se imprime cada vez que se produce una ruptura en el control especificado, opción-2 (*FINAL* o nombre-dato). Se puede definir un grupo de tipo *CONTROL HEADING* por cada control especificado en la cláusula *CONTROL* de la inscripción de informe.

Con la opción *DETAIL*, el grupo de informes se imprime cada vez que se efectúa una orden *GENERATE* referente al grupo de informe.

Con la opción *CONTROL FOOTING* el grupo de informe se imprime cada vez que se produce una ruptura en el control especificado, opción-2 (*FINAL* o nombre-dato). Se puede definir un grupo de tipo *CONTROL FOOTING* por cada control especificado en la cláusula *CONTROL* de la inscripción de informe.

Con la opción *PAGE FOOTING* el grupo de informe se imprime al final de cada página. Para un informe, solamente se puede definir un grupo de tipo *PAGE FOOTING*.

Con la opción *REPORT FOOTING* al grupo de informe se imprime el final del informe. Sólo se imprime cuando se ejecuta la orden *TERMINANTE*.

#### V-7.5.2. Cláusula **LINE**

[ **LINE** {opción} ]

La cláusula *LINE* especifica el renglón en que se han de imprimir todos los datos definidos hasta la siguiente cláusula *LINE* o hasta el final de la descripción del grupo de informe.

Existen tres posibilidades:

- Número-entero.
- *PLUS* número-entero.
- *NEXT PAGE*.

Con la opción *número-entero* se especifica el número de renglón absoluto en que se han de imprimir los datos definidos.

Con la opción *PLUS* número-entero se especifica el número de renglones que hay que saltar antes de imprimir los datos.

Con la opción *NEXT PAGE* se indica que el grupo de datos que se define debe aparecer en la página siguiente. Esta opción debe especificarse solamente en la cláusula referente a la primera línea de un

**OBJETO**

**OPCIONES**

grupo de informe. Hace que se impriman el pie y la cabecera de página si se han especificado.

**PARTICULARIDADES  
DE SU USO**

Si se ha especificado más de una cláusula **LINE** dentro de la definición de un grupo de informe, hay que tener en cuenta las siguientes restricciones:

- Los números de línea absolutos han de especificarse en orden ascendente.
- Un número de línea absoluto no debe seguir a uno relativo.

### V-7.5.3. Cláusula **NEXT GROUP**

[ **NEXT GROUP** {opción} ]

**OBJETO**

La cláusula **NEXT GROUP** especifica un control de línea después de la impresión del grupo de informe que se esté definiendo.

Existen las mismas posibilidades que en la cláusula **LINE**.

**OPCIONES**

Con la opción *número-entero* se indica la línea absoluta a la que se ha de saltar.

Con la opción *PLUS número-entero* se indica la línea relativa a la que se ha de saltar.

Con la opción *NEXT PAGE* se indica un salto a la página siguiente. Si se han especificado, se imprimen pie y cabecera de página.

### V-7.5.4. Cláusula **COLUMN**

[ **COLUMN** número-entero ]

**OBJETO**

La cláusula **COLUMN** especifica la posición que ocupan en la línea de impresión los datos que se están definiendo.

*Número-entero* especifica la posición de la línea de impresión en que ha de estar situado el carácter extremo izquierdo de los datos definidos por la cláusula. La posición extrema de la izquierda de la línea de impresión es la columna 1.

**PARTICULARIDADES  
DE SU USO**

Si no se especifica la cláusula **COLUMN** no se imprimen los datos que se definen en la inscripción.

### V-7.5.5. Cláusula **GROUP INDICATE**

[ **GROUP** INDICATE ]

**OBJETO**

La cláusula **GROUP INDICATE** especifica que los datos de la inscripción que se define han de imprimirse solamente después de una ruptura de control o de un cambio de página.

**PARTICULARIDADES  
DE SU USO**

Esta cláusula sólo debe especificarse dentro de una descripción de grupo de informe de tipo **DETAIL**.

**V-7.5.6. Cláusula SOURCE**

[ **SOURCE** nombre-dato ]

**OBJETO**

La cláusula **SOURCE** indica la fuente de los datos, especificada por nombre-dato, que han de ser transferidos al campo de datos que se esté definiendo.

**V-7.5.7. Cláusula SUM**

[ **SUM** {opción-1} [ **RESET ON** {opción-2} ] ]

**OBJETO**

La cláusula **SUM** se utiliza para producir la suma automática de los datos de un grupo de informe. Produce el mismo efecto que si el programador hubiese calculado la suma a través de un verbo de la **PROCEDURE DIVISION** y se hubiese referido a ella por medio de la cláusula **SOURCE**.

**OPCIONES**

Las opción-1 posibles son:

- Nombre-dato-1 ..... **UPON** nombre-dato-2.
- Nombre-dato-3.

Los nombres de datos que se describen a continuación de **SUM** son los nombres de los campos que se han de sumar. Pueden estar definidos en la **FILE SECTION**, **WORKING-STORAGE SECTION**, **CONSTANT SECTION** o **LINKAGE SECTION** o tratarse de nombres de contadores definidos en la propia **REPORT SECTION**.

Cada vez que se ejecuta una orden **GENERATE** referida a un grupo de informe de tipo **DETAIL**, el contenido de nombre de dato-1 se suma a un contador.

Con la opción **UPON** esta suma solamente se realiza cuando se ejecuta una orden **GENERATE** que haga referencia a nombre-dato-2.

Después de haberse impreso el grupo del informe que contiene un contador, éste se restaura a cero, a menos que exista la opción **RESET** que haga referencia a él. Cuando existe esta opción se suprime la restauración automática del contador y éste se restaura a cero únicamente en el caso de producirse una ruptura de control especificada por opción-2 que puede ser **FINAL** o un nombre-dato que ha de ser el nombre de un control.

## V-7.6. CONTADORES GENERADOS POR LA RUTINA DE IMPRESION DE INFORMES

El propio compilador genera a través de la rutina de impresión de informes diversos contadores y switches de los que puede hacer uso el programador en la PROCEDURE DIVISION del programa.

Estos contadores y switches son:

- **LINE-COUNTER** o contador de renglones.
- **PRINT-SWITCH** de impresión.
- **CBL-CTR** o contador de nivel de ruptura de control.

*LINE-COUNTER* lleva la cuenta de los renglones impresos y consecuentemente se restaura para cada cambio de página. Como pueden existir diversos informes en el mismo programa, cuando se utilice deberá ir acompañado de un calificador que indique el informe específico al cual se hace referencia.

*PRINT-SWITCH* proporciona al programador la posibilidad de suprimir la impresión de un grupo de informe. Se genera un único **PRINT-SWITCH** por programa. El valor de **PRINT-SWITCH** se restaura después de cada ciclo de impresión.

*CBL-CTR* se utiliza para dos funciones distintas. Cuando el grupo de informe **CONTROL-FOOTING** contiene una cláusula **SOURCE** en la que el nombre-dato utilizado es el mismo que el especificado en la cláusula **CONTROL** y se desea imprimir el valor de nombre-dato de **CONTROL** antes que el valor que produjo la ruptura, se inicializa **CBL-CTR** con el valor 1 antes de haberse generado el primer detalle y después de haberse iniciado el informe. Si no se transfiere un 1 a **CBL-CTR** en el momento especificado, se imprimirá el valor de nombre-dato de **CONTROL**. La segunda función de **CBL-CTR** es la de permitir al programador su consulta para determinar el nivel de la ruptura de control en un momento determinado.

001010 IDENTIFICATION DIVISION.  
001020 PROGRAM-ID. INFORMEV  
001030 DATE-WRITTEN. FEB-75.  
001040 REMARKS. ESTE PROGRAMA ESTA REDACTADO COMO COMPLEMENTO  
001050 AL CAPITULO V-7 PARA EXPLICAR COMO SE UTILIZAN  
001060 LAS DISTINTAS CLAUSULAS DE LA REPORT SECTION EN  
001070 UN EJEMPLO FICTICIO QUE OBTIENE EL RESUMEN DE  
001080 LAS VENTAS DE UNA EMPRESA MANUFACTURERA.  
001090 ENVIRONMENT DIVISION.  
001100 CONFIGURATION SECTION.  
001110 SOURCE-COMPUTER. UNIVAC-1108.  
001120 OBJECT-COMPUTER. UNIVAC-1108.  
001130 INPUT-OUTPUT SECTION.  
001140 FILE-CONTROL.  
001150 SELECT ENTRADA ASSIGN TO UNISERV.  
001160 SELECT SALIDA ASSIGN TO PRINTER.  
001170 DATA DIVISION.  
001180 FILE SECTION.  
001190 FD ENTRADA BLOCK CONTAINS 20 RECCRDS LABEL RECORD IS OMIT  
001200 DATA RECORD IS REG-ENT.  
001210 01 REG-ENT.  
001220 02 DELEGACION PICTURE X(8).  
001230 02 PROVINCIA PICTURE X(9).  
001240 02 CIUDAD PICTURE X(8).  
001250 02 PRODUCTO-A PICTURE 9(4).  
002010 02 PRODUCTO-B PICTURE 9(4).  
002020 02 PRODUCTO-C PICTURE 9(4).  
002030 02 FILLER PICTURE X(42).  
002040 FD SALIDA LABEL RECORD IS OMITTED REPORT IS VENTAS.  
002050 WORKING-STORAGE SECTION.  
002060 77 TOTAL-P PICTURE 9(5).  
002070 REPORT SECTION.  
002080 RD VENTAS. INDICADOR DE NIVEL DE DESCRIPCION DE INFORME  
002090 CONTROL ARE FINAL, DELEGACION, PROVINCIA.  
002100 PAGE LIMIT IS 62 LINES.  
002110 HEADING 4, FIRST DETAIL 10, LAST DETAIL 40, FOOTING 50

INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V

D

---

Inscripciones de descripción de los campos componentes de los registros del fichero de entrada

El formato de los registros es:

Delegación	Provincia	Ciudad	prod. A	prod. B	prod. C	
1	9	18	26	30	34	38

---

INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V  
INFORME V

TOTAL-P

Area de memoria de trabajo destinada a contener la suma de los contenidos de los campos PRODUCTO-A, PRODUCTO-B, PRODUCTO-C

Con esta cláusula se indica que los totales parciales se obtienen a nivel de PROVINCIA, DELEGACION y FINAL (TOTAL GENERAL)

Con esta cláusula se especifica que el máximo número de líneas por página del informe es de 62

Con las opciones utilizadas se indica que la primera línea de imposición es la 4.ª; la primera línea de detalle, la 10.ª; la última línea de detalle, la 40.ª; y que la última línea en que se puede imprimir un total parcial es la 50.ª

002120 01 TYPE IS REPORT HEADING NEXT GROUP IS NEXT PAGE.  
002130 02 LINE 5 COLUMN 31 PIC X(23) VALUE 'RESUMEN GENERAL VE  
002140 02 LINE 6 COLUMN 33 PIC X(20) VALUE 'PROD. MANUFACTURAD  
002150 02 LINE 7 COLUMN 34 PIC X(10) VALUE 'ENERO 1975'.  
002160 02 LINE 35 COLUMN 29 PIC X(19) VALUE 'DIRECCION COMERCI  
002170 02 LINE 36 COLUMN 29 PIC X(19) VALUE 'AVDA.....  
002180 02 LINE 37 COLUMN 29 PIC X(6) VALUE 'MADRID'.  
002190 01 TYPE IS PAGE HEADING.  
002200 02 LINE PLUS 1 COLUMN 31 PIC X(22) VALUE 'RESUMEN GENER  
002210- 'TAS'.  
002220 02 LINE PLUS 1 COLUMN 35 PIC X(10) VALUE 'ENERO 1975'  
002230 02 LINE PLUS 3.  
002240 03 COLUMN 38 PIC X(10) VALUE 'PRODUCTO-A'  
002250 03 COLUMN 49 PIC X(10) VALUE 'PRODUCTO-B'  
003010 03 COLUMN 63 PIC X(10) VALUE 'PRODUCTO-C'.  
003020 03 COLUMN 74 PIC X(7) VALUE 'TOTAL-P'.  
003030 01 TYPE IS CONTROL HEADING DELEGACION.  
003040 02 LINE PLUS 2.  
003050 03 COLUMN 1 PIC X(13) VALUE 'DELEGACION DE'.  
003060 03 COLUMN 15 PIC X(8) SOURCE DELEGACION.  
003070 02 LINE PLUS 1 COLUMN 1 PIC X VALUE ' '  
003080 01 LINEA-D TYPE IS DE LINE PLUS 1.  
003090 02 COLUMN 20 PIC X(9) SOURCE PROVINCIA GROUP INDICATE.  
003100 02 COLUMN 30 PIC X(8) SOURCE CIUDAD  
003110 02 COLUMN 41 PIC Z,999 SOURCE PRODUCTO-A.  
003120 02 COLUMN 51 PIC Z,999 SOURCE PRODUCTO-B.  
003130 02 COLUMN 63 PIC Z,999 SOURCE PRODUCTO-C.  
003140 02 COLUMN 76 PIC ZZ,999 SOURCE TOTAL-P.  
003150 01 TYPE IS CONTROL FOOTING PROVINCIA NEXT GROUP PLUS 1.  
003160 02 LINE PLUS 1 COLUMN 41 PIC X(5) VALUE '-----'.  
003170 02 COLUMN 51 PIC X(5) VALUE '-----'.  
003180 02 COLUMN 63 PIC X(5) VALUE '-----'.  
003190 02 COLUMN 75 PIC X(6) VALUE '-----'.  
003200 02 LINE PLUS 1.  
003210 03 COLUMN 17 PIC X(9) SOURCE IS PROVINCIA.  
003220 03 COLUMN 27 PIC X(13) VALUE 'TOTAL.....'  
003230 03 TO-PROA COLUMN 40 PIC ZZ,999 SUM PRODUCTO-A.  
003240 03 TO-PRCB COLUMN 50 PIC ZZ,999 SUM PRODUCTO-B.  
003250 03 TO-PROC COLUMN 62 PIC ZZ,999 SUM PRODUCTO-C.  
004010 03 TO-TOTP COLUMN 74 PIC ZZZ,999 SUM TO-PROA TO-PROB  
004020 TC-PROC.  
004030 01 TYPE IS CONTROL FOOTING DELEGACION NEXT GROUP NEXT PAG  
004040 02 LINE PLUS 2.  
004050 03 COLUMN 1 PIC X(8) SOURCE DELEGACION.  
004060 03 COLUMN 10 PIC X(30) VALUE 'TOTAL DELEGACION.....  
004070- '.....'  
004080 03 R-PROA COLUMN 40 PIC Z9,999 SUM TO-PROA.  
004090 03 R-PRCB COLUMN 50 PIC Z9,999 SUM TO-PROB.  
004100 03 R-PRCC COLUMN 62 PIC Z9,999 SUM TO-PROC.  
004110 03 R-PROT COLUMN 74 PIC Z99,999 SUM TO-TOTP.  
004120 02 TOT-F PIC Z99,999 SUM R-PROA R-PRCB R-PROC.  
004130 01 TYPE IS CONTROL FOOTING FINAL.  
004140 02 LINE PLUS 4.  
004150 03 COLUMN 1 PIC X(36) VALUE 'TOTAL GENERAL ENERO....  
004160 '.....'.  
004170 03 COLUMN 39 PIC Z99,999 SUM R-PROA.  
004180 03 COLUMN 49 PIC Z99,999 SUM R-PRCB.  
004190 03 COLUMN 61 PIC Z99,999 SUM R-PROC.  
004200 03 COLUMN 72 PIC Z,999,999 SUM R-PROT.  
004210 01 TYPE IS PAGE FOOTING LINE 51.  
004220 02 COLUMN 38 PIC X(6) VALUE 'PAGINA'.  
004230 02 COLUMN 45 PIC 99 SOURCE PAGE-COUNTER.  
004240 01 TYPE IS REPORT FOOTING LINE 52.  
004250 02 COLUMN 36 PIC X(15) VALUE 'FIN DEL INFORME'.

INFORMEV S'. INFORMEV • INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión, por una sola vez, al comienzo del informe del grupo de literales figurados. En la siguiente separata se figuran las líneas impresas correspondientes, marcadas con la referencia ①
VEN INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión, cada vez que se produce un cambio de página, del grupo de literales figurados. se corresponden con la referencia ② del informe.
INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión, cada vez que se produce un cambio de control por el campo DELEGACION, es decir, cada vez que cambia el contenido de dicho campo y después de imprimirse los totales referidos a dicho campo. Se corresponde con la referencia ③ del informe.
INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión de tantas líneas de detalle como conjuntos de registros existan con el mismo valor en el campo CIUDAD. Se corresponden con las referencias ④ del informe. Se imprimen, por consiguiente, para cada cambio de valor del campo PROVINCIA y después de imprimirse los totales de dicho campo o los totales del campo DELEGACION y su correspondiente línea de referencia ③
INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión de una línea de total para cada cambio de valor del campo PROVINCIA y refleja el total de líneas de detalle de los registros correspondientes a la misma PROVINCIA. Previamente se imprime una línea de guiones con el formato indicado en la línea de referencia ⑤ del informe. El campo TO-PROA es = $\sum$ campos PRODUCTO-A para el mismo valor de provincia y análogamente para los campos TO-PROB, TO-PROC y TO-TOTP.
INFORMEV INFORMEV INFORMEV ••••• INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión de una línea de total para cada cambio de valor del campo DELEGACION y refleja el total de todas las líneas de referencia ⑤ existentes detrás de la anterior línea de total de DELEGACION editada. Se corresponden con la referencia ⑥ del informe. El campo R-PROA es = $\sum$ campos PRODUCTO-A para el mismo valor de delegación y análogamente para los campos R-PROB, R-PROC, R-PROT.
INFORMEV INFORMEV ••••• INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión de una línea, una sola vez en el informe, cuando se han procesado todos los registros. Refleja el total final o resumen general. Se corresponden con la referencia ⑦ del informe.
INFORMEV INFORMEV INFORMEV	Estas inscripciones producen la impresión de una línea, al final de cada página del informe, editándose el contenido del contador PAGE-COUNTER. Se corresponden con la referencia ⑧ del informe.
INFORMEV INFORMEV	Estas inscripciones producen la impresión de una línea al final del informe con el literal signado. Se corresponden con la referencia ⑨ del informe.





---

# CAPITULO VI

LAS DIVISIONES DE UN PROGRAMA :

PROCEDURE DIVISION

---

INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV  
 INFORMEV

Esta división, como se estudia en el capítulo VI, es la que contiene los verbos que generan el informe definido en la REPORT SECTION.

→ Inicia el proceso de generación del informe de VENTAS.

→ Genera las líneas de datos.

→ Termina el proceso de generación del informe.

7273747576				
			①	Líneas producidas por el grupo de inscripciones correspondientes a las líneas de codificación 002120 a 002180.
			②	Estas líneas se producen cada vez que existe un cambio de página. Se corresponden con el grupo de inscripciones codificadas en las líneas 002190 a 003020.
				LÍNEA INICIAL + 4
TO-C	TOTAL-P		③	Esta línea se produce cada vez que existe un cambio de contenido del campo DELEGACION, variando cada vez el literal editado en correspondencia con el contenido de dicho campo. Se corresponde con el grupo de inscripciones codificadas en las líneas 003030 a 003070.
	8,598		④	Este grupo de líneas se producen para cada conjunto de registros que tuviera el mismo contenido en el campo PROVINCIA. El número de líneas variará con los conjuntos de registros que tengan el mismo contenido en el campo CIUDAD, es decir, una línea de detalle para cada conjunto de registros con el mismo valor del campo CIUDAD.
	4,940			
	1,545			
	15,083		⑤	Estas dos líneas se producen inmediatamente después de cada grupo de líneas de referencia ④, reflejando el total de los registros con el mismo valor en el campo PROVINCIA.
	5,050		④	
	4,790		⑤	
	9,840		④	
	1,370		⑤	
	3,500		④	
	4,870		⑤	
	2,200		④	
	4,400		⑤	
	6,600		④	
	13,200		⑤	
	42,993		⑥	Esta línea se produce cada vez que hay un cambio en el valor del campo DELEGACION y refleja el total de los registros con el mismo valor en dicho campo. Se corresponden con el grupo de inscripciones codificadas en las líneas 004030 a 004120.
	260,220		⑦	Esta línea se produce una sola vez en el informe al final del mismo y una vez procesados todos los registros. Se corresponden con el grupo de inscripciones codificadas en las líneas 004130 a 004200.
			⑧	Esta línea se produce cada vez que existe un cambio de página.
			⑨	Esta línea se produce únicamente al final del informe. Se corresponden con el grupo de inscripciones codificadas con las líneas 004240 a 004250.



---

**C O N T E N I D O**

- VI-1. GENERALIDADES DE LA  
PROCEDURE DIVISION
  - VI-2. VERBOS DE  
ENTRADA-SALIDA
  - VI-3. VERBOS DE  
MANIPULACION  
Y TRANSFERENCIA  
DE DATOS
  - VI-4. VERBOS ARITMETICOS
  - VI-5. VERBOS DE CONTROL  
DE SECUENCIA
  - VI-6. VERBOS  
DEL COMPILADOR
  - VI-7. VERBO DE CONTROL  
DE ENTRADA-SALIDA
-



VI - 1

GENERALIDADES DE LA  
PROCEDURE DIVISION





## VI-1. GENERALIDADES DE LA PROCEDURE DIVISION

### VI-1.1. OBJETO DE LA DIVISION

La División de Procedimientos especifica los órdenes o instrucciones COBOL propiamente dichas, que se deben establecer para resolver un problema dado.

Los procedimientos contenidos en esta División son cálculos, decisiones lógicas, órdenes de entrada-salida, etc., expresadas mediante declaraciones significativas, similares al inglés, empleándose *verbos* para indicar acciones y declaraciones condicionales para describir las distintas alternativas que se pueden seguir al tomar una decisión.

Esta División es ordinariamente conocida como «el programa», por ser la parte del mismo que establece la secuencia de operaciones a realizar, aunque no es suficiente por sí misma para describir el problema total, ya que para ejecutar éste se necesita hacer repetidas referencias a informaciones que están contenidas en las otras divisiones.

### VI-1.2. SINTAXIS DE LA PROCEDURE DIVISION

El formato de la División de Procedimientos es libre. Debe comenzar siempre por el encabezamiento PROCEDURE DIVISION, seguida de punto y espacio.

La estructura de esta División consta de cuatro unidades, repetidas las veces que sean necesarias en el conjunto total.

El orden jerárquico de estas unidades es:

- SECCION.
- PARRAFO.
- SENTENCIA.
- DECLARACION.

Cuyo significado y cometido se estudió en el capítulo II.

### VI-1.3. VERBOS INTEGRANTES DE LA PROCEDURE DIVISION

Los distintos verbos que componen la sintaxis de la División de Procedimientos son:

### **Verbos de entrada-salida**

- OPEN.
- READ.
- WRITE.
- REWRITE (1).
- CLOSE.
- ACCEPT.
- DISPLAY.
- SEEK (1).
- DELETE (1).

### **Verbos de manipulación de datos**

- MOVE.
- EXAMINE.
- TRANSFORM (1).

### **Verbos aritméticos**

- ADD.
- SUBTRACT.
- MULTIPLY.
- DIVIDE.
- COMPUTE.

### **Verbos de control de secuencia**

- GO TO.
- ALTER.
- PORFORM.
- STOP.
- IF (2).

### **Verbos de compilador**

- EXIT (3).
- ENTER.
- NOTE.

### **Verbo de clasificación**

- SORT (4).

### **Verbo de control de entrada-salida (verbo declarativo)**

- USE.

---

(1) No existentes en todos los compiladores.

(2) Aunque la palabra IF no es propiamente un verbo en sentido estricto, posee una de las características más importantes de los verbos, a saber, la de generar codificación en el programa objeto.

(3) Se estudia dentro de los verbos complementarios de control de secuencia.

(4) Se estudia en apéndice B.

VI-2

VERBOS DE ENTRADA-SALIDA



## VI-2. VERBOS DE ENTRADA-SALIDA

### VI-2.1. GENERALIDADES

Los verbos de entrada-salida son necesarios, por una parte para introducir los datos del problema desde el soporte exterior (ficheros de entrada) a la memoria central y por otra para extraer los resultados desde la memoria central a los periféricos exteriores (ficheros de salida).

El sistema de control del ordenador es el encargado de asegurar estas uniones entre la memoria central y los periféricos de entrada-salida a través de los canales de comunicación.

En estas transferencias de información se comprueba la validez de los datos, el agrupamiento y desglose de bloques, la alternación de las unidades de banda magnética, etc., de forma automática, sin que sea necesaria por parte del programador otra aportación que la adecuada elección de los verbos necesarios en cada momento del proceso.

Por consiguiente, para cada fichero se atribuye una zona de la memoria central que es:

- Para los ficheros de entrada, una zona denominada de lectura cuya dimensión corresponde a la descripción del registro realizada en la DATA DIVISION.
- Para los ficheros de salida, una zona denominada de escritura cuya dimensión corresponde a la descripción del registro realizada en la DATA DIVISION.

Además de los datos organizados en forma de ficheros, existe normalmente la necesidad de proporcionar al ordenador o de obtener del mismo datos y resultados que por sus características especiales no deben o no se pueden organizar en ficheros; generalmente el volumen de estos datos o resultados es muy pequeño.

Por consiguiente, existen dos tipos o categorías de verbos de entrada-salida:

- Verbos para tratar datos y resultados no organizados en ficheros y por consiguiente sin descripción FD en la DATA DIVISION (generalmente, poco volumen de información).
- Verbos para tratar datos y resultados organizados en ficheros (con descripción FD en la DATA DIVISION) (normalmente, gran volumen de información).

## VI-2.2. VERBOS QUE TRATAN INFORMACION NO ORGANIZADA EN FICHEROS

### VI-2.2.1. Verbo ACCEPT

**OBJETO**

El verbo ACCEPT permite leer en el curso de la ejecución del programa una pequeña cantidad de datos no organizados en ficheros.

Normalmente este verbo se utiliza para admitir a través de la consola, que maneja el operador, algún dato aislado como puede ser la fecha en la que se realiza el proceso.

**FORMATO**

El formato de este verbo es:

**ACCEPT** nombre-dato [ **FROM** { CONSOLE  
nombre-nemotécnico } ]

**PARTICULARIDADES DE SU USO**

En su forma más simple, ACCEPT nombre-dato permite obtener un registro de longitud determinada a partir de un periférico definido por el constructor, normalmente la lectora de fichas. En este caso la longitud máxima del registro obtenido sería de 80 caracteres que son transferidos a la zona de la memoria central que haya sido definida, obligatoriamente, en la WORKING-STORAGE SECTION con nombre-dato.

Si nombre-dato se ha definido con un tamaño de 80 caracteres [SIZE 80 o PICTURE X(80)], toda la información contenida en la ficha se transfiere a nombre-dato. Si nombre-dato se ha descrito con un tamaño inferior a 80 caracteres se transferirán solamente los caracteres de la izquierda de la ficha en longitud igual al tamaño definido por nombre-dato. Si nombre-dato se ha descrito con un tamaño superior a 80 caracteres se transfieren los 80 caracteres contenidos en la ficha completándose el resto del espacio reservado con blancos (ver figura VI-1).

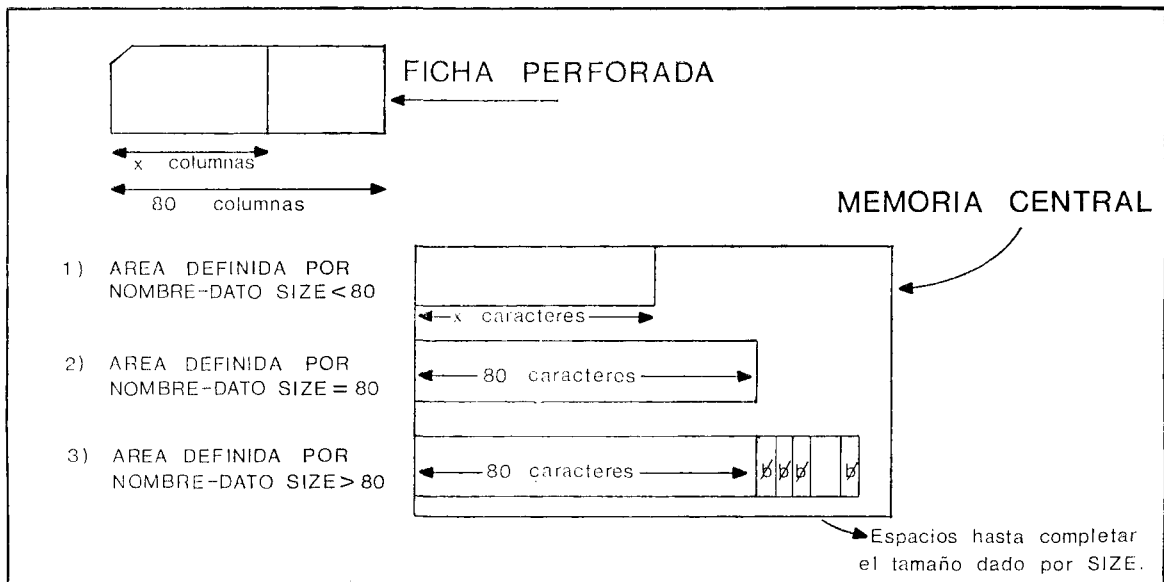


FIGURA VI-1. Efecto del verbo ACCEPT sin la opción FROM

**OPCIONES**

El verbo ACCEPT permite, opcionalmente, utilizar como elemento de comunicación la CONSOLA o cualquier otro periférico de entrada.

Con la forma ACCEPT nombre-dato FROM CONSOLE se transfiere al área definida por nombre-dato toda la información que el operador teclea en la máquina de escribir o consola de mando del equipo, si bien el número de caracteres que se pueden transmitir varía con el tipo de ordenador.

Con la forma ACCEPT nombre-dato FROM nombre-nemotécnico se permite efectuar la transmisión de la información desde el equipo periférico que designe nombre-nemotécnico en el párrafo SPECIAL-NAMES de la ENVIRONMENT DIVISION (ver epígrafe IV-2.2.3.4).

**EJEMPLOS**

ACCEPT FECHA  
ACCEPT FECHA FROM CONSOLE  
ACCEPT FECHA FROM LECTORA

Estas sentencias permiten almacenar en el campo FECHA definido en la WORKING-STORAGE SECTION (por ejemplo, a nivel 77 FECHA PICTURE XXXXXX) el valor que se haya perforado en las seis primeras columnas de una ficha para el primer y tercer ejemplo (si LECTORA es el nombre nemotécnico asignado a la lectora de fichas) y por el teclado en la consola de mando para el caso del segundo ejemplo.

**VI-2.2.2. Verbo DISPLAY****OBJETO**

El verbo DISPLAY permite extraer, desde la memoria central a un periférico, determinada cantidad de resultados.

**FORMATO**

El formato de este verbo es:

$$\text{DISPLAY } \left\{ \begin{array}{l} \text{nombre-dato-1} \\ \text{literal-1} \end{array} \right\} \left[ \left[ \begin{array}{l} \text{nombre-dato-2} \\ \text{literal-2} \end{array} \right] \dots \left[ \text{UPON} \right. \right. \\ \left. \left. \left\{ \begin{array}{l} \text{CONSOLE} \\ \text{nombre-nemotécnico} \end{array} \right\} \right] \right]$$
**PARTICULARIDADES DE SU USO**

En su forma más simple, DISPLAY  $\left\{ \begin{array}{l} \text{nombre-dato} \\ \text{literal} \end{array} \right\}$  permite imprimir sobre la impresora del sistema la información contenida en el área definida por nombre-dato o el valor del literal especificado en el formato.

Nombre-dato-1 puede ser cualquier campo de dato definido en la FILE SECTION o en la WORKING-STORAGE SECTION.

Literal-1 puede ser cualquier tipo de literal numérico, no-numérico e incluso una constante figurativa.

**OPCIONES**

El número de campos, representados por nombre-dato-2, nombre-dato-3, ... así como el número de literales que pueden figurarse, opcionalmente, dentro de la misma instrucción DISPLAY está doblemente condicionado. Por una parte al tamaño o longitud del registro de salida y por otra a la característica del compilador empleado. En general, el número total de caracteres que pueden ser exhibidos a través de una

instrucción DISPLAY es de 72 para salidas sobre consola o ficha perforada y 120 para la impresora, bien sea por medio de un nombre-dato, bien como suma de las distintas longitudes de cada uno de los nombre-dato o literales expresados en la instrucción.

La opción UPON es totalmente análoga a la opción FROM del verbo DISPLAY y produce los mismos efectos ya comentados en el estudio de dicho verbo.

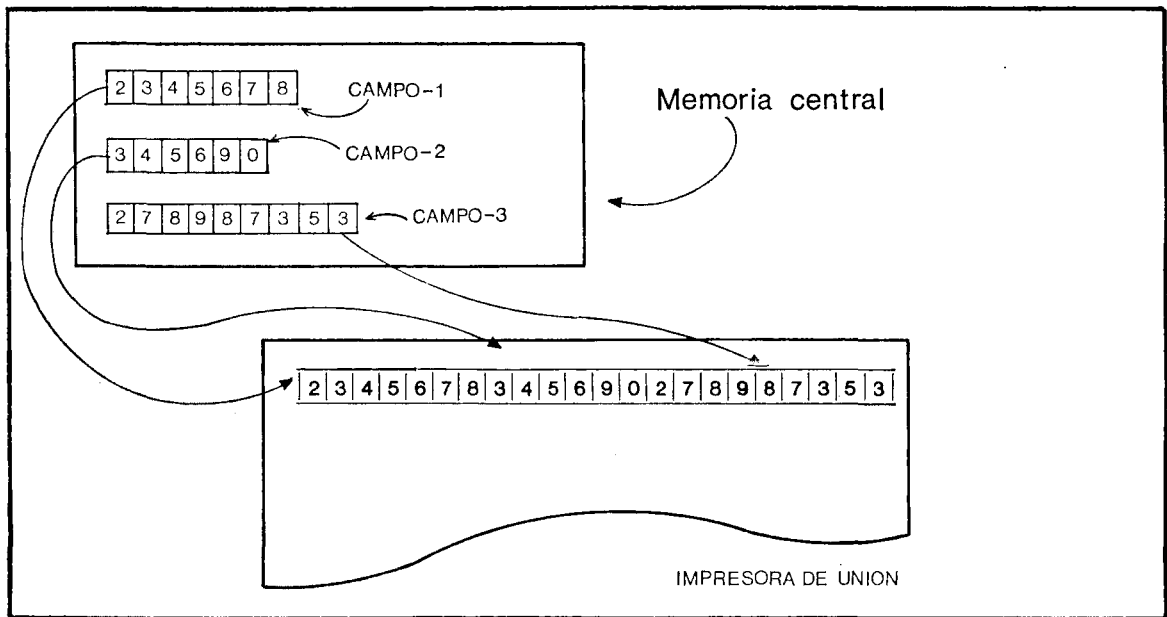


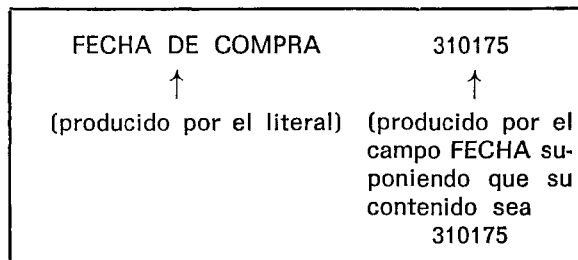
FIGURA VI-2. Efecto del verbo DISPLAY sin la opción UPON

**EJEMPLOS**

DISPLAY CAMPO-1 CAMPO-2 CAMPO-3

Producirá sobre la impresora una línea impresa con el contenido de CAMPOS-1, CAMPOS-2 y CAMPO-3 sin solución de continuidad (ver figura VI-2).

DISPLAY 'FECHA DE COMPRA' FECHA UPON CONSOLE emitirá sobre la consola del equipo un mensaje con el siguiente contenido:





## VI-2.3. VERBOS QUE TRATAN INFORMACION ORGANIZADA EN FICHEROS

### VI-2.3.1. Verbo OPEN

#### OBJETO

El verbo OPEN se utiliza para abrir ficheros, previamente a la obtención de datos del mismo (ficheros de entrada) o a la escritura de resultados sobre él (ficheros de salida). Abrir un fichero significa:

- Establecer comunicación entre la memoria central (área de entrada y/o salida) y el dispositivo externo que soporta el fichero.
- Iniciar la ejecución de comprobación de etiquetas para los ficheros de entrada y la creación de etiquetas para los ficheros de salida.
- Posicionar el fichero de modo que quede dispuesto para suministrar el primer registro de datos en caso de ficheros de entrada o para recibir el primer registro en caso de ficheros de salida.

#### FORMATO

El verbo OPEN tiene dos formatos dependiendo del acceso que se vaya a realizar a los ficheros.

*En ficheros con acceso secuencial*

```
OPEN [ INPUT nombre-fichero-1 [ REVERSED  
WITH NO REWIND ] ]  
[ nombre-fichero-2 [ REVERSED  
WITH NO REWIND ] ] ] .....  
[ OUTPUT nombre-fichero-3 [ WITH NO REWIND ] ]  
[ nombre-fichero-4 [ WITH NO REWIND ] ] ] .....
```

*En ficheros con acceso directo*

```
OPEN { INPUT-OUTPUT  
I-O } nombre-fichero-1  
[ nombre-fichero-2 ] ...
```

FORMATO 1.º

#### PARTICULARIDADES DE SU USO

Con el primer formato la instrucción OPEN permite abrir todos los ficheros de entrada que se incluyan dentro de la opción INPUT y todos los ficheros de salida incluidos en la opción OUTPUT.

El verbo OPEN debe ser el primero que se ejecute dentro del programa para poder hacer referencia a los datos en él contenidos (lo que no implica que deba ser la primera instrucción del programa).

Para un mismo fichero no pueden ejecutarse dos órdenes OPEN si no ha mediado entre ambas una orden CLOSE relativa a él.

**OPCIONES**

Debe figurarse obligatoriamente una de las opciones INPUT o OUTPUT y solamente una de cada tipo.

En otros términos, dentro de la misma orden OPEN es posible efectuar la apertura de dos ficheros, por ejemplo de entrada, mediante la opción INPUT, pero no es posible mediante dos opciones INPUT.

OPEN INPUT FICHERO-A FICHERO-B es correcto mientras OPEN INPUT FICHERO-A INPUT FICHERO-B sería incorrecto.

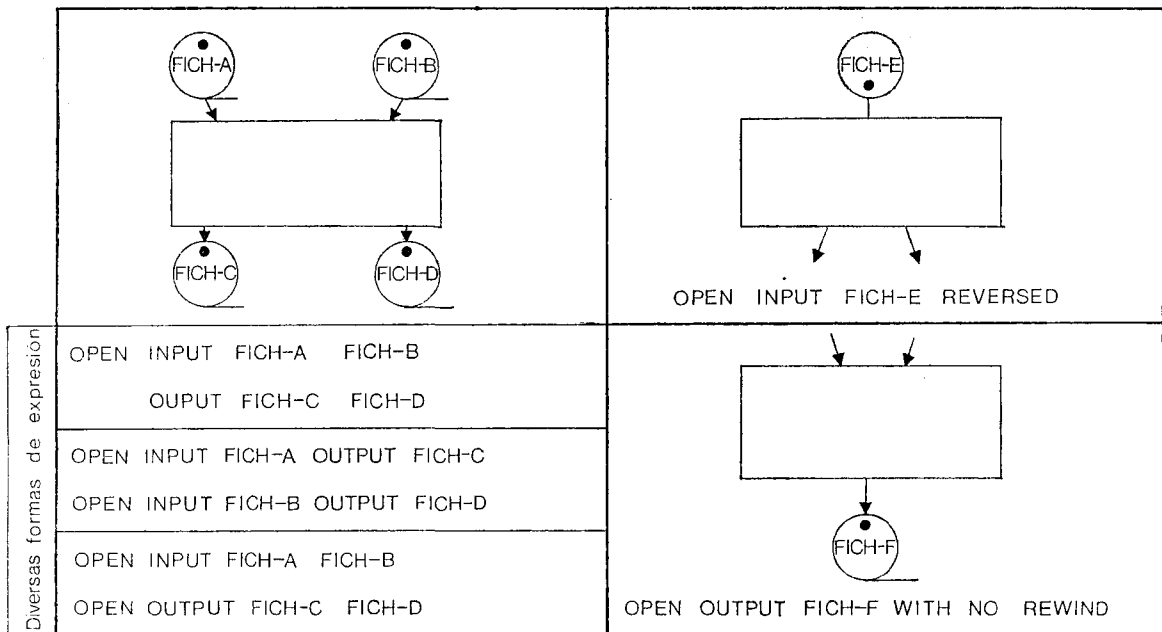
Iguales circunstancias concurren en la opción OUTPUT.

La opción REVERSED no puede aplicarse a ficheros con registros de tipo V y tiene solamente utilización en dispositivos que permitan la posibilidad de lectura de bandas magnéticas de sentido inverso. Esta opción supone que un fichero que se abre no se posiciona al comienzo del mismo, como ocurriría en el caso de no especificarse dicha opción, sino que obligatoriamente debe estar posicionado en su final.

La opción WITH NO REWIND permite abrir un fichero posicionado en su principio sin efectuarse la comprobación de etiquetas que la orden genera sin esta opción.

**EJEMPLOS**

En la figura VI-3 se muestran diversas formas correctas de efectuar la apertura de los ficheros indicados en los organigramas.



El • en la parte superior indica que el fichero debe estar posicionado en su principio, y en la parte inferior que debe estar posicionado en su final.

FIGURA VI-3

**FORMATO 2.º**

**PARTICULARIDADES DE SU USO**

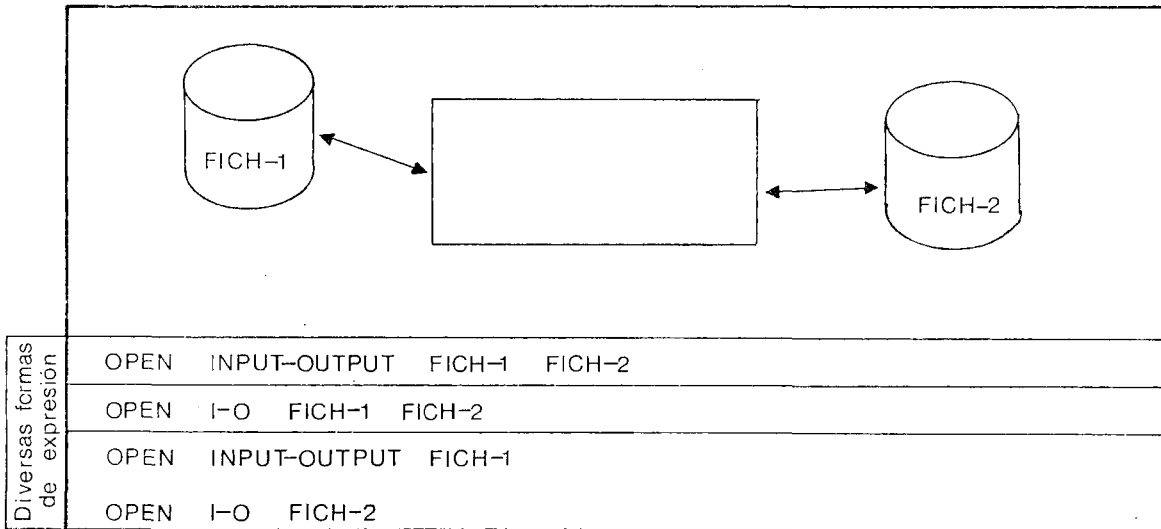
Con el segundo formato la instrucción OPEN permite abrir todos los ficheros que hayan sido definidos como de entrada-salida, es decir, aquellos que poseen en la cláusula SELECT de la ENVIRONMENT

DIVISION la opción ORGANIZATION IS DIRECT o la cláusula ACCES IS RANDOM.

**OPCIONES**

Las únicas posibles opciones de este formato son: la elección de las palabras reservadas INPUT-OUTPUT o I-O, que no producen acciones diferentes, y el número de ficheros a especificar que puede variar entre un mínimo de uno y un máximo únicamente limitado por las propias necesidades del proceso.

**EJEMPLO**



**VI-2.3.2. Verbo READ**

**OBJETO**

El verbo READ permite efectuar una operación de entrada o lectura, es decir, introducir en la memoria central (área de entrada) los registros que se encuentran sobre un fichero de entrada.

**EJEMPLO**

Existen dos posibles formatos.

*Para ficheros con organización seccional*

**READ** nombre-fichero RECORD [**INTO** nombre-dato]  
**AT END** declaración-imperativa

*Para ficheros con organización directa*

**READ** nombre-fichero RECORD [**INTO** nombre-dato]  
[**INVALID** KEY declaración-imperativa]

**PARTICULARIDADES DE SU USO**

Aunque la función primaria del verbo READ es la definida en el objeto de esta orden, este verbo efectúa otras diversas funciones:

- Verifica la longitud de cada registro de entrada para asegurarse que se corresponde con la especificada en la cláusula RECORD CONTAINS de la inscripción FD referida al mismo fichero. Si existe discrepancia emite un mensaje de error y se termina la ejecución del programa.

- Verifica el factor de bloqueo especificado por la cláusula **BLOCK CONTAINS** de la inscripción **FD** correspondiente al mismo fichero.
- Establece el flujo correcto de registros lógicos dentro del área de entrada. Este flujo tiene lugar de la forma siguiente:

La primera orden **READ** obtiene del fichero de entrada un bloque físico de registros que almacena en el área de entrada, posicionándose un indicador en el primer registro lógico del registro físico de forma tal que en la ejecución del programa se accede únicamente a la información contenida en él.

En las subsiguientes órdenes **READ** del programa y mientras haya registros lógicos del registro físico almacenado no se produce ninguna introducción de nuevos registros físicos y sí en cambio un avance en el posicionamiento del citado indicador hacia el siguiente registro lógico del registro físico almacenado. Esta operación se repite por cada orden **READ** hasta que se ha leído íntegramente, registro lógico a registro lógico, el registro físico almacenado. En la nueva orden **READ** se produce una nueva lectura real de un registro físico desde el fichero de entrada, iniciándose otra vez el posicionamiento del indicador en el primer registro lógico de nuevo registro físico leído. Esta operación se repite a lo largo del proceso hasta la terminación total del fichero en el caso de ficheros-secuenciales, interviniendo entonces la opción **AT END**.

- En el caso de tratarse ficheros multivolumenes, independientemente de que se haya especificado la opción **MULTIPLE REEL** de la cláusula **SELECT** en la **ENVIRONMENT DIVISION** el verbo **READ** ejecuta, cuando se llega al final de uno de los carretes que componen el fichero, las funciones de verificación de las etiquetas de fin de volumen del carrete leído, la apertura de un número carrete y la comprobación de etiquetas de éste, continuándose el proceso como si no hubiese habido discontinuidad física en las bandas magnéticas del fichero tratado.
- La única limitación en el uso de este orden viene dada por la necesidad de que exista una instrucción **OPEN**, ejecutada con anterioridad, referida al mismo fichero que se figura en la orden **READ**.
- Cuando se detecta la etiqueta fin de fichero, la orden **READ**, en lugar de ejecutar una nueva lectura, ejecuta la declaración imperativa que se codifique en el formato detrás de las palabras **AT END**.
- No puede ejecutarse una nueva lectura del fichero cuando se ha producido la condición de fin de fichero y se ha ejecutado la declaración imperativa correspondiente.
- La opción **INTO**, válida en ambos formatos, permite, al efectuarse la lectura de un registro, transferir el contenido del mismo a otra zona de la memoria central, además del área de entrada.

<b>OPCIONES</b>
-----------------

Esta zona viene especificada por nombre-dato, el cual debe haber sido definido en la **WORKING-STORAGE SECTION** o en la **FILE SECTION** como área de salida de otro fichero, que debe, por consiguiente, haber sido abierto con anterioridad a la ejecución de la orden **READ**. Los datos contenidos en el registro se mueven a dicha zona con la misma normativa que la especificada para el verbo **MOVE**.

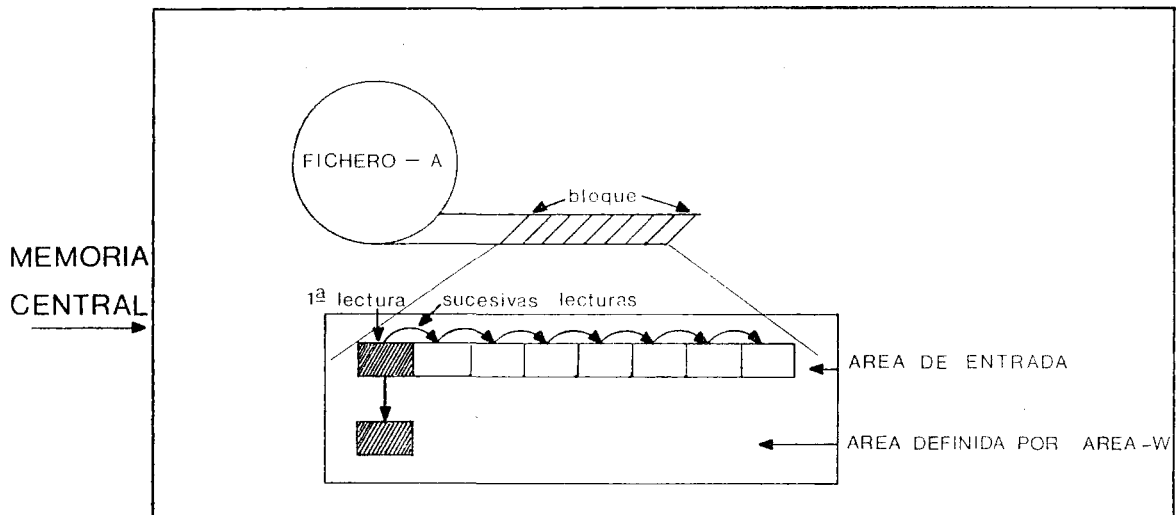
- La opción **INVALID KEY**, existente únicamente en el segundo formato, tiene por efecto ejecutar la declaración imperativa que le sigue cuando el contenido de la clave real y/o de la clave simbólica (ver cláusulas **KEY**) no son válidas siempre y cuando no se haya especificado para el fichero una sección declarativa de tratamiento de errores (ver capítulo VI-7).

Se considera que las claves no son válidas en los casos siguientes:

A) Cuando se especifican las cláusulas **ORGANIZATION IS DIRECT** y **ACCESS IS RANDOM** y el registro no se encuentra en los límites de la búsqueda o el número relativo de pista se encuentra fuera de los límites del fichero.

B) Cuando no se han establecido los valores de la clave simbólica y de la clave real del fichero antes de que se ejecute la operación de lectura.

**EJEMPLO**



**READ FICHERO-A RECORD INTO AREA-W AT END declaración-imperativa.**

Esta orden permite transferir del fichero contenido en el dispositivo correspondiente un bloque físico (compuesto, en el ejemplo, de 8 registros lógicos) y dejar accesible el primer registro lógico (el rayado si fuese la primera lectura). Además, por contener la opción **INTO AREA-W** transfiere el contenido de dicho registro lógico el área definida por dicho nombre de dato.

### VI-2.3.3. Verbo WRITE

#### OBJETO

El verbo WRITE permite efectuar una operación de salida o escritura, es decir, obtener del área de salida de la memoria central el resultado de un tratamiento y almacenarlo en un fichero de salida.

#### FORMATO

Existen dos posibles formatos:

*Para ficheros con organización secuencial.*

```
WRITE nombre-registro [ FROM nombre-dato-1 ]  
{ AFTER } ADVANCING { n.º-entero } LINES  
{ BEFORE }
```

*Para ficheros con organización directa.*

```
WRITE nombre-registro [ FROM nombre-dato-1 ]  
[ INVALID KEY declaración-imperativa ]
```

#### PARTICULARIDADES DE SU USO

Además de su función básica, definida en el objeto de este orden, el verbo WRITE efectúa otras diversas funciones:

- Establece el flujo correcto de registros lógicos desde el área de salida al fichero de salida.

Este flujo tiene lugar de la siguiente forma:

- Cada orden WRITE almacena en el área de salida un registro lógico, obtenido del proceso, en posiciones consecutivas de la misma hasta completar su capacidad. Esta capacidad viene definida por la cláusula BLOCK CONTAINS de la inscripción FD en la descripción del fichero. En esta situación la siguiente orden WRITE genera la escritura, grabación o perforación —en función del dispositivo utilizado— de un registro físico en el fichero de salida correspondiente.
- Cuando el fichero de salida es una cinta magnética y se trata de un fichero multivolumen, al completarse físicamente un carrete, la orden WRITE produce una orden de creación de un registro de etiquetas —si así ha sido especificado en la cláusula LABEL RECORD— y una orden de cierre y rebobinado del carrete completado. En este momento se produce o bien una interrupción del proceso hasta que el carrete se haya rebobinado totalmente y pueda ser liberado del dispositivo para ser sustituido por otro nuevo, o bien una alteración del dispositivo de salida a otro en el que se va a crear la continuación del fichero, de esta forma se ahorra durante el proceso, el tiempo de rebobinado. En cualquiera de estas circunstancias, el tercer paso es la generación de una etiqueta de comienzo de volumen en el nuevo carrete, continuación del anteriormente completado y grabación sobre él del registro físico creado.
- El verbo WRITE necesita, para el fichero especificado en la orden por nombre del registro, que se haya dado una orden previa de apertura del mismo.

**OPCIONES**

- Para ficheros con organización directa, es decir, cuando se ha especificado la cláusula ACCESS IS RANDOM; se deberá haber almacenado en los campos definidos por ACTUAL KEY y/o SYMBOLIC KEY los valores deseados antes de ejecutarse la declaración WRITE.

Las tres posibles opciones que pueden añadirse al formato básico del verbo WRITE son:

- Opción FROM.
- Opción  $\left\{ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right\}$
- Opción INVALID.

La opción FROM permite efectuar la escritura del registro desde el área definida por nombre-dato-1 (aunque en la realidad, y ya que únicamente puede efectuarse tal escritura desde el área de salida, esta opción produce una transferencia del registro desde el área definida por nombre-dato-1 hasta el área de salida).

La opción BEFORE o ADVANCING tiene uso exclusivamente para ficheros de salida impresos o para ficheros perforados en fichas.

Esta opción, aplicada a la impresión de resultados provoca un avance de un determinado número de líneas en el carro de la impresora antes o después —en función de la palabra utilizada— de imprimirse el registro de salida.

El número de líneas saltadas varía de un sistema a otro. Como ejemplo se muestra, en la figura VI-4, la codificación utilizada en los sistemas 360-370 de IBM.

		UTILIZANDO PARA		Produce en la impresora un:
		n.º-entero	contenido de nombre-dato-2	
VALORES		∅	1	SALTO A CANAL 1 (página siguiente)
		1	∅	Espaciado sencillo
		2	∅	Espaciado doble
		3	—	Espaciado triple
			+	Supresión de salto
			2 a 9	SALTO A CANALES 2 A 9, respectivamente
			A	SALTO A CANAL 10
			B	SALTO A CANAL 11
			C	SALTO A CANAL 12

FIGURA VI-4

Cuando la opción se aplica a salida perforada en fichas se debe emplear como identificación nombre-dato-2, y en este caso los posibles

valores de dicho campo representarán, en lugar de saltos del carro de la impresora, selección de casilleros receptores.

La opción INVALID KEY tiene aplicación únicamente en ficheros con organización directa, es decir, cuando se han especificado alguna de las cláusulas ORGANIZATION IS DIRECT y ACCESS IS RANDOM.

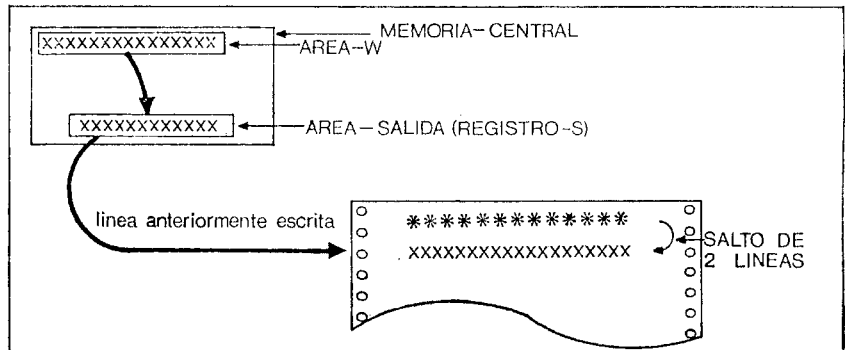
Esta opción provoca la ejecución de la declaración imperativa que le sigue cuando el contenido de la clave real y/o simbólica no es válida.

El contenido de la clave no es válida cuando ocurre alguna de las siguientes circunstancias:

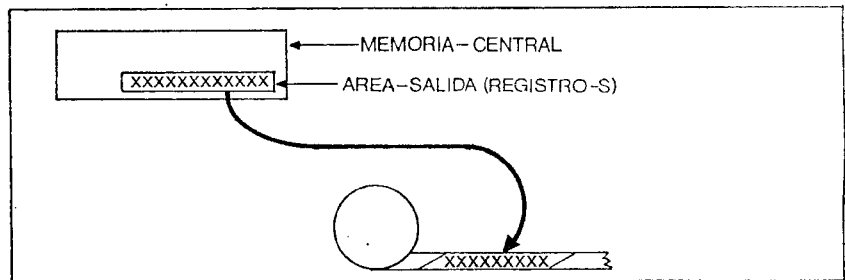
- La pista relativa especificada en el campo de ACTUAL KEY se encuentra fuera de los límites del fichero.
- Se ha movido la constante figurativa HIGH-VALUE a la primera posición de caracteres del contenido del campo SYMBOLIC KEY.
- Cuando se va a añadir un registro, en una organización indexada secuencial y el contenido de SYMBOLIC KEY es igual al de la clave de un registro ya existente en el fichero.

**EJEMPLOS**

La orden WRITE REGISTRO-S FROM AREA-W AFTER ADVANCING 2 LINES producirá la escritura sobre la impresora de una línea almacenada en el área descrita por REGISTRO-S (área de salida) después de haber sido transferida desde el área representada por AREA-W y después de haber saltado el carro de la impresora 2 líneas.



La orden WRITE REGISTRO-S produce la grabación de un registro en una banda magnética (si REGISTRO-S es el DATA RECORD de dicho fichero).





#### VI-2.3.4. Verbo CLOSE

**OBJETO**

El verbo CLOSE se utiliza para terminar el proceso de uno o varios ficheros.

**FORMATO**

Existe un único formato tanto para ficheros secuenciales como para fichero de azar.

```

CLOSE nombre-fichero-1 [ REEL UNIT ] WITH { NO REWIND LOCK }
[ nombre-fichero-2 [ REEL UNIT ] WITH { NO REWIND LOCK } ] .....
    
```

**PARTICULARIDADES DE SU USO**

- Las operaciones que tienen lugar al ejecutarse una orden CLOSE son diferentes según se trate de un fichero de entrada o de un fichero de salida.

FICHEROS DE ENTRADA	
ACCESO SECUENCIAL	ACCESO DIRECTO
<ul style="list-style-type: none"> <li>• Control de etiquetas si se ha detectado el fin del fichero.</li> <li>• No se controlan etiquetas si el cierre tiene lugar antes del fin del fichero, o se ha especificado la cláusula LABEL RECORD OMITTED.</li> <li>• Cambio de volumen, si se tratan ficheros multivolumen.</li> </ul>	Ninguna operación.

FICHEROS DE SALIDA	
ACCESO SECUENCIAL	ACCESO DIRECTO
<ul style="list-style-type: none"> <li>• Creación automática de etiquetas, si se ha especificado la cláusula LABEL RECORD STANDARD.</li> <li>• Cambio de volumen, si se está cuando un fichero multivolumen.</li> <li>• Creación de etiquetas en el nuevo carrete.</li> </ul>	Creación de etiquetas de fin de ficheros.

En el caso de tratarse de ficheros en banda magnética y de no emplearse ninguna de las opciones especificadas en el formato se produce un rebobinado del carrete soporte del fichero.

Después de ejecutarse una orden CLOSE no puede utilizarse ningún otro verbo de entrada-salida referido al mismo fichero, salvo el verbo OPEN, que en el caso de emplearse daría paso a la utilización de los restantes.

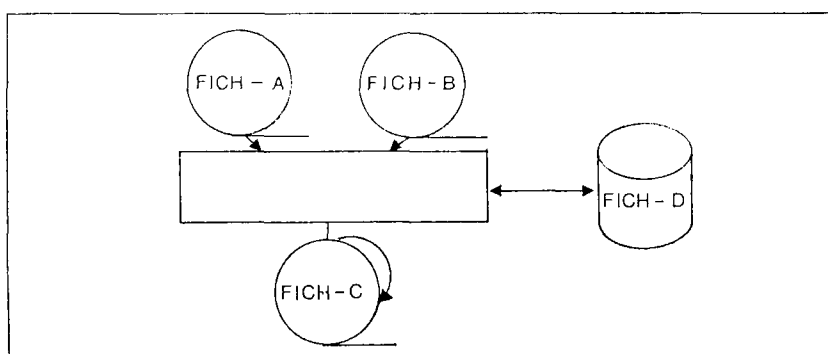
## OPCIONES

Las opciones REEL o UNIT tienen aplicación, respectivamente, en ficheros, en banda magnética y en disco magnético. Sirven para indicar que los ficheros tratados son multivolumen (varios carretes o varias unidades de disco).

La opción WITH NO REWIND, aplicable a bandas magnéticas, se utiliza para anular el rebobinado que se produce al cerrarse normalmente un fichero.

La opción WITH LOCK, aplicable a bandas y discos magnéticos, produce la liberación del fichero que se cierra. Esta liberación consiste en impedir un nuevo acceso al fichero, es decir, la imposibilidad de que se aplique una nueva orden OPEN al mismo fichero. Físicamente, se produce un desenhebrado de la cinta, en el caso de bandas magnéticas, y una marca de seguridad, en el caso de unidades en disco.

## EJEMPLOS



La orden CLOSE FICH-A FICH-B WITH NO REWIND FICH-C REEL FICH-D WITH LOCK producirá:

- Cierre y rebobinado del fichero denominado FICH-A.
- Cierre sin rebobinado del fichero denominado FICH-B.
- Cierre y rebobinado del carrete del fichero FICH-C.
- Cierre y protección contra nueva apertura del fichero FICH-D.
- Comprobación de etiquetas, si las hubiese, de los ficheros FICH-A, FICH-B y FICH-C.

### VI-2.4. OTROS VERBOS DE ENTRADA-SALIDA

Los verbos estudiados en los epígrafes anteriores tienen aplicación en todos los compiladores comentados en esta obra.

Los verbos que se estudian a continuación tienen aplicación solamente en determinados compiladores; por ello, en su estudio, nos limitamos a exponer únicamente sus características principales sin profundizar en todos los detalles particulares de su uso.

#### VI-2.4.1. Verbo **REWRITE**

Este verbo permite sustituir un registro lógico, situado en un dispositivo de acceso directo, por otro registro. El formato de una orden **REWRITE** es:

**REWRITE** nombre-de-registro [**FROM** nombre-dato] [**INVALID KEY** declaración-imperativa], en donde:

*nombre-de-registro* es el registro leído previamente por una orden **READ**.

*FROM nombre de dato* es una opción que tiene el mismo efecto que la opción **FROM** del verbo **WRITE**.

*INVALID KEY* es otra opción que produce la ejecución de la declaración-imperativa que le sigue cuando el contenido de la clave real y/o simbólica no es válido.

#### VI-2.4.2. Verbo **DELETE**

El verbo **DELETE** permite invalidar un registro anteriormente leído de un fichero que se ha abierto para actualizarlo.

El formato de este verbo es:

**DELETE** *nombre-registro*.

Para poder aplicar este verbo es necesario que exista un carácter de invalidación asociado a nombre-registro.

Cuando el verbo **DELETE** se aplica a ficheros secuenciales el borrado es físico (desaparición del registro del fichero); cuando el verbo **DELETE** se aplica a ficheros indexados la invalidación no se produce físicamente, sino únicamente con marca de borrado.

#### VI-2.4.3. Verbo **SEEK**

Este verbo *inicia* el acceso a un registro almacenado en un fichero organizado al azar. La ejecución del verbo **SEEK** precede a la del verbo **READ** o **WRITE** y la localización no produce el traslado al área de entrada o la grabación desde el área de salida, ya que éstas son efectuadas, necesariamente, por los verbos **READ** y **WRITE**, respectivamente. De hecho, cuando no se utiliza el verbo **SEEK** las funciones de este verbo son asumidas por los verbos **READ** o **WRITE**.

El formato de este verbo es:

**SEEK** *nombre-fichero* **RECORD**.



VI - 3

VERBOS DE MANIPULACION  
Y TRANSFERENCIA DE DATOS



### VI-3. VERBOS DE MANIPULACION Y TRANSFERENCIA DE DATOS

#### VI-3.1. GENERALIDADES SOBRE LOS VERBOS DE MANIPULACION DE DATOS

El fin de cualquier programa es el de obtener de los datos de un problema unos determinados resultados. Para ello son necesarias varias operaciones; admitirlos en la memoria para ser tratados, función que realizan los verbos READ o ACCEPT, ya estudiados, *tratarlos* para convertirlos en resultados y extraerlos de la memoria para conocerlos, función que realizan los verbos WRITE y DISPLAY. El tratamiento de los datos hasta convertirlos en resultados consiste, entre otras cosas, en transferirlos de una zona a otra de la memoria y transformarlos de valor. Esta transformación puede ser un cálculo más o menos complejo o una manipulación del contenido para alterar algunos de sus caracteres.

#### VI-3.2. VERBO MOVE O DE TRANSFERENCIA DE DATOS

<b>OBJETO</b>
---------------

Este verbo produce el traslado de un dato o grupo de datos de una zona a otra de la memoria central.

<b>FORMATO</b>
----------------

Existen dos posibles formatos:

*Para transferencia de datos considerados como campos elementales.*

<b>MOVE</b> { nombre-dato-1 } { <b>TO</b> nombre-dato-2 { literal-1 } [nombre-dato-3].....
--------------------------------------------------------------------------------------------------

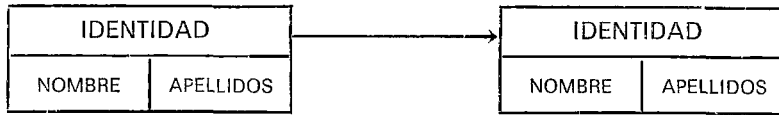
*Para transferencia de datos considerados como campos de grupos.*

<b>MOVE CORRESPONDING</b> nombre-dato-4 TO nombre-dato-5 [nombre-dato-6]...
--------------------------------------------------------------------------------

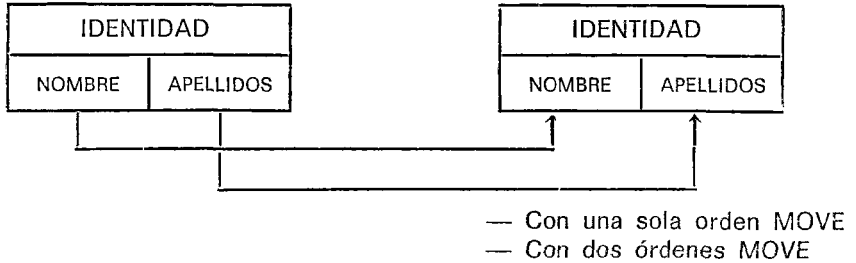
<b>PARTICULARIDADES DE SU USO</b>
---------------------------------------

Conviene hacer observar antes de proseguir en el estudio de este verbo lo que se considera, únicamente a efectos de transferencia, campo de grupo y campo elemental, ya que difiere de la definición ya realizada en el epígrafe V-1.2.1. En el esquema que figura en dicho epígrafe, el campo IDENTIDAD se define como campo de grupo por contener a los campos NOMBRE y APELLIDOS. Una transferencia de este campo se puede realizar de dos formas:

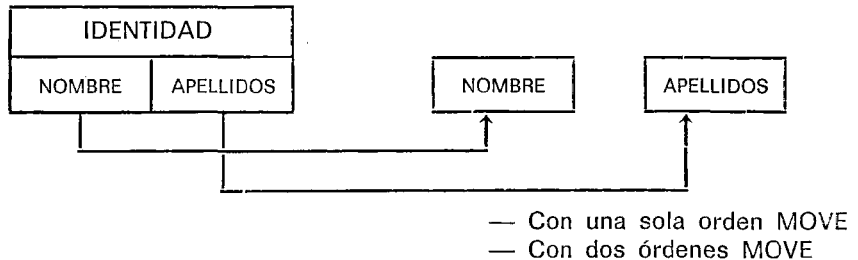
a) Como conjunto, es decir, como integrante de dos campos que forman un todo.



b) Moviendo sus dos componentes a posiciones consecutivas.



c) Moviendo sus dos componentes a posiciones distantes.

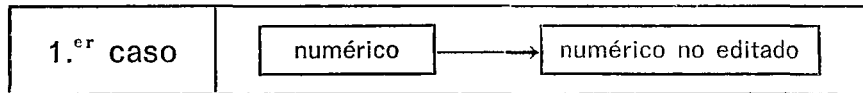


Para el primer tipo de movimiento *a)* se considera *IDENTIDAD*, para todos los efectos, como un campo elemental (cuya longitud es la suma de las longitudes de los campos elementales que lo forman) y se considera el campo receptor con contenido alfanumérico no editado. Para el segundo y tercer tipos de movimiento *b)* y *c)*, si se efectúa con una sola orden *MOVE*, se considera el campo como *campo de grupo*, y si se efectúa con dos órdenes *MOVE* no puede hacerse referencia a su nombre de grupo, debiendo hacerse, por consiguiente, a los campos elementales que lo componen, uno para cada movimiento.

En cualquier tipo de transferencia el contenido del campo emisor no se destruye ni transforma.

- Cuando se hace una transferencia de información entre dos campos elementales se pueden presentar cuatro casos diferentes:
  - 1.º) Campo emisor numérico y campo receptor numérico no editado.
  - 2.º) Campo emisor numérico y campo receptor numérico editado.
  - 3.º) Campo emisor de cualquier tipo y campo receptor alfanumérico o alfabético no editado.
  - 4.º) Campo emisor alfanumérico y campo receptor alfanumérico editado.





En esta transferencia, el emplazamiento del punto decimal supuesto del campo emisor se alinea con el punto decimal supuesto del elemento receptor. Se produce entonces un posible truncamiento de la parte entera y/o de la decimal o una adición de ceros no significativos en cualquiera de las partes, si el tamaño de los dos campos es diferente o si el emplazamiento relativo del punto decimal también difiere.

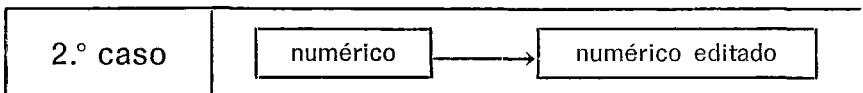
Si la cláusula USAGE de la descripción de ambos campos es diferente para cada uno de ellos, antes de efectuarse realmente la transferencia el contenido del campo emisor se transforma automáticamente al mismo tipo que la del campo receptor.

Si ambos campos tienen signo, el signo del campo emisor se transmite al emisor y le desplaza. Si el campo emisor no tiene signo y el campo receptor sí lo posee, se genera un signo + en el campo receptor. Si el campo emisor no tiene signo y el campo receptor lo posee se elimina el signo del campo receptor.

**EJEMPLO 1.º**

MOVE CAMPO-E TO CAMPO-R; para los siguientes supuestos:

PICTURE DEL CAMPO-E	Contenido del CAMPO-E	Valor numérico del CAMPO-E	PICTURE DEL CAMPO-R	Después de la transferencia desde el CAMPO-E al CAMPO-R	
				El contenido del CAMPO-R será	y su valor
999V9	3 2 1 8	321.8	999V9	3 2 1 8	321.8
999V9	3 2 1 8	321.8	99	2 1 Se produce truncamiento del 3 y del 8.	21
999V9	3 2 1 8	321.8	99V99	2 1 8 0 Se produce truncamiento del 3 y el relleno automático con un cero.	21.80
S999V99	1 2 3 4 5	-123.45	S999V99	1 2 3 4 5	-123.45
99V9	1 2 3	12.3	S99V9	1 2 3 +	12.3
S99V9	1 2 3	-12.3	99V9	1 2 3	12.3



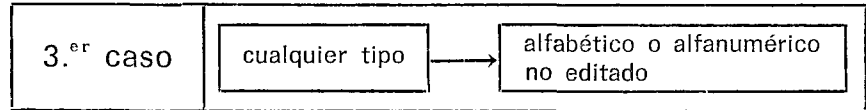
En esta transferencia pueden ocurrir dos circunstancias:

- a) Que el campo emisor no esté representado en caracteres.
- b) Que el campo emisor esté representado en caracteres.

En el caso a) se efectúa una transformación automática en caracteres y se pasa al caso b).

En este caso, pues, la transformación tiene lugar con las mismas reglas y ejemplos que hemos visto en el epígrafe V-3.8.2.6 para la PICTURE de edición de los campos numéricos editados.

En este caso el campo receptor no puede ser utilizado para efectuar cálculos aritméticos y debe ser considerado como un campo alfanumérico.



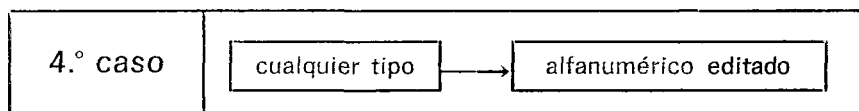
En este tercer caso el campo emisor se considera como alfanumérico a efectos de transferencia y por consiguiente no se puede producir ninguna transformación, salvo en el truncamiento de caracteres cuando el campo emisor sea de mayor longitud que el receptor o la adición de espacios en el campo receptor cuando éste sea de mayor longitud que el emisor.

En este tipo de transferencia tiene aplicación la cláusula JUSTIFIED para el campo receptor con sus correspondientes implicaciones (ver epígrafe V-3.8.2.2).

**EJEMPLO 2.º**

MOVE CAMPO-E TO CAMPO-R; para los siguientes supuestos:

PICTURE DEL CAMPO-E	Contenido del CAMPO-E	Valor del CAMPO-E	PICTURE DEL CAMPO-R	Después de la transferencia desde el CAMPO-E al CAMPO-R	
				el contenido del CAMPO-R será	y su valor
A(5)	A   B   C   D   E	'ABCDE'	X(6)	A   B   C   D   E   $\emptyset$ Se adiciona un espacio	'ABCDE $\emptyset$ '
X(5)	A   B   C   D   E	'ABCDE'	X(4)	A   B   C   D Se trunca la E	'ABCD'
A(5)	A   B   C   D   E	'ABCDE'	A(6) JUSTIFIED RIGHT	$\emptyset$   A   B   C   D   E Se adiciona un espacio	' $\emptyset$ ABCDE'
A(5)	A   B   C   D   E	'ABCDE'	X(4) JUSTIFIED RIGHT	B   C   D   E Se trunca la A	'BCDE'
9(3)	1   2   3	123	X(3)	1   2   3	'123'
99V99	1   2   3   4	12.34	X(6)	1   2   3   4   $\emptyset$   $\emptyset$	'1234 $\emptyset\emptyset$ '
9(2)	1   2	12	X(3) JUSTIFIED RIGHT	$\emptyset$   1   2	' $\emptyset$ 12'



En este tipo de transferencia se siguen las reglas de edición consideradas para campos alfanuméricos editados (ver epígrafe V-3.8.2.6).

**EJEMPLO 3.º**

MOVE CAMPO-E TO CAMPO-R; para los siguientes supuestos:

PICTURE DEL CAMPO-E	Contenido del CAMPO-E	Valor del CAMPO-E	PICTURE DEL CAMPO-R	Después de la transferencia desde el CAMPO-E al CAMPO-R	
				el contenido del CAMPO-R será	y su valor
A(5)	A B C D E	'ABCDE'	XXBXXØX	A B Ø C D Ø E	'AØBØCDØE'
X(5)	A B C D E	'ABCDE'	XØØØXXXX	A Ø Ø Ø B C D E	'AØØØBCDE'
X(2)	2 5	'25'	XØØX	2 Ø Ø 5	'2ØØ5'
9V9	1 2	1.2	XØX	1 Ø 2	'1Ø2'
9V99	1 2 3	1.23	XBXX	1 Ø 2 3	'1Ø23'

Una transferencia entre campos considerados de grupo es equivalente a un determinado número de transferencias entre campos considerados elementales.

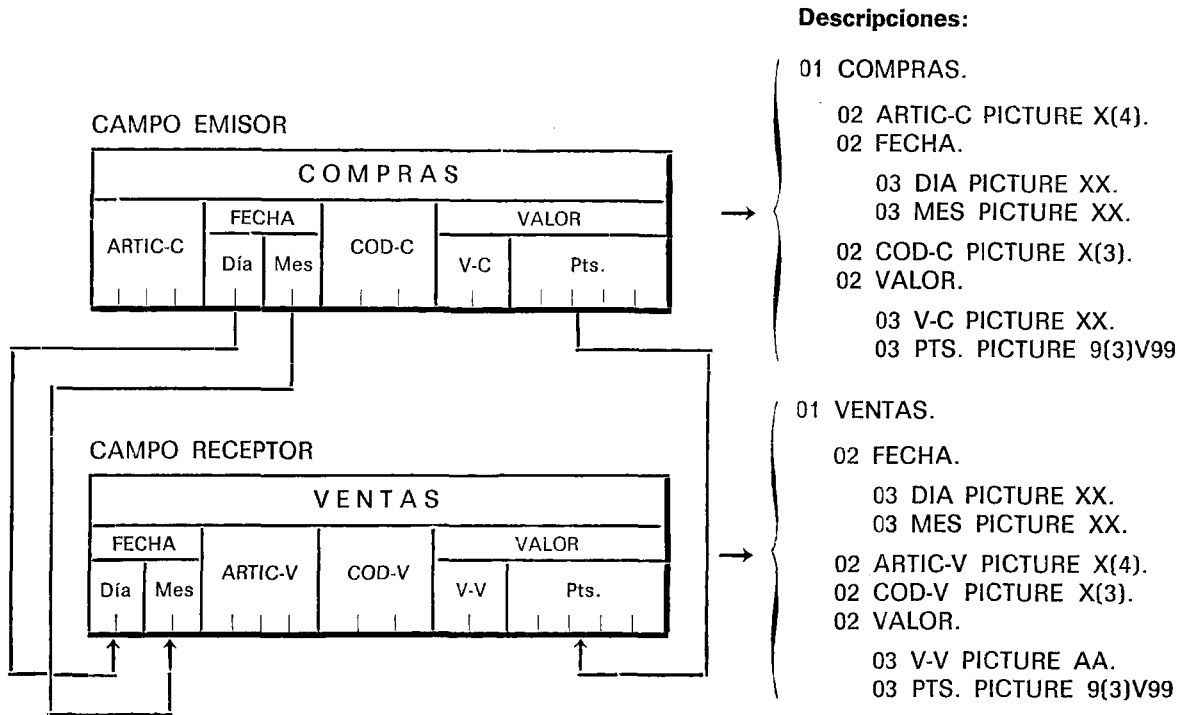
Para poder efectuar una transferencia entre campos de grupo es necesario que se cumplan simultáneamente las condiciones siguientes:

- a) Que alguno de los campos que forman parte del campo de grupo emisor tenga el mismo nombre de dato que alguno de los campos del campo de grupo receptor (por consiguiente, deben estar calificados).
- b) Que ocupen los campos con igual nombre de dato la misma posición jerárquica en ambos campos de grupo.
- c) Que, en el caso de existir niveles de grupo intermedios entre los campos elementales y el campo de grupo, los nombres de dato de estos niveles de grupo intermedios sean también los mismos.

Solamente con esta triple condición se puede aplicar, o surte efecto la orden MOVE CORRESPONDING y su acción es la de producir tantas órdenes MOVE del primer formato, ya estudiado, como campos existen en ambos grupos que posean el mismo nombre de dato.

**EJEMPLO 4.º**

Veamos gráficamente lo expuesto anteriormente:



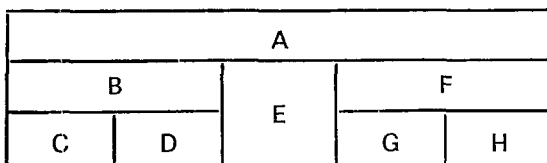
La instrucción **MOVE CORRESPONDING COMPRAS TO VENTAS** produce la transferencia de los contenidos de los campos:

DIA DE COMPRAS A DIA DE VENTAS  
 MES DE COMPRAS A MES DE VENTAS  
 PTS. DE COMPRAS A PTS. DE VENTAS

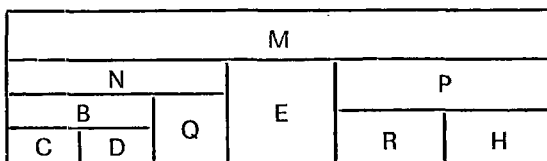
por tener en ambos el mismo nombre, la misma posición jerárquica y los mismos nombre de dato en los niveles de grupo intermedios (FECHA y VALOR).

**EJEMPLO 5.º**

CAMPO EMISOR



CAMPO RECEPTOR



MOVE CORRESPONDING A TO M

Produce únicamente la transferencia del campo E del campo emisor A al campo receptor M, ya que aunque B, C y D tienen el mismo nombre en A y H, no se produce transferencia por no tener igual jerarquía, ni tampoco el campo H, por pertenecer a un grupo intermedio de distinto nombre, en ambos campos.

### VI.3.3. VERBOS DE MANIPULACION DE DATOS

#### VI-3.3.1. Verbo EXAMINE o de manipulación simple

**OBJETO**

El verbo EXAMINE permite examinar el contenido en caracteres de un campo para determinar la existencia de un determinado carácter (variable) y/o transformarlo por otro diferente.

**FORMATO**

Existen dos posibles formatos:

Primer formato: Cómputo de determinados caracteres y/o transformación.

<p><u>EXAMINE</u> nombre-de-dato <u>TALLYING</u> { <u>ALL</u> <u>LEADING</u> <u>UNTIL FIRST</u> } 'carácter-1' [REPLACING BY 'carácter-2']</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------

Segundo formato: Únicamente transformación.

<p><u>EXAMINE</u> nombre-de-dato-2 <u>REPLACING</u> { <u>ALL</u> <u>LEADING</u> <u>FIRST</u> <u>UNTIL FIRST</u> } 'carácter-3' BY 'carácter-4'</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------

**PARTICULARIDADES DE SU USO**

Este verbo tiene únicamente aplicación cuando el campo examinado, representado por nombre-de-dato, tiene asignada una cláusula USAGE DISPLAY, es decir, está representado en caracteres.

Por otra parte, tanto los caracteres a localizar y/o transformar como los caracteres que se utilicen en la transformación, 'carácter-x', deben ser de la misma clase que nombre-de-dato.

'Carácter-x' puede ser también cualquier constante figurativa (excepto ALL).

**OPCIONES**

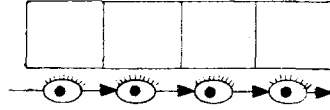
Las dos posibles opciones son TALLYING y REPLACING o ambas simultáneamente.

La elección de TALLYING produce la activación de un contador especial, representado por un registro denominado TALLY, al cual puede acceder el programador en cualquier momento del programa. En este contador se almacena el valor del cómputo de caracteres localizados en el examen del campo.

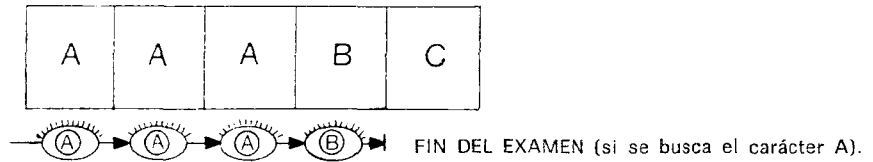
La elección de REPLACING produce la sustitución de los caracteres localizados en el examen del campo por el carácter indicado detrás de BY.

En las dos opciones el examen de caracteres contenidos en el campo se produce de acuerdo con la palabra reservada elegida de la opción.

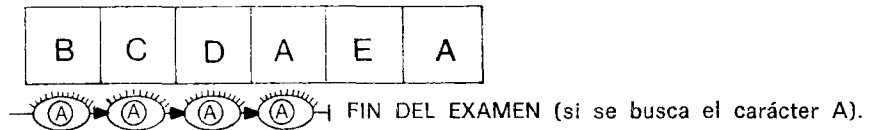
**ALL:** Origina el examen de la totalidad de caracteres del campo.



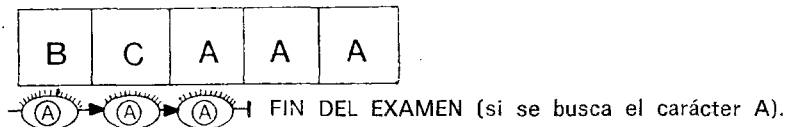
**LEADING:** Origina el examen de los caracteres del campo, de izquierda a derecha del mismo, hasta que detecta un carácter distinto del buscado. En dicho carácter se termina la investigación.



**FIRST:** Solamente localiza el primer carácter (de izquierda a derecha) que aparece en el campo. Únicamente tiene efecto con la opción REPLACING.



**UNTIL FIRST:** Es la contraria de LEADING. Examina de izquierda a derecha todos los caracteres del campo hasta que aparece por primera vez el carácter buscado. En este carácter se termina la investigación.



**EJEMPLOS**

Orden EXAMINE	Contenido de CAMPO-W		Valor del contador TALLY
	Antes	Después	
EXAMINE CAMPO-W TALLYING ALL 'Ø'	1   Ø   Ø   1   Ø   Ø ↑    ↑    ↑    ↑	1   Ø   Ø   1   Ø   Ø	4
EXAMINE CAMPO-W TALLYING LEADING '1'	1   1   1   Ø   1   Ø ↑    ↑    ↑	1   1   1   Ø   1   Ø	3
EXAMINE CAMPO-W TALLYING UNTIL FIRST '1'	Ø   1   Ø   1   Ø   1 ↑	Ø   1   Ø   1   Ø   1	1
EXAMINE CAMPO-W TALLYING ALL ZERO REPLACING BY '2'	Ø   1   Ø   1   Ø   1 ↑    ↑    ↑	2   1   2   1   2   1	3
EXAMINE CAMPO-W TALLYING LEADING SPACES REPLACING BY ZEROS	Ø   Ø   Ø   2   3   4             ↑	0   0   0   2   3   4	3
EXAMINE CAMPO-W TALLYING UNTIL FIRST '5' REPLACING BY '3'	A   7   4   6   5   1                 ↑	3   3   3   3   5   1	4
EXAMINE CAMPO-W REPLACING ALL 'A' BY '1'	A   Ø   5   A   B   7 ↑           ↑	1   Ø   5   1   B   7	no se activa
EXAMINE CAMPO-W REPLACING LEADING '3' BY 'A'	3   3   1   2   Ø   7                 ↑	A   A   1   2   Ø   7	no se activa
EXAMINE CAMPO-W REPLACING FIRST '2' BY '*'	1   2   3   4   2   6	1   *   3   4   2   6	no se activa
EXAMINE CAMPO-W REPLACING UNTIL FIRST 'A' BY 'B'	D   C   A   B   E   F	B   B   A   B   E   F	no se activa

**VI-3.3.2. Verbo TRANSFORM o de manipulación múltiple**

**OBJETO**

El verbo TRANSFORM se utiliza para alterar parte o la totalidad de los caracteres de un campo de acuerdo con una regla de transformación.

**FORMATO**

Existe un único formato para este verbo:

**TRANSFORM** nombre-dato CHARACTERS **FROM** { opción-1 } **TO** { opción-2 }

**PARTICULARIDADES DE SU USO**

El campo que se examine, para ser alterado, representado por nombre-dato, de ser un campo alfabético elemental, alfanumérico o un campo de grupo.

El uso de este verbo es, normalmente, equivalente al empleo de un determinado número de órdenes EXAMINE aplicadas al mismo campo (este número de órdenes dependerá de la regla de transformación utilizada).

La regla de transformación está en función de las combinaciones elegidas, entre las posibles, de opción-1 y opción-2.

**OPCIONES**

Tanto en opción-1 como en opción-2 se pueden elegir uno de los siguientes elementos:

CONSTANTE FIGURATIVA

LITERAL-NO-NUMERICO

NOMBRE-DATO

Existen, por consiguiente, nueve posibles combinaciones que determinan los nueve tipos de reglas de transformación existentes en esta orden.

Las reglas de transformación son, pues:

OPCION-1	OPCION-2	Regla de transformación
CONSTANTE-FIGURATIVA-1	CONSTANTE-FIGURATIVA-2	<b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> al valor representado por CONSTANTE-FIGURATIVA-1, son sustituidos por el valor representado por CONSTANTE-FIGURATIVA-2.
CONSTANTE-FIGURATIVA-1	LITERAL-NO-NUMERICO-2	<b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> al valor representado por CONSTANTE-FIGURATIVA-1, son sustituidos por el valor de LITERAL-NO-NUMERICO-2 (que en esta regla debe tener un solo carácter).



OPCION-1	OPCION-2	Regla de transformación
CONSTANTE-FIGURATIVA-1	NOMBRE-DATO-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> al valor representado por CONSTANTE-FIGURATIVA, son sustituidos por el contenido del NOMBRE-DATO-2 en el momento de la ejecución (En esta regla, la PICTURE de NOMBRE-DATO debe ser PICTURE-X, es decir, con un solo carácter).</p>
LITERAL-NO-NUMERICO-1	CONSTANTE-FIGURATIVA-1	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales a cualquier carácter</b> de los representados por LITERAL-NO-NUMERICO-1, son sustituidos por el del valor representado por CONSTANTE-FIGURATIVA-2.</p>
LITERAL-NO-NUMERICO-1	LITERAL-NO-NUMERICO-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales a cualquier carácter</b> de los representados por LITERAL-NO-NUMERICO-1, son sustituidos por:</p> <ul style="list-style-type: none"> <li>a) al <b>correspondiente</b> carácter (que ocupe la misma posición relativa) de LITERAL-NO-NUMERICO-2, cuando la longitud de LITERAL-NO-NUMERICO-1 sea igual a la longitud de LITERAL-NO-NUMERICO-2.</li> <li>b) el <b>carácter</b> representado por LITERAL-NO-NUMERICO-2 cuando la longitud de éste es de un carácter.</li> </ul> <p>En esta regla, la longitud de ambos literales no puede ser diferente, salvo que la de LITERAL-NO-NUMERICO-2 sea de 1 carácter.</p>
LITERAL-NO-NUMERICO-2	NOMBRE-DATO-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> a cualquier carácter de los representados por LITERAL-NO-NUMERICO-1 son sustituidos por:</p> <ul style="list-style-type: none"> <li>a) el correspondiente carácter (que ocupe la misma posición relativa) del contenido de NOMBRE-DATO-2, cuando la longitud de LITERAL-NO-NUMERICO-1, es igual al representado por la PICTURE de NOMBRE-DATO-2.</li> </ul>

OPCION-1	OPCION-2	Regla de transformación
		<p>b) el contenido de NOMBRE-DATO-2 cuando la PICTURE de NOMBRE-DATO-2 es PICTURE X.</p> <p>En esta regla, la longitud de LITERAL-NO-NUMERICO-1 debe ser igual a la representada por la PICTURE de NOMBRE-DATO-2, salvo cuando ésta es PICTURE X.</p>
NOMBRE-DATO-1	CONSTANTE-FIGURATIVA-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> a cualquier carácter de los figurados en el contenido de NOMBRE-DATO-1 son sustituidos por el valor representado por CONSTANTE-FIGURATIVA-2.</p>
NOMBRE-DATO-1	LITERAL-NO-NUMERICO-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> a cualquier carácter de los figurados en el contenido de NOMBRE-DATO-1 son sustituidos por:</p> <p>a) el correspondiente carácter (que ocupa la misma posición relativa) del representado por LITERAL-NO-NUMERICO-2, cuando la longitud representada por la PICTURE de NOMBRE-DATO-1 es igual a la longitud de LITERAL-NO-NUMERICO-2.</p> <p>b) el valor de LITERAL-NO-NUMERICO-2 cuando éste tiene una longitud de 1 carácter.</p> <p>En esta regla, la longitud representada por la PICTURE de NOMBRE-DATO-1, debe ser igual a la longitud de LITERAL-NO-NUMERICO-2, salvo cuando la de éste es de 1 carácter.</p>
NOMBRE-DATO-1	NOMBRE-DATO-2	<p><b>Todos</b> los caracteres del campo examinado <b>que sean iguales</b> a cualquier carácter de los figurados en el contenido de NOMBRE-DATO-1 son sustituidos por:</p> <p>a) el correspondiente carácter (que ocupe la misma posición relativa) del contenido de NOMBRE-DATO cuando la longitud representada por las PICTURE de NOMBRE-DATO-1 y de NOMBRE-DATO-2 sea la misma.</p> <p>b) el contenido de NOMBRE-DATO-2 cuando éste tiene PICTURE-X.</p> <p>En esta regla, la longitud representada por las PICTURE de NOMBRE-DATO-1 y NOMBRE-DATO-2 debe ser la misma, salvo que NOMBRE-DATO-2 tenga PICTURE-X.</p>





VI - 4

VERBOS ARITMETICOS



## VI-4. VERBOS ARITMETICOS

### VI-4.1. GENERALIDADES

La misión de cualquier programa es, como ya se ha indicado, transformar los datos de un problema en resultados. ¿Cómo se realiza esta transformación? Algunas veces, un dato puede ser también resultado, y, por consiguiente, no sufre ninguna transformación en el programa, salvo una posible edición; en estos casos no interviene la unidad aritmético-lógica y el dato pasa directamente de la zona de entrada o lectura a la zona de salida o escritura. En la mayoría de los casos los datos sufren transformaciones, elaboraciones, operaciones, etc., hasta quedar convertidos en resultados. Estas, en su sentido más amplio, suponen que los datos cambien de posición dentro de la memoria, se sumen, resten, multipliquen o dividan por otros datos, etc. Todas estas transformaciones se realizan, bien con los verbos de manipulación y transferencia de datos, estudiados en el apartado VI-3, bien con los verbos aritméticos que se estudian en este apartado.

Debido a que en estos verbos, tanto las opciones como las particularidades de su uso, son totalmente análogas para cada uno de los cinco verbos que componen el conjunto de verbos aritméticos se va a efectuar la exposición de los mismos, con una sistemática ligeramente diferente a la utilizada en apartados anteriores.

En primer lugar, y en vez de realizarse el estudio particular de cada verbo, se exponen para el conjunto las particularidades que entrañan el uso de los verbos aritméticos. En segundo término, estudiamos las opciones de los distintos formatos, que son de común aplicación a todos y cada uno de los verbos. En último lugar se expone el objeto y ejemplos de cada uno de los verbos en particular.

### VI-4.2. PARTICULARIDADES DEL USO DE LOS VERBOS ARITMETICOS

Los cálculos aritméticos se deben realizar entre operandos que deben ser literales-numéricos, constantes con nombre, constantes figurativas o nombres de datos, previamente descritos en la DATA DIVISION como elementos numéricos no editados.

Los resultados de tales cálculos son colocados en un nombre de dato que puede ser bien uno de los diversos operandos puestos en juego, bien

uno o varios nombres de datos nuevos, reservados a este efecto y que también deben haber sido previamente definidos en la DATA DIVISION.

La única constante figurativa que puede figurar como operando de un verbo aritmético es ZERO. Si se emplea esta constante, no puede ser como operando utilizado también para albergar el resultado.

La longitud máxima admitida, en el caso de utilizar literales numéricos está limitada, normalmente a 18, en la mayoría de los compiladores.

Es necesario que la descripción del operando que debe recibir el resultado, esté en concordancia con las descripciones de los operandos puestos en juego.

Esta concordancia se refiere al tamaño de la cláusula PICTURE y no a su estructura, ya que ésta puede diferir de un operando a otro.

Por ejemplo, si efectuamos la suma de varios operandos con PICTURE diferentes

```
operando-1 PICTURE 999V99
operando-2 PICTURE 99V999
operando-3 PICTURE 9999V9
```

la PICTURE del operando que albergue el resultado deberá ser, como mínimo, PICTURE 99999V999, si no se desea perder ninguna cifra significativa.

Durante la compilación, el programa traductor tiene en cuenta los operandos y naturaleza de los operadores aritméticos y examina la descripción del operando utilizado para albergar el resultado y emite un diagnóstico de aviso si no se respeta la concordancia apuntada en lo que se refiere a la parte entera del resultado.

Durante la ejecución se pueden presentar los siguientes casos:

- a)  $T_e > T_n$
- b)  $T_e < T_n$
- c)  $T_d > T_n$
- d)  $T_d < T_n$

indicándose, abreviadamente por T tamaño  
e parte entera del resultado  
d parte decimal del resultado  
n necesario

Las acciones que se toman en cada caso son:

- a) Se rellena el espacio con ceros no significativos.
- b) Se produce un truncamiento de las posiciones más significativas (dígitos de mayor valor relativo).
- c) Se rellena el espacio sobrante con ceros no significativos.
- d) Se produce un truncamiento de las posiciones menos significativas (dígitos decimales de menor valor relativo).



El programador puede conocer y controlar las acciones que ocurren en los casos *b)* y *d)*, que suponen una modificación del valor del resultado real, utilizando las opciones que se pueden incluir en los formatos de los verbos.

#### VI-4.3. OPCIONES COMUNES A TODOS LOS VERBOS ARITMETICOS

Existen tres posibles opciones que son de aplicación simultánea en cada uno de los formatos.

opción 1: **GIVING** (nombre-dato) [opción-2] [nombre-dato-2 [opción-2] ]...

opción 2: **ROUNDED**

opción 3: **ON SIZE ERROR** declaración-imperativa

##### VI-4.3.1. Opción **GIVING**

La opción **GIVING** se utiliza cuando se desea que el resultado de la operación aritmética se sitúe en el campo o campos representados por los nombres de datos que aparecen detrás de esta palabra. Cuando se ejecuta la operación aritmética el resultado obtenido reemplaza los posibles valores que pudieran tener los nombres de datos de la opción antes de efectuarse la operación.

Esta opción permite, como caso excepcional, que la descripción de los nombres de datos que van a contener los resultados contengan caracteres de edición. Si esto ocurre no se podrán utilizar posteriormente, como es lógico, dichos campos en otras operaciones aritméticas.

El hecho de que existan varios nombres de datos dentro de la opción **GIVING** permite tener almacenados en distintas partes de la memoria el resultado de una misma operación. Normalmente esto se efectúa cuando se desea tener distintas versiones de un mismo resultado, en cuanto a grados de significación se refiere: con un decimal, con dos decimales, sin decimales, redondeando a una determinada cifra decimal, editado, sin editar, etc.

##### VI-4.3.2. Opción **ROUNDED**

Cuando esta opción es especificada en una orden de cálculo aritmético, la cual debe seguir al nombre de dato que representa al campo que deberá albergar el resultado, la operación se efectúa de la siguiente forma:

Si en la ejecución de la orden se presenta el caso *d)* descrito en el epígrafe VI-4.2., se efectúa el truncamiento allí mencionado, pero antes se redondea la última cifra decimal almacenada. Este redondeo consiste en añadir una unidad en dicha cifra si la primera cifra decimal truncada es superior o igual a 5.

De esta forma se obtiene una precisión mayor en el resultado.

### VI-4.3.3. Opción ON SIZE ERROR

Cuando esta opción se especifica en una orden de cálculo aritmético, la operación se efectúa de la siguiente forma:

Si durante la ejecución de la orden se presenta el caso *b)* descrito en el epígrafe VI-4.2., no se efectúa el truncamiento allí mencionado, y sí, en su lugar, la ejecución de la declaración imperativa que acompaña a la opción. De esta forma se controla cualquier anomalía que pueda ocurrir en el desarrollo de un cálculo. Un ejemplo muy claro se presenta en el caso de utilizar el verbo DIVIDE y contener el divisor, en el momento de la ejecución de la orden, el valor  $\emptyset$  ( $X/\emptyset$ , es infinito y, por consiguiente, imposible de almacenar).

### VI-4.4. ESTUDIO PARTICULAR DE CADA UNO DE LOS VERBOS ARITMETICOS

La figura VI-5, situada al final de este capítulo, resume todos los formatos posibles de los verbos aritméticos, así como los que tienen aplicación en cada uno de los compiladores mencionados.

#### VI-4.4.1. Verbo ADD

**OBJETO**

Este verbo permite sumar el contenido de dos o más campos elementales y/o literales y almacenar el resultado de la suma en uno o varios campos elementales.

**FORMATOS**

Existen cuatro formatos posibles, que son los indicados en la figura VI-5 (formato 1-1 a 1-4).

**PARTICULARIDADES DE SU USO**

*Con el primer formato*, dos o más literales o campos, representados por operando-1, operando-2, ... se suman a un único campo representado por operando-n. En este campo se almacena también el resultado de la suma.

*Con el segundo formato, utilizando la opción-1*, uno o más campos representados por operando-1; operando-2, ... se suman entre sí, y el resultado se almacena en cada uno de los campos representados por los distintos operandos de opción-1.

*Con el segundo formato, y utilizándose la opción TO*, uno o más campos representados por operando-1, operando-2, ... se suman entre sí, y el resultado parcial se suma a cada uno de los operandos: operando-m, operando-n, ...

Los distintos resultados finales se almacenan, respectivamente, en operando-m; operando-n, ...

*Con el tercer formato*, uno o más literales o campos elementales representados por operando-1, operando-2, ... se suman al literal o campo representado por operando-p, y el resultado se almacena en cada uno de los campos representados por los distintos operandos de opción-1.

Con el cuarto formato, cada uno de los campos elementales del campo de grupo, representado por operando-1, se suma a cada uno de los campos elementales de los campos de grupo representados por operando-2, ... siempre y cuando los campos elementales de dichos operandos cumplan las mismas condiciones establecidas en la opción CORRESPONDING del verbo MOVE.

**EJEMPLOS**

Suponiendo que el contenido y PICTURE de los campos: CAMPO-1, CAMPO-2, CAMPO-3, CAMPO-4, CAMPO-5 y CAMPO-6 sean los siguientes:

CAMPO-1	PICTURE	99V99	Contenido	<table border="1"><tr><td>3</td><td>2</td><td>1</td><td>Ø</td></tr></table>	3	2	1	Ø	
3	2	1	Ø						
CAMPO-2	PICTURE	999V99	»	<table border="1"><tr><td>2</td><td>5</td><td>6</td><td>1</td><td>7</td></tr></table>	2	5	6	1	7
2	5	6	1	7					
CAMPO-3	PICTURE	99V9	»	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr></table>	1	2	3		
1	2	3							
CAMPO-4	PICTURE	999V99	»	<table border="1"><tr><td>3</td><td>2</td><td>1</td><td>Ø</td><td>Ø</td></tr></table>	3	2	1	Ø	Ø
3	2	1	Ø	Ø					
CAMPO-5	PICTURE	99V99	»	<table border="1"><tr><td>2</td><td>5</td><td>4</td><td>7</td></tr></table>	2	5	4	7	
2	5	4	7						
CAMPO-6	PICTURE	999V9	»	<table border="1"><tr><td>2</td><td>3</td><td>Ø</td><td>3</td></tr></table>	2	3	Ø	3	
2	3	Ø	3						

Los resultados de aplicar las siguientes órdenes de suma serán:

	RESULTADOS	Que se almacenan en									
a) ADD CAMPO-1 CAMPO-2 CAMPO-6 ROUNDED 32.10 + 256.17 + 230.4 = 518.67 Redondeado a un decimal = 518.7	<table border="1"><tr><td>5</td><td>1</td><td>8</td><td>7</td></tr></table> (coma decimal supuesta)	5	1	8	7	CAMPO-6					
5	1	8	7								
b) ADD CAMPO-1 CAMPO-2 TO CAMPO-4 32.10 + 256.17 + 321.00 = 609.27	<table border="1"><tr><td>6</td><td>Ø</td><td>9</td><td>2</td><td>7</td></tr></table>	6	Ø	9	2	7	CAMPO-4				
6	Ø	9	2	7							
c) ADD CAMPO-1 CAMPO-2 GIVING CAMPO-6 ROUNDED CAMPO-4 32.10 + 256.17 = 288.27 Redondeado a un decimal = 288.3	<table border="1"><tr><td>2</td><td>8</td><td>8</td><td>3</td></tr></table> <table border="1"><tr><td>2</td><td>8</td><td>8</td><td>2</td><td>7</td></tr></table>	2	8	8	3	2	8	8	2	7	CAMPO-6 CAMPO-4
2	8	8	3								
2	8	8	2	7							
d) ADD CAMPO-1 TO CAMPO-3 GIVING CAMPO-5 ROUNDED ON SIZE ERROR declaración-imperativa 32.10 + 12.3 = 54.30 Redondeado a dos decimales = 54.30	<table border="1"><tr><td>5</td><td>4</td><td>3</td><td>Ø</td></tr></table>	5	4	3	Ø	CAMPO-5					
5	4	3	Ø								
e) ADD CAMPO-2 TO CAMPO-4 GIVING CAMPO-5 ROUNDED ON SIZE ERROR declaración-imperativa 256.17 + 321.00 = 577.17 Redondeado a dos decimales = 577.17	No se almacena el resultado en CAMPO-6 ya que existe un truncamiento. Se ejecuta la declaración-imperativa.										

**VI-4.4.2. Verbo SUBTRACT**

**OBJETO**

El verbo SUBTRACT permite restar el contenido de un campo o de un literal o la suma del contenido de varios campos y/o literales del contenido de un campo elemental y almacenar el resultado de la sustracción en uno o varios campos elementales.

**FORMATOS**

**PARTICULARIDADES DE SU USO**

Existen tres formatos posibles, que son los indicados en la figura VI-5 (formatos 2-1 a 2-3).

Con el primer formato, un literal o un campo, representado por operando-1, o la suma de dos o más literales o campos, representados por operando-1, operando-2, ... se resta de un único campo representado por operando-n. En este campo se almacena también el resultado de la sustracción.

Con el segundo formato, un literal o un campo, representado por operando-1, o la suma de dos o más literales o campos, representados por operando-1, operando 2, ... se resta de un literal o de un campo representado por operando-n. El resultado de la sustracción se almacena en cada uno de los campos representados por los distintos operandos de opción-1.

Con el tercer formato, cada uno de los campos elementales, de los campos de grupo representado por operando-1, se resta de cada uno de los campos elementales de los campos de grupo representado por operando-2 ... siempre y cuando los campos elementales de dichos operandos cumplan las mismas condiciones establecidas en la opción CORRESPONDING del verbo MOVE.

**EJEMPLOS**

Suponiendo que el contenido y PICTURE de los campos: CAMPO-1, CAMPO-2, CAMPO-3, CAMPO-4 y CAMPO-5 sean los siguientes:

CAMPO-1	PICTURE	999V99	Contenido	<table border="1"><tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	2	3	4	5	6
2	3	4	5	6					
CAMPO-2	PICTURE	999V99	»	<table border="1"><tr><td>2</td><td>2</td><td>3</td><td>1</td><td>5</td></tr></table>	2	2	3	1	5
2	2	3	1	5					
CAMPO-3	PICTURE	99V99	»	<table border="1"><tr><td>2</td><td>8</td><td>3</td><td>7</td></tr></table>	2	8	3	7	
2	8	3	7						
CAMPO-4	PICTURE	99V99	»	<table border="1"><tr><td>2</td><td>4</td><td>∅</td><td>5</td></tr></table>	2	4	∅	5	
2	4	∅	5						
CAMPO-5	PICTURE	999V99	»	<table border="1"><tr><td>3</td><td>3</td><td>∅</td><td>5</td></tr></table>	3	3	∅	5	
3	3	∅	5						

Los resultados de aplicar las siguientes órdenes de resta serán:

	RESULTADOS	Que se almacenan en										
a) SUBTRACT CAMPO-1 FROM CAMPO-5 ROUNDED 330.5 — 234.56 = 95.44 Redondeado a un decimal = 95.4	<table border="1"><tr><td>∅</td><td>9</td><td>5</td><td>4</td></tr></table> ↑	∅	9	5	4	CAMPO-5						
∅	9	5	4									
b) SUBTRACT CAMPO-4 22.50 FROM CAMPO-1 CAMPO-2 234.56 — (24.05 + 22.50) = 188.01 223.15 — (24.05 + 22.50) = 76.60	<table border="1"><tr><td>1</td><td>8</td><td>8</td><td>∅</td><td>1</td></tr></table> ↑ <table border="1"><tr><td>∅</td><td>7</td><td>6</td><td>6</td><td>∅</td></tr></table> ↑	1	8	8	∅	1	∅	7	6	6	∅	CAMPO-1 CAMPO-2
1	8	8	∅	1								
∅	7	6	6	∅								
c) SUBTRACT CAMPO-3 CAMPO-4 FROM CAMPO-1 GIVING CAMPO-5 ROUNDED CAMPO-2 234.56 — (28.37 + 24.05) = 182.14 Redondeado a un decimal = 182.1	<table border="1"><tr><td>1</td><td>8</td><td>2</td><td>1</td></tr></table> ↑ <table border="1"><tr><td>1</td><td>8</td><td>2</td><td>1</td><td>4</td></tr></table> ↑	1	8	2	1	1	8	2	1	4	CAMPO-5 CAMPO-2	
1	8	2	1									
1	8	2	1	4								

### VI-4.4.3. Verbo MULTIPLY

**OBJETO**

El verbo MULTIPLY permite multiplicar un literal o el contenido de un campo elemental por el contenido de un campo elemental y almacenar el resultado del producto en uno o varios campos elementales.

**FORMATOS**

Existen dos formatos posibles, que son los indicados en la figura VI-5 (formatos 3-1 y 3-2).

**PARTICULARIDADES DE SU USO**

Con el primer formato, un nombre de dato o un literal, representado por operando-1, se multiplica por los nombres de datos representados por operando-2 ... El resultado del producto se almacena con los campos que participan como multiplicandos, es decir, operando-2 ...

Con el segundo formato, un nombre de dato o un literal, representado por operando-1 se multiplica por un literal o por un nombre de dato representado por operando-2. El resultado del producto se almacena en cada uno de los campos representados por los distintos operandos de opción-1.

**EJEMPLOS**

Suponiendo que el contenido y PICTURE de los campos: CAMPO-1, CAMPO-2, CAMPO-3 y CAMPO-4 sean:

CAMPO-1	PICTURE	999V9	Contenido	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	
1	2	3	4						
CAMPO-2	PICTURE	9V99	»	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr></table>	1	2	3		
1	2	3							
CAMPO-3	PICTURE	999V9	»	<table border="1"><tr><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	2	3	4	5	
2	3	4	5						
CAMPO-4	PICTURE	999V99	»	<table border="1"><tr><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table>	3	4	5	6	7
3	4	5	6	7					

Los resultados de aplicar las siguientes órdenes de multiplicación serán:

	RESULTADOS	Que se almacenan en																		
a) MULTIPLY CAMPO-2 BY CAMPO-1 ROUNDED CAMPO-3 $1.23 \times 123.4 = 151.782$ Redondeado a un decimal = 151.8 $1.23 \times 234.5 = 288.435$	<table border="1"> <tr><td>1</td><td>5</td><td>1</td><td>8</td></tr> <tr><td></td><td></td><td>↑</td><td></td></tr> <tr><td>2</td><td>8</td><td>8</td><td>4</td></tr> <tr><td></td><td></td><td>↑</td><td></td></tr> </table>	1	5	1	8			↑		2	8	8	4			↑		CAMPO-1  CAMPO-3		
1	5	1	8																	
		↑																		
2	8	8	4																	
		↑																		
b) MULTIPLY CAMPO-1 BY CAMPO-2 ON SIZE ERROR declaración-imperativa $123.4 \times 1.23 = 151.782$	No se almacena el producto, pues existe truncamiento de posiciones enteras. Se ejecuta la declaración imperativa.																			
c) MULTIPLY CAMPO-2 BY CAMPO-1 GIVING CAMPO-3 CAMPO-4 ROUNDED $1.23 \times 123.4 = 151.782$ Redondeado a dos decimales = 151.78	<table border="1"> <tr><td>1</td><td>5</td><td>1</td><td>7</td></tr> <tr><td></td><td></td><td>↑</td><td></td></tr> <tr><td>1</td><td>5</td><td>1</td><td>7</td><td>8</td></tr> <tr><td></td><td></td><td></td><td>↑</td><td></td></tr> </table>	1	5	1	7			↑		1	5	1	7	8				↑		CAMPO-3  CAMPO-4
1	5	1	7																	
		↑																		
1	5	1	7	8																
			↑																	

#### VI-4.4.4. Verbo DIVIDE

**OBJETO**

El verbo DIVIDE permite dividir varios literales y/o el contenido de uno o varios campos elementales entre un literal o entre el contenido de un campo elemental. El cociente se almacena en uno o varios campos elementales.

**FORMATOS**

Existen tres formatos posibles, que son los indicados en la figura VI-5 (formatos 4-1 a 4-3).

**PARTICULARIDADES DE SU USO**

Con el primer formato, varios campos elementales, representados por operando-2 ... se dividen entre un literal o un campo elemental, representado por operando-1. Los diversos cocientes se almacenan, respectivamente, en los campos que participan como dividendos, es decir, operando-2 ...

Con el segundo formato, un literal o un campo elemental, representado por operando-2, se divide entre un literal o un campo elemental, representado por operando-1. El cociente se almacena en cada uno de los campos, representados por los distintos operandos de opción-1.

Con el tercer formato, un literal o un campo elemental, representado por operando-1, se divide entre un literal o un campo elemental, representado por operando-2. El cociente se almacena en cada uno de los campos, representados por los distintos operandos de opción-1.

**OPCIONES**

En la versión ANSI COBOL existe la posibilidad de utilizar la opción [REMAINDER nombre-dato]. Esta opción permite almacenar el resto de la división en el campo, representado por nombre-dato.

**EJEMPLOS**

Utilizando los mismos campos y valores que en el ejemplo anterior, los resultados de aplicar las siguientes órdenes de división serán:

	RESULTADOS	Que se almacenan en
a) DIVIDE CAMPO-2 INTO CAMPO-1 ROUNDED CAMPO-3 $123.4 : 1.23 = 100.32$ Redondeado a un decimal = 100.3 $234.5 : 1.23 = 190.650$		CAMPO-1  CAMPO-3
b) DIVIDE CAMPO-2 INTO CAMPO-3 GIVING CAMPO-4 ROUNDED $234.5 : 1.23 = 190.650$ Redondeado a dos decimales = 190.65		CAMPO-4
c) DIVIDE CAMPO-2 BY CAMPO-3 GIVING CAMPO-4 $1.23 : 234.5 = 0.005$ Redondeado a dos decimales = 0.001		CAMPO-4

#### VI-4.4.5 Verbo COMPUTE

**OBJETO**

El verbo COMPUTE permite transferir el valor de un literal, de un nombre de dato o el valor de una expresión aritmética a uno o varios campos elementales.

**FORMATO**

**PARTICULARIDADES DE SU USO**

El único formato de este verbo es el indicado en la figura VI-5 (formato 5-1).

El uso de un literal o un nombre de dato, como operando-n, es equivalente a la utilización del verbo MOVE desde dicho nombre de dato, o para el literal, a los campos representados por operando-1, operando-2 ...

La forma más usual de utilizar el verbo COMPUTE es con una expresión aritmética como operando-n. En este caso se produce la evaluación de dicha expresión, con las reglas vistas en el capítulo II y el resultado de la evaluación se almacena en cada uno de los campos representados por operando-1, operando-2 ...

**EJEMPLOS**

Suponiendo que A, B, C, D y E sean los nombres de dato de diversos campos cuyas PICTURE y contenidos son los siguientes:

A	PICTURE	99V99	Contenido	<table border="1"><tr><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	2	3	4	5	
2	3	4	5						
B	PICTURE	999V99	»	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5
1	2	3	4	5					
C	PICTURE	9V9	»	<table border="1"><tr><td>2</td><td>3</td></tr></table>	2	3			
2	3								
D	PICTURE	99V9	»	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr></table>	8	7	6		
8	7	6							
E	PICTURE	999V9	»	<table border="1"><tr><td>2</td><td>4</td><td>5</td><td>6</td></tr></table>	2	4	5	6	
2	4	5	6						

Los resultados de aplicar las siguientes órdenes de cálculo serán:

	RESULTADOS	Que se almacenan en									
a) COMPUTE A = 27.34	<table border="1"><tr><td>2</td><td>7</td><td>3</td><td>4</td></tr></table> ↑	2	7	3	4	A					
2	7	3	4								
b) COMPUTE A ROUNDED B = (B/A) ** 2 Evaluación de $\left(\frac{B}{A}\right)^2 = \left(\frac{123.45}{23.45}\right)^2 = 27.6676$ Redondeado a dos decimales = 27.67	<table border="1"><tr><td>2</td><td>7</td><td>6</td><td>6</td></tr></table> ↑ <table border="1"><tr><td>∅</td><td>2</td><td>7</td><td>6</td><td>7</td></tr></table> ↑	2	7	6	6	∅	2	7	6	7	A B
2	7	6	6								
∅	2	7	6	7							
c) COMPUTE A = D	<table border="1"><tr><td>8</td><td>7</td><td>6</td><td>∅</td></tr></table> ↑	8	7	6	∅	A					
8	7	6	∅								
d) COMPUTE A = E ON SIZE ERROR declaración-imperativa	No se almacena el resultado y se ejecuta la declaración-imperativa.										

Referencia	FORMATOS	APLICACION EN LOS COMPILADORES							
		CDC Serie 6000	IBM Series 360/370	UNIVAC 1108	CII IRIS-50	HW-B GE 600	Siemens 4004	UNIVAC 9200/9300	HW-B GE-100
1-1	<b>ADD</b> operando-1 [operando-2] ... operando-n [opción-2] [opción-3]	*							
1-2	<b>ADD</b> operando-1 [operando-2] ... $\left\{ \begin{array}{l} \text{opción-1} \\ \text{TO operando m [opción-2] [operando-n]} \end{array} \right.$ [opción-2] .. $\left\{ \begin{array}{l} \text{opción-3} \end{array} \right.$	*	*	*	*	*	*	*	*
1-3	<b>ADD</b> operando-1 [operando-2] ... <b>TO</b> operando-p [opción-1] [opción-3]	*							
1-4	<b>ADD</b> $\left\{ \begin{array}{l} \text{CORRESPONDING} \\ \text{CORR} \end{array} \right\}$ operando-1 <b>TO</b> operando-2 [opción-2] [operando-3 [opción-2] ] .. [opción-3 ]	*	*	*			*		
2-1	<b>SUBTRACT</b> operando-1 [operando-2] ... <b>FROM</b> operando-m [opción-2] [operando-n [opción-2] ] ... [opción-3 ]	*	*	*	*		*	*	
2-2	<b>SUBTRACT</b> operando-1 [operando-2] ... <b>FROM</b> operando-n [opción-1] [opción-3]	*	*	*	*		*	*	
2-3	<b>SUBTRACT</b> $\left\{ \begin{array}{l} \text{CORRESPONDING} \\ \text{CORR} \end{array} \right\}$ operando-1 <b>FROM</b> operando-2 [opción-2] [operando-3 [opción-2] ] .. [opción-3 ]	*	*	*			*		

FIGURA VI-5 (1.ª parte)



Referencia	FORMATOS	APLICACION EN LOS COMPILADORES						
		CDC Serie 6000	IBM Series 360/370	UNIVAC 1108	CII IRIS-50	HW-B GE 600	Siemens 4004	UNIVAC 9200/9300
3-1	<b>MULTIPLY</b> operando-1 <b>BY</b> operando-2 [opción-2] [ operando-3 [opción-2] ] ... [opción-3]	*	*	*	*	*	*	*
3-2	<b>MULTIPLY</b> operando-1 <b>BY</b> operando-2 [opción-1] [opción-3]	*	*	*	*	*	*	*
4-1	<b>DIVIDE</b> operando-1 <b>INTO</b> operando-2 [opción-2] [ operando-3 [opción-2] ] ... [opción-3]	*	*	*	*	*	*	*
4-2	<b>DIVIDE</b> operando-1 <b>INTO</b> operando-2 [opción-1] [opción-3]	*	*	*	*	*	*	*
4-3	<b>DIVIDE</b> operando-1 <b>BY</b> operando-2 [opción-1] [opción-3]	*	*	*	*	*	*	*
5-1	<b>COMPUTE</b> operando-1 [opción-2] [ operando-2 [opción-2] ] ... { <b>FROM</b> <b>EQUAL</b> } { literal expresión aritmética nombre-dato } [opción-3]	*	*	*	*	*	*	*

OBSERVACIONES: Se puede incluir el signo , entre operandos para mayor claridad del formato. Nótese la similitud de formatos entre los de referencia 1-3, 2-2, 3-2, 4-2 y 4-3, 1-4 y 2-3; 3-1 y 4-1 (Entre ellos sólo varían el verbo y las palabras reservadas).

FIGURA VI-5 (2.ª parte)



V - 5

VERBOS DE CONTROL  
DE SECUENCIA

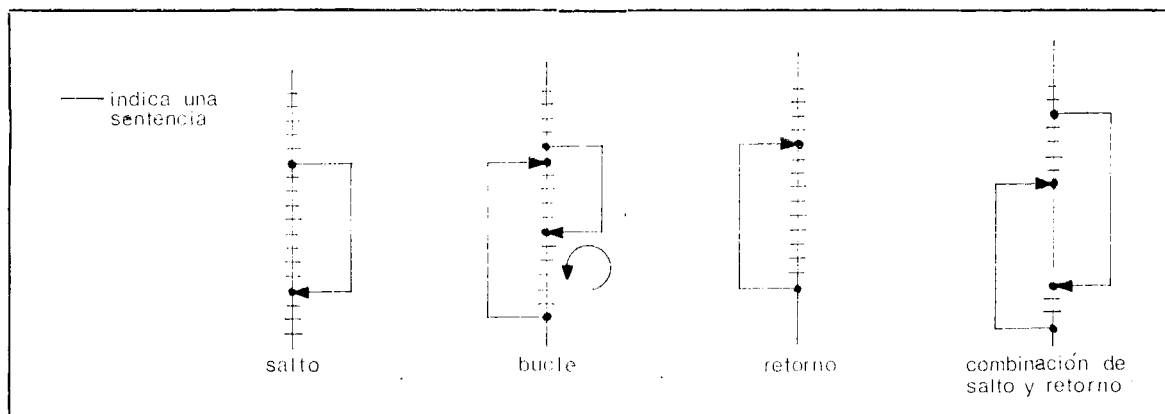


## VI-5. VERBOS DE CONTROL DE SECUENCIA

### VI-5.1. GENERALIDADES DE LOS VERBOS DE CONTROL DE SECUENCIA

Con los verbos de programa estudiados hasta este apartado el programa se ejecutaría de una forma secuencial, es decir, sentencia a sentencia, en el mismo orden que éstas aparecen en la PROCEDURE DIVISION del programa fuente.

Sin embargo, en la mayoría de los programas se debe alterar, en algún modo, esta secuencia normal en alguno de los siguientes aspectos: omitiéndose alguno de los párrafos contenidos en el programa (saltos de secuencia), repitiendo un determinado número de veces uno o varios párrafos (bucles), retrocediendo a párrafos anteriormente ejecutados (retornos) o combinaciones de estos aspectos.



Estas alteraciones de la ejecución secuencial de instrucciones se realiza mediante los verbos de control de secuencia. En su estudio vamos a establecer tres tipos o categorías.

- Verbo de decisión o comparación.
- Verbos de bifurcación.
- Verbos complementarios.

## VI-5.2. VERBO DE DECISION O COMPARACION

### VI-5.2.1. Pseudo-verbo IF o declaración IF

**OBJETO**

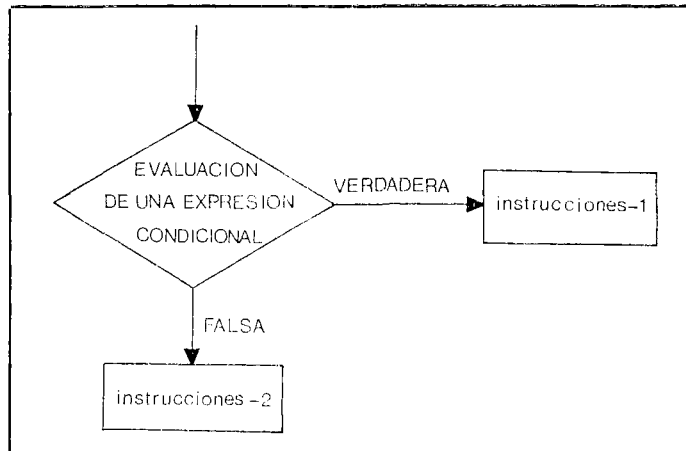
La orden IF permite ejecutar la evaluación de una expresión condicional (ver capítulo II, epígrafe II-2.1.4).

**FORMATO**

```
IF expresión-condicional { opción-1 }
{ OTHERWISE } { opción-2 }
{ ELSE } { opción-2 }
```

**PARTICULARIDADES DE SU USO**

Este verbo debe utilizarse en todas las partes de un programa en los que como resultado de la evaluación de una expresión condicional (verdadera o falsa) deba ejecutarse una u otra (entre dos) determinada secuencia de instrucciones.



**OPCIONES**

- La opción-1, siempre obligatoria, permite indicar el camino a seguir en el caso de resultar la evaluación de la expresión condicional VERDADERA.

Tiene dos posibilidades:

```
opción-1 = { declaración-1
            NEXT SENTENCE (sólo aplicable cuando existe opción-2) }
```

*declaración-1*, puede ser una declaración imperativa o una declaración condicional. En este último caso se produce una sucesión de declaraciones condicionales en cascada, también denominadas IF jerarquizadas.

Quando el resultado de la evaluación es FALSA y no existe **opción-2** el programa sigue su curso normal de ejecución en secuencia.

- La opción-2, no obligatoria, permite indicar el camino a seguir en el caso de resultar la evaluación de la expresión condicional FALSA.

Tiene dos posibilidades:

$$\boxed{\text{opción-2}} = \left\{ \begin{array}{l} \text{declaración-2} \\ \text{NEXT SENTENCE} \end{array} \right\}$$

*declaración-2* tiene las mismas particularidades que *declaración-1* en  $\boxed{\text{opción-1}}$ .

NEXT SENTENCE indica que el camino a seguir es el de la secuencia normal de ejecución del programa.

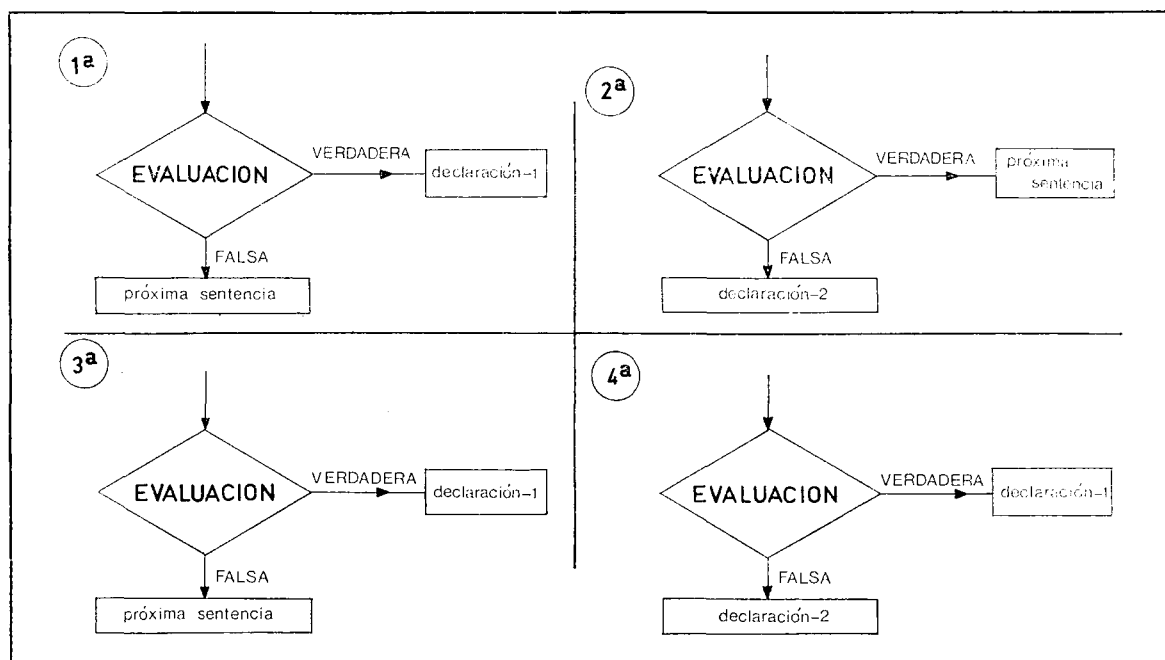
Las opciones NEXT SENTENCE de  $\boxed{\text{opción-1}}$  y  $\boxed{\text{opción-2}}$  no deben estar presentes simultáneamente, ya que no existiría decisión.

Delante de  $\boxed{\text{opción-1}}$  y de las palabras reservadas OTHERWISE o ELSE puede emplearse la palabra reservada THEN.

En resumen, las cuatro posibilidades existentes son:

- 1.ª IF expresión-condicional declaración-1.
- 2.ª IF expresión-condicional NEXT SENTENCE ELSE declaración-2.
- 3.ª IF expresión-condicional declaración-1 ELSE NEXT SENTENCE (equivalente a 1.ª).
- 4.ª IF expresión-condicional declaración-1 ELSE declaración-2.

La representación gráfica de cada una de las posibilidades es:



**EJEMPLOS**

a) IF A + B < 2 MOVE A TO B. Se efectuará el movimiento del campo A al B cuando el valor de la suma de los campos A y B sea menor que 2.

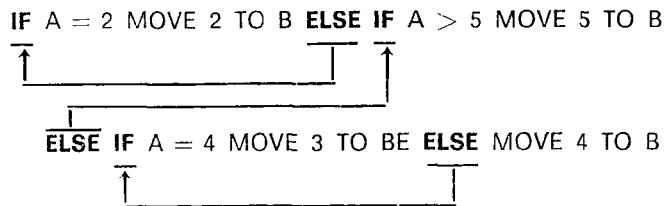
- |       |                                                                                             |                                                                                                                           |
|-------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| b)    | IF A < 2 NEXT SENTENCE ELSE<br>ADD 2 TO A.                                                  | Se suma 2 al valor del campo A cuando el valor del campo A sea igual o mayor que 2.                                       |
| <hr/> |                                                                                             |                                                                                                                           |
| c)    | IF A + B < 2 MOVE A TO B<br>ELSE NEXT SENTENCE.                                             | Es equivalente al primer ejemplo.                                                                                         |
| <hr/> |                                                                                             |                                                                                                                           |
| d)    | IF A = 2 ADD 2 TO A ELSE<br>SUBTRACT 2 FROM A.                                              | Se suma 2 al valor del campo A cuando el valor de este campo es 2, y en caso contrario (< o >) se resta 2 del valor de A. |
| <hr/> |                                                                                             |                                                                                                                           |
| e)    | IF A = MOVE 2 TO B ELSE IF A > 5 MOVE 5 TO B ELSE IF A = 4<br>MOVE 3 TO B ELSE MOVE 4 TO B. |                                                                                                                           |

Este es un ejemplo de un IF jerarquizado. Es obvio que para el lector supone una dificultad conocer en cada caso que IF pertenece a la jerarquía de otro IF. Por ello no debe proliferarse el uso de los IF jerarquizados, ya que dificultan la comprensión del programa.

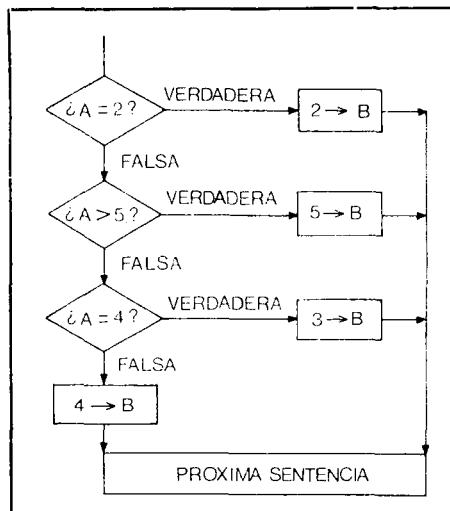
Sin embargo, la norma a seguir para correlacionar cada IF con su correspondiente ELSE es:

Se relaciona el último ELSE existente con el IF más próximo precedente, si no existe entre ambos otro ELSE. En este caso es este último ELSE el que se relaciona con el IF.

En el ejemplo que estamos tratando se tendrá:



El organigrama correspondiente sería:





La ejecución de la orden, equivale para los distintos valores de A, a:

Valor de A	Se almacena en B
$A < 2$	4
$A = 2$	2
$2 < A < 4$	4
$A = 4$	3
$4 < A \leq 5$	4
$A > 5$	5

### VI-5.3. VERBOS DE BIFURCACION

#### VI-5.3.1. Verbo GO TO

##### OBJETO

El verbo GO TO permite efectuar una bifurcación condicional o incondicional, en la secuencia del programa, a otra parte del mismo identificada por un nombre de procedimiento sin retornar al punto del programa en donde se ha realizado la bifurcación.

##### FORMATO

Existen dos posibles formatos:

*Para bifurcaciones incondicionales.*

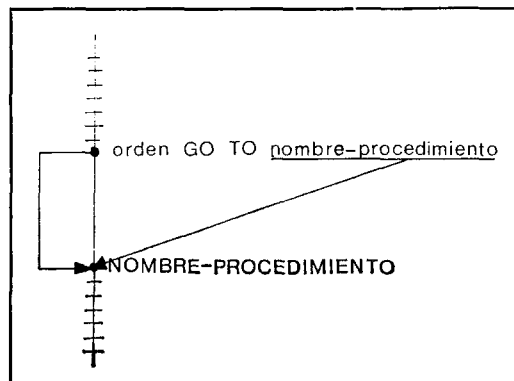
**GO TO** [nombre-procedimiento]

*Para bifurcaciones condicionales.*

**GO TO** nombre-procedimiento-1  
[nombre-procedimiento-2] ... **DEPENDING ON** nombre-dato

##### PARTICULARIDADES DE SU USO

*El primer formato*, denominado bifurcación incondicional o bifurcación simple, permite interrumpir la ejecución secuencial del programa para continuar en otra parte del mismo identificada por nombre de procedimiento.



Se puede omitir la escritura del nombre de procedimiento, siempre y cuando se haya ejecutado, previamente al verbo GO TO, un verbo ALTER que haya hecho referencia al párrafo en el cual se encuentra el verbo GO TO (ver epígrafe VI-5.3.2.).

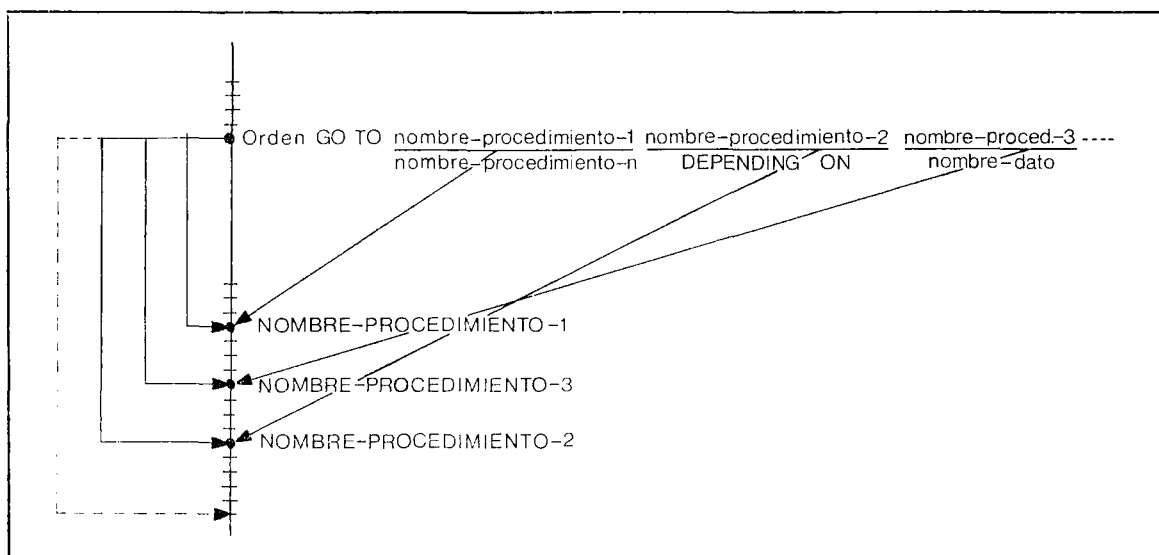
Con el segundo formato, denominado bifurcación condicional o múltiple, se permite interrumpir la ejecución secuencial del programa para continuar en otras partes del mismo identificadas por nombre-procedimiento-1, nombre-procedimiento-2, ... El párrafo o procedimiento cuyas sentencias pasan a ser ejecutadas dependen del valor que tenga *nombre-dato* en el momento de la ejecución del verbo.

Si *nombre-dato* tiene valor 1, se pasa a ejecutar las sentencias contenidas en el nombre de procedimiento escrito en primer lugar (nombre-procedimiento-1).

Si el *nombre-dato* tiene valor 2, se pasa a ejecutar las sentencias contenidas en el nombre de procedimiento escrito en segundo lugar (nombre-procedimiento-2).

En general, si *nombre-dato* tiene un valor n (entero y positivo), se pasa a ejecutar las sentencias contenidas en el nombre de procedimiento escrito en N-SIMO lugar (nombre-procedimiento-n).

Cuando el valor de *nombre-dato* es negativo, cero o mayor que el número de nombres de procedimientos especificados en la orden GO TO, el programa sigue su curso normal de ejecución secuencial de sentencias.



**EJEMPLOS**

a) GO TO CALCULO-PRIMA.

En este ejemplo, el programa, al ejecutarse la orden GO TO, bifurca a ejecutar las sentencias contenidas en el párrafo CALCULO-PRIMA.

b) IF A > 2 GO TO CALCULO-HABERES ELSE GO TO CALCULO-DOCUMENTOS.

En este ejemplo el programa, al ejecutarse la evaluación de la expresión  $A > 2$ , bifurca a ejecutar las sentencias contenidas en el párrafo CALCULO-HABERES cuando el valor del campo A es mayor que 2 y a ejecutar las sentencias contenidas en el párrafo CALCULO-DESCUENTOS cuando el valor de A es igual o inferior a 2.

```
c) GO TO A1 A2 A3 A4 A5 DEPENDING ON X.
    MOVE A TO B GO TO FIN.
A1.
    MOVE A TO C GO TO FIN.
A2.
    MOVE A TO D GO TO FIN.
A3.
    MOVE A TO E GO TO FIN.
A4.
    MOVE A TO F GO TO FIN
A5.
    MOVE A TO G GO TO FIN.
```

Con este ejemplo la ejecución del verbo GO TO efectúa:

- Un salto al párrafo identificado por A1 cuando  $X = 1$ , equivale a GO TO A1.
- Un salto al párrafo identificado por A2 cuando  $X = 2$ , equivale a GO TO A2.
- Un salto al párrafo identificado por A3 cuando  $X = 3$ , equivale a GO TO 3.
- Un salto al párrafo identificado por A4 cuando  $X = 4$ , equivale a GO TO A4.
- Un salto al párrafo identificado por A5 cuando  $X = 5$ , equivale a GO TO A5.
- Un salto a la siguiente sentencia (MOVE A TO B GO TO FIN) cuando  $5 < X < 1$ , equivale a la ausencia de la instrucción.

#### VI-5.3.2. Verbo ALTER

**OBJETO**

El verbo ALTER permite alterar el camino indicado por una o varias órdenes GO TO incondicionales.

**FORMATO**

**ALTER** nombre-procedimiento-1 **TO PROCEED TO**  
 nombre-procedimiento-2  
 [nombre-procedimiento-3 **TO PROCEED TO**  
 nombre-procedimiento-4] .....

**PARTICULARIDADES  
 DE SU USO**

Para poder ejecutar el cambio o dirección especificado por una orden GO TO el verbo ALTER utiliza dos nombres de procedimiento. El primer nombre de procedimiento, nombre-procedimiento-1, indica

el párrafo en el cual está contenida la orden GO TO que se desea modificar. El segundo nombre de procedimiento, nombre-procedimiento-2, indica la nueva dirección o párrafo que pasa a sustituir al nombre de procedimiento contenido en la orden GO TO.

En el párrafo indicado por nombre-procedimiento-1 no puede existir, por consiguiente, más de un verbo GO TO.

Cuando el verbo GO TO contenido en el nombre de procedimiento al que hace referencia la orden ALTER no contenga ningún nombre de procedimiento, el verbo ALTER debe ejecutarse previamente a aquél en el total del programa. En esta ejecución el verbo ALTER especifica, y no altera, el párrafo que debe ejecutarse con el verbo GO TO (ver particularidades del uso del verbo GO TO).

#### OPCIONES

La única opción existente en el formato del verbo ALTER es la de poder coexistir varias órdenes de alteración en un mismo verbo, efectuándose en este caso tantas alteraciones como opciones + 1 se incluyen en el formato.

#### EJEMPLOS

INICIO.

GO TO CALCULO-MENSUAL.

CALCULO-TRIMESTRAL.

DIVIDE PRIMA BY 4 GIVING PTS-MES GO TO ESCRITURA.

CALCULO-MENSUAL.

DIVIDE PRIMA BY 12 GIVING PTS-MES GO TO ESCRITURA.

ESCRITURA.

MOVE PTS-MES TO PTS-EDIT WRITE REG-SAL.

PRUEBA.

IF COD = 1 NEXT SENTENCE ELSE ALTER INICIO TO

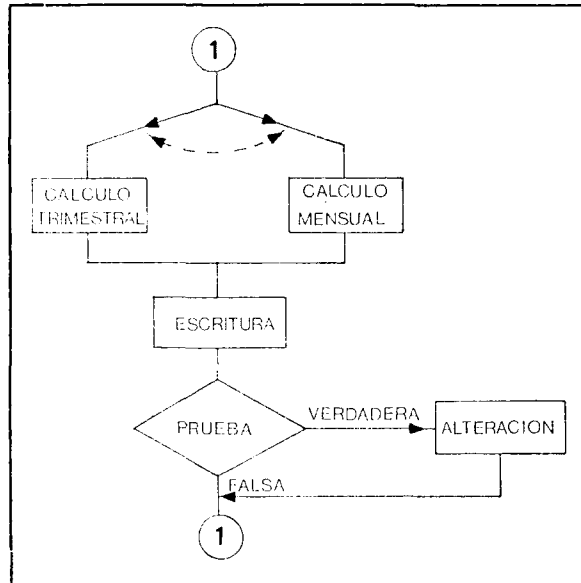
PROCEED TO CALCULO-TRIMESTRAL.

GO TO INICIO.

Estas sentencias pueden formar parte de un programa que calculan una prima mensual en función de un código que puede valer 1 ó 3. Si el código vale 1 se debe dividir por 12; si el código vale 3 se debe dividir por 4.

Obsérvese que el párrafo denominado PRUEBA efectúa la alteración de la dirección indicada por el párrafo INICIO que bifurca a CALCULO-MENSUAL.

Cuando COD no es 1, la sentencia IF, a través del verbo ALTER, modifica la dirección del párrafo INICIO por la del párrafo CALCULO-TRIMESTRAL. El ordinograma correspondiente sería:



Es evidente que existen otras formas de resolver el problema, como, por ejemplo, el verbo GO TO ... DEPENDING ON, cuyo uso es normalmente más recomendable que el del verbo ALTER.

#### VI-5.3.3. Verbo **PERFORM**

**OBJETO**

El verbo **PERFORM** permite efectuar una bifurcación a una parte del programa, al igual que el verbo **GO TO**, para ejecutar una o varias veces un determinado procedimiento y después retornar a la sentencia siguiente a la que ha producido la bifurcación.

**FORMATOS**

Existen cuatro formatos posibles del verbo **PERFORM**.

a) *PERFORM simple*; permite ejecutar una vez un determinado procedimiento:

**PERFORM** nombre-procedimiento-1 [**THRU** nombre-procedimiento-2]

b) *PERFORM compuesto*; permite ejecutar un número fijo de veces un determinado procedimiento:

**PERFORM** nombre-procedimiento-1 [**THRU** nombre-procedimiento-2]

} n.º entero { **TIMES**  
 } nombre-dato {

c) *PERFORM condicional*; permite ejecutar un determinado procedimiento tantas veces como sea necesario hasta que se cumpla una condición:

**PERFORM** nombre-procedimiento-1 [**THRU** nombre-procedimiento-2] **UNTIL** expresión condicional

d) *PERFORM* con subíndices; permite ejecutar un determinado procedimiento un número variable de veces:

**PERFORM** nombre-procedimiento-1 [**THRU** nombre-procedimiento-2] **VARYING** subíndice-1

**FROM** { núm.-entero-1 } **BY** { núm.-entero-2 }  
 { nombre-dato-1 } { nombre-dato-2 }

**UNTIL** expresión-condicional-1

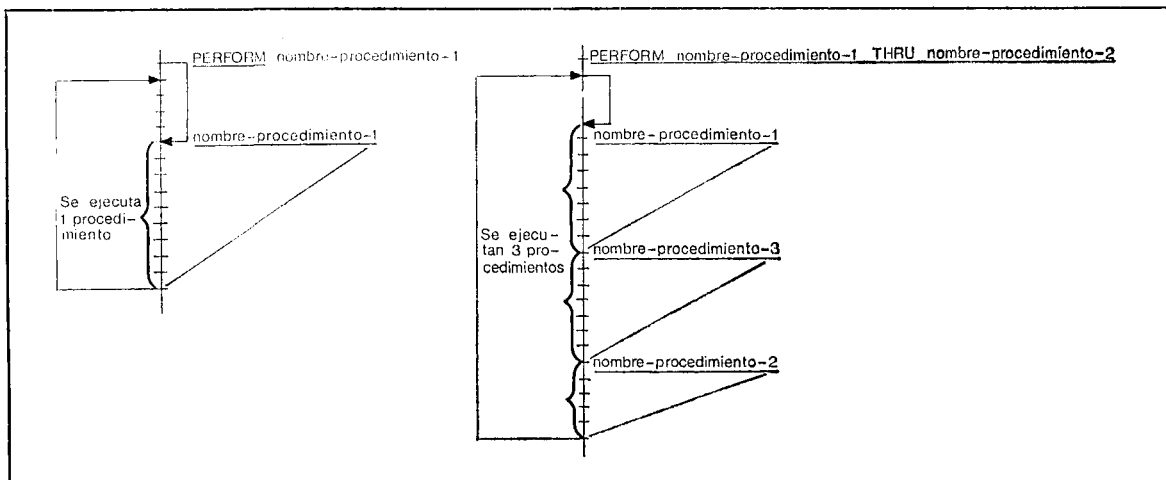
[ **AFTER** subíndice-2 **FROM** { núm.-entero-3 } **BY**  
 { núm.-entero-4 } **UNTIL** expresión-condicional-2  
 { nombre-dato-4 }

[ **AFTER** subíndice-3 **FROM** { núm.-entero-5 } **BY**  
 { núm.-entero-6 } **UNTIL** expresión-condicional-3  
 { nombre-dato-6 }

**OPCIONES**

La única opción existente, común a todos los formatos, es [THRU nombre-procedimiento-2]. Esta opción permite ampliar el campo de acción de la orden PERFORM a más de un procedimiento.

Sin la opción THRU el verbo PERFORM ejecuta un determinado número de veces el procedimiento indicado por nombre-procedimiento-1. Con la opción THRU se ejecutan todos los procedimientos existentes en el programa desde los límites indicados por nombre-procedimiento-1 hasta nombre-procedimiento-2 (ambos inclusive).

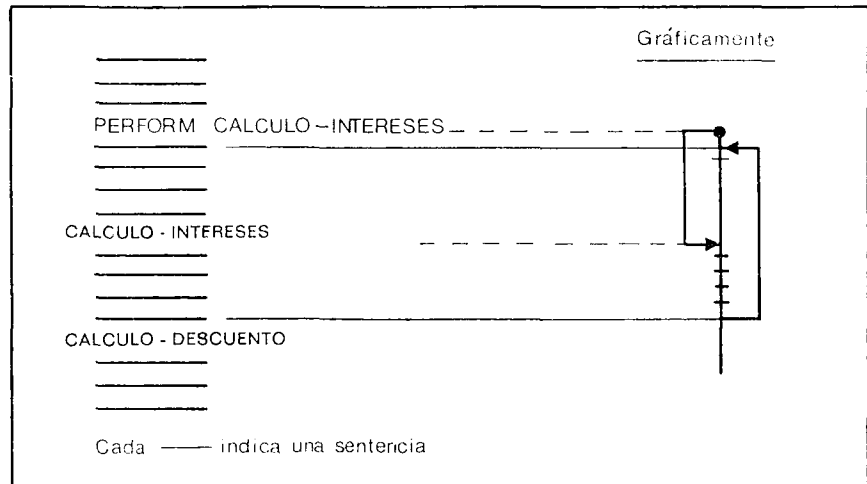


**PARTICULARIDADES DE SU USO**

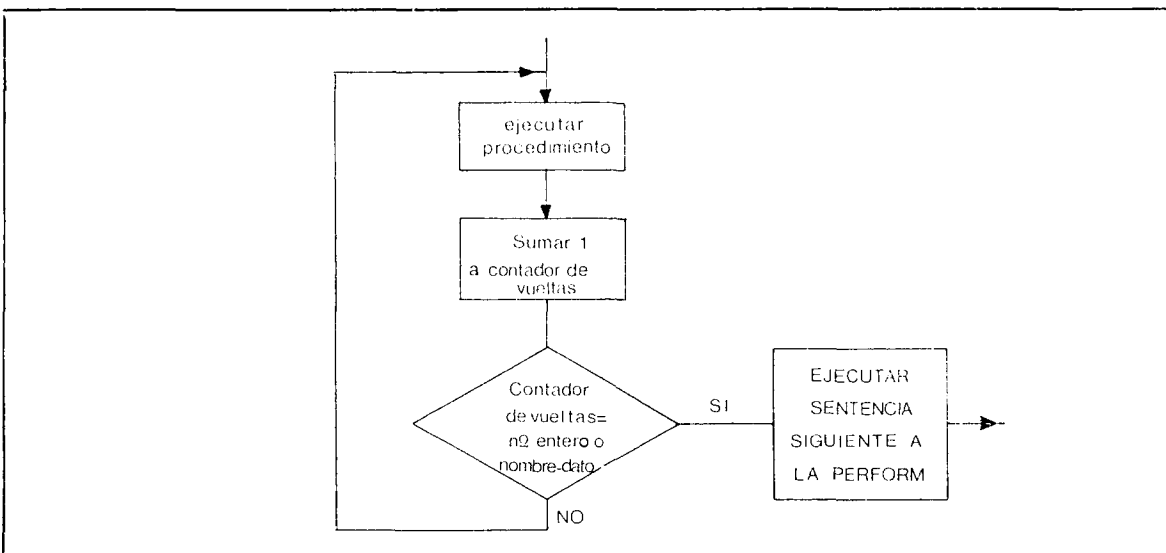
Nombre-procedimiento-1 y nombre-procedimiento-2 pueden ser un nombre de un párrafo o un nombre de una sección. En cualquiera de los casos, la orden **PERFORM** ejecuta desde la primera sentencia del primer párrafo hasta la última sentencia del último párrafo abarcado por la orden. Si por cualquier razón de la lógica del programa existe una o varias bifurcaciones, dentro del ámbito abarcado por la orden **PERFORM**, es necesario que todas se dirijan a la última sentencia del último párrafo para que éste sea ejecutado y por consiguiente la orden **PERFORM**.

Con el primer formato, como ya se ha indicado, el verbo **PERFORM** ejecuta una vez los procedimientos indicados y retorna la secuencia del programa a la siguiente sentencia, a la que ha producido la bifurcación.

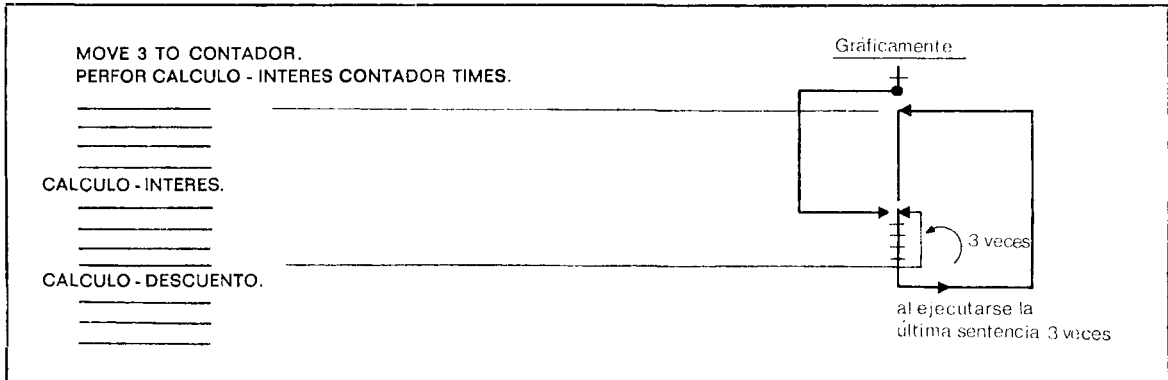
**EJEMPLO-1**



Con el segundo formato, el número de veces que se ejecutan los procedimientos indicados está determinado por el valor de número entero o por el contenido de nombre-dato en el momento de la ejecución. El ordinograma lógico equivalente a este formato es:

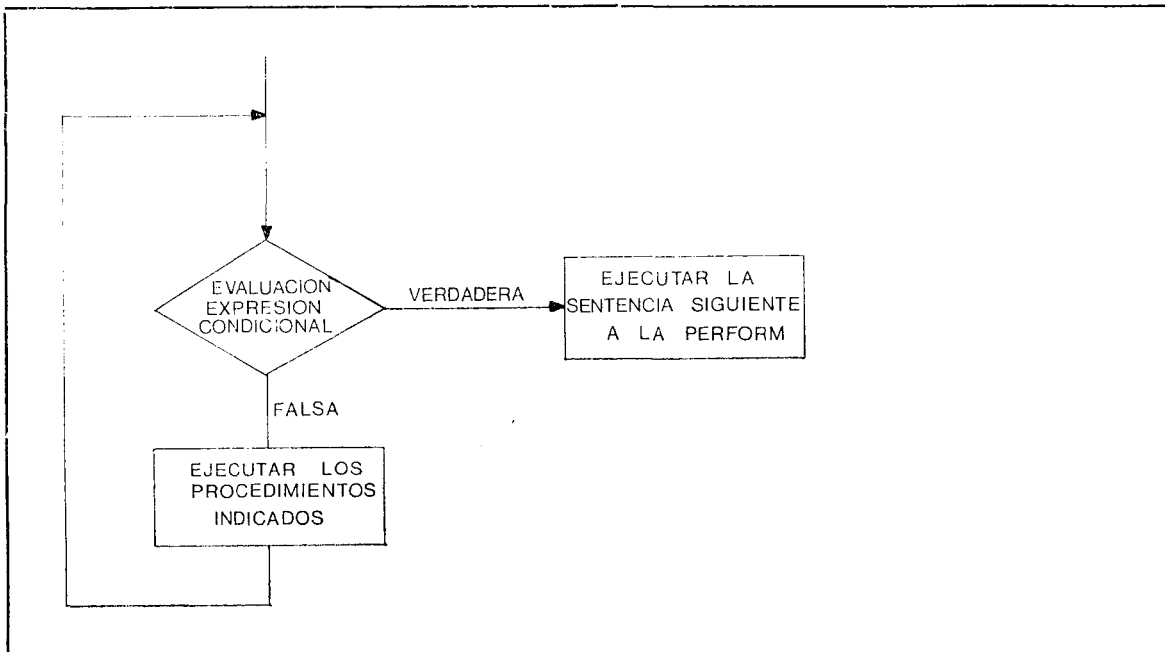


**EJEMPLO-2**



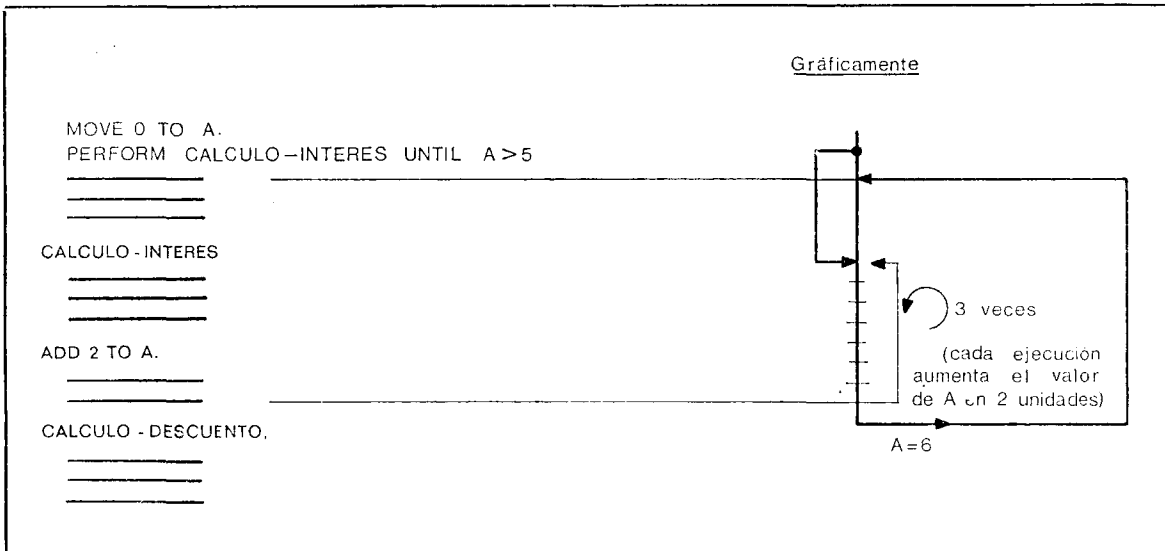
Con el tercer formato, el número de veces que se ejecutan los procedimientos indicados está determinado por el valor de la expresión condicional, contenida en la orden PERFORM. Si la expresión condicional incluida es FALSA se ejecutan los procedimientos abarcados por la orden PERFORM. Si la expresión condicional incluida es VERDADERA no se ejecutan. Es obvio indicar que si la primera evaluación de la expresión condicional da como resultado valor FALSO, ejecutándose por consiguiente los procedimientos indicados, debe existir en las mismas alguna orden que modifique alguno de los valores de los elementos que componen la expresión condicional, ya que de no ocurrir así el programa entraría en un «loop» o bucle de ejecución continuo.

El ordinograma lógico equivalente a este formato es:





**EJEMPLO-3**



Se ejecuta el procedimiento CALCULO-INTERES tres veces:

- 1.^a ejecución; valor de A = 0, se incrementa a valor 2.
- 2.^a ejecución; valor de A = 2, se incrementa a valor 4.
- 3.^a ejecución; valor de A = 4, se incrementa a valor 6.

No se ejecuta más veces, ya que se cumple la condición A > 5.

Con el cuarto formato, el número de veces que se ejecuten los procedimientos indicados está en función de:

- el número de opciones AFTER empleadas;
- la evaluación de las expresiones-condicionales;
- los valores iniciales de los subíndices utilizados especificados por  

$$\text{FROM } \left\{ \begin{array}{l} \text{núm.-entero-i} \\ \text{nombre-dato-i} \end{array} \right\} \quad i = 1, 3, 5;$$
- los incrementos de los subíndices en cada ejecución, representados por BY  $\left\{ \begin{array}{l} \text{núm.-entero-j} \\ \text{nombre-dato-j} \end{array} \right\} \quad j = 2, 4, 6.$

El uso de este formato está relacionado con la utilización de nombres de índice, y por consiguiente es necesario que existan cláusulas OCCURS relativas a los elementos que intervienen en el formato.

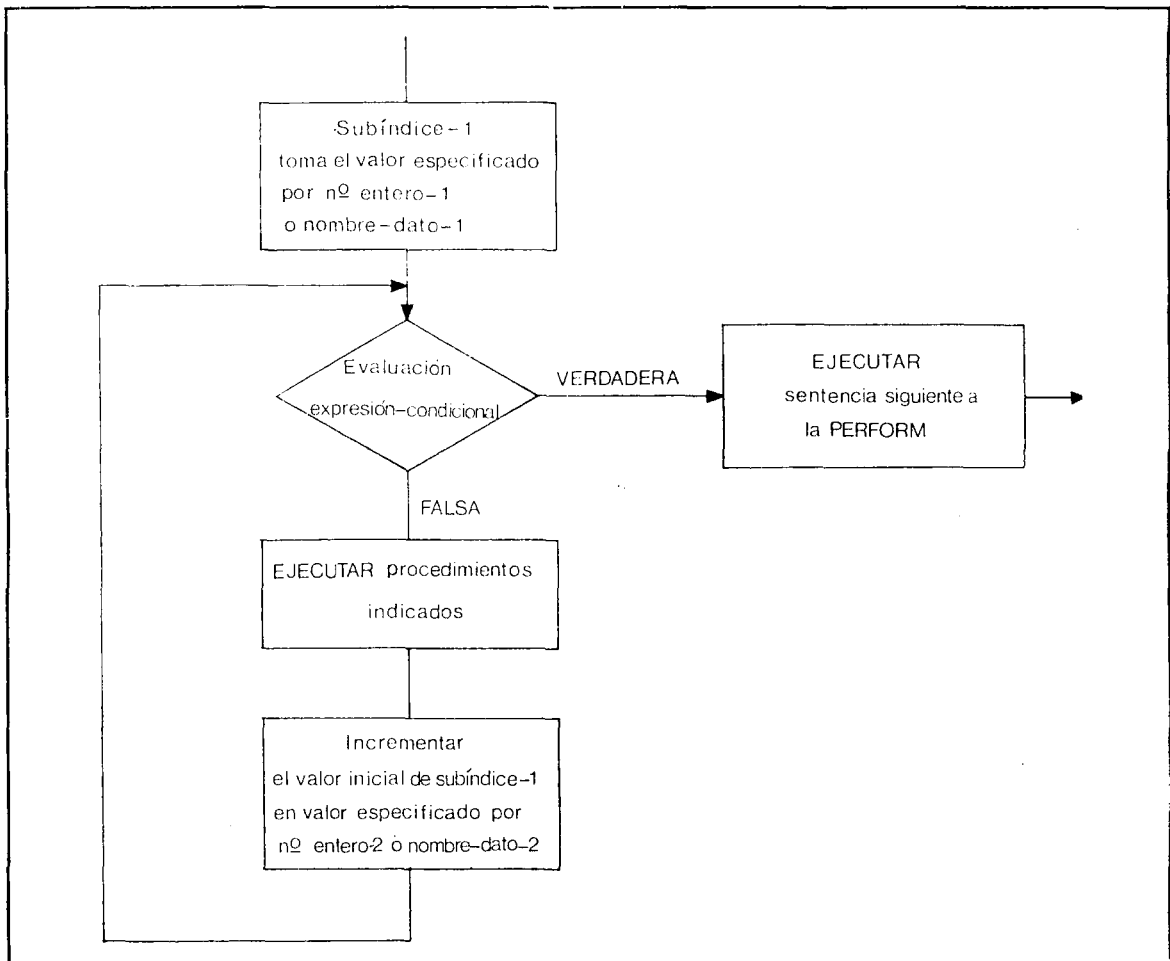
La mejor forma de expresar cómo se ejecuta una orden PERFORM con el formato cuarto es mediante sus correspondientes, o equivalentes, ordinogramas lógicos.

**Sin opción AFTER (un índice)**

PERFORM nombre-procedimiento-1 [THRU nombre-procedimiento-2]

VARYING subíndice-1 FROM { núm.-entero-1  
nombre-dato-1 } BY

{ núm.-entero-2  
nombre-dato-2 } UNTIL expresión-condicional-1



**EJEMPLO-4**

PERFORM RUTINA VARYING I FROM 1 BY 2 UNTIL I > 6.  
GO TO FIN.

RUTINA.

MOVE I TO ELEMENTO (I).

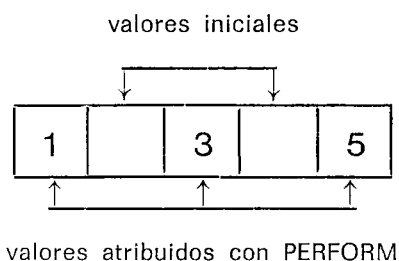
El procedimiento RUTINA se ejecuta tres veces, que corresponderán a los valores de I siguientes:

- 1.^a Para I = 1 (valor inicial) ELEMENTO (1) tendrá el valor 1
- 2.^a Para I = 3 (incremento de 2) ELEMENTO (3) tendrá el valor 3
- 3.^a Para I = 5 (incremento de 2) ELEMENTO (5) tendrá el valor 5

Si, por ejemplo, TABLA es el nombre de un campo compuesto de cinco campos elementales denominados ELEMENTO, y se ha definido como:

```
02 TABLA
03 ELEMENTO OCCURS 5 TIMES PICTURE 9...
```

Después de la ejecución de la orden PERFORM, el contenido del campo TABLA será:



**Con la opción AFTER (1 vez ~~X~~ 2 subíndices)**

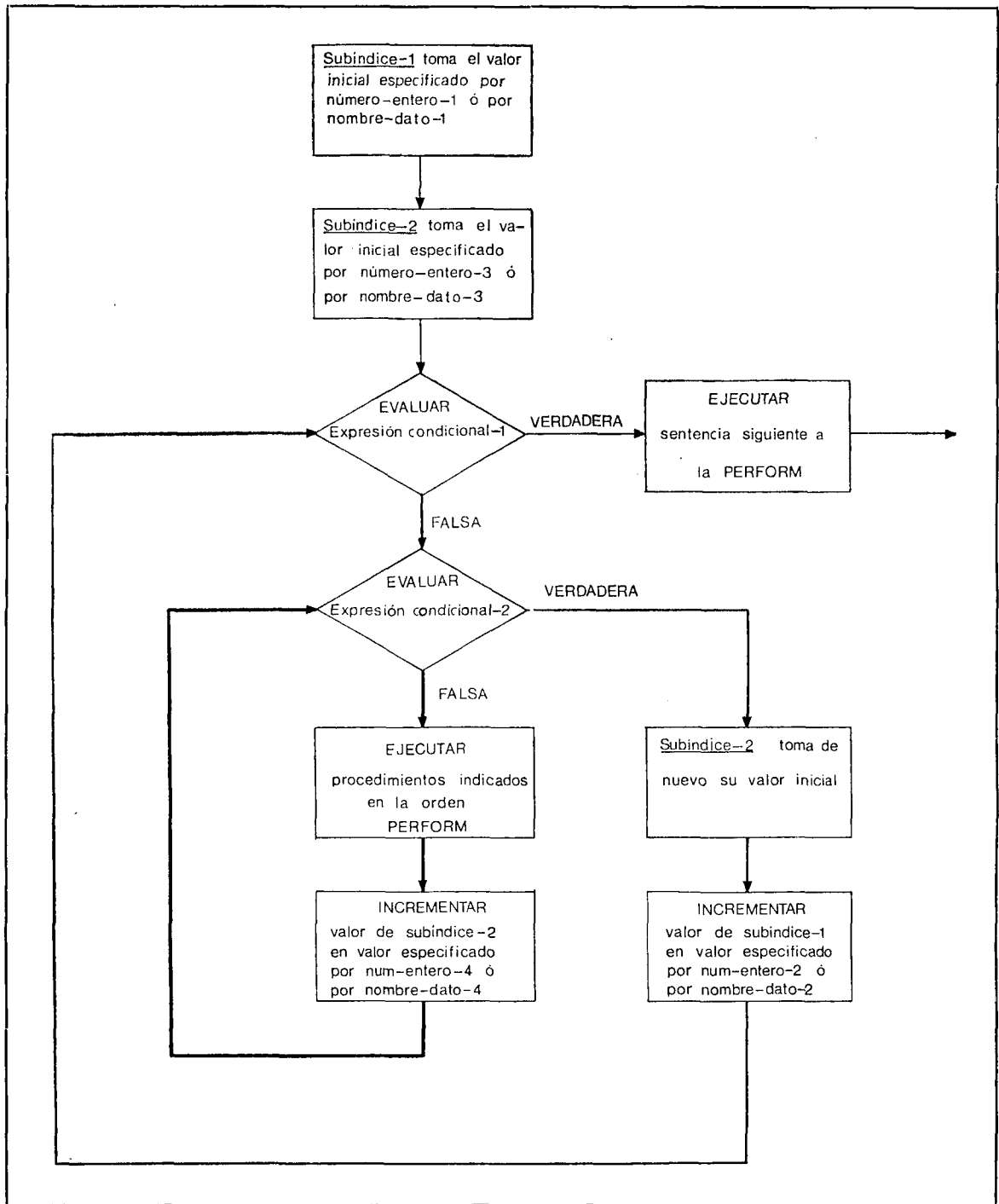
PERFORM nombre-procedimiento-1 [THRU nombre-procedimiento-2] VARYING subíndice-1

FROM { núm.-entero-1 { BY } núm.-entero-2 }  
 nombre-dato-1 { nombre-dato-2 }

UNTIL expresión-condicional-1 AFTER subíndice-2

FROM { núm.-entero-3 { BY } núm.-entero-4 }  
 nombre-dato-3 { nombre-dato-4 }

UNTIL expresión-condicional-2



**EJEMPLO-5**

PERFORM RUTINA VARYING I FROM 1 BY 2 UNTIL I > 6  
 AFTER J FROM 1 BY 1 UNTIL J = 3.  
 GO TO FIN.

RUTINA.

COMPUTE X = I + J.  
 MOVE X TO ELEMENTO (I, J).

El procedimiento RUTINA se ejecutará seis veces, que corresponderán a los valores sucesivos de I y J siguientes:

Contenido de

- |                                                                             |                      |
|-----------------------------------------------------------------------------|----------------------|
| 1. ^a ) I = 1 y J = 1 (valores iniciales)                         | ELEMENTO (1, 1) = 2. |
| 2. ^a ) I = 1 y J = 2 (se incrementa J en 1)                      | ELEMENTO (1, 2) = 3. |
| 3. ^a ) I = 3 y J = 1 (se incrementa I en 2<br>y J valor inicial) | ELEMENTO (3, 1) = 4. |
| 4. ^a ) I = 3 y J = 2 (se incrementa J en 1)                      | ELEMENTO (3, 2) = 5. |
| 5. ^a ) I = 5 y J = 1 (se incrementa I en 2<br>y J valor inicial) | ELEMENTO (5, 1) = 6. |
| 6. ^a ) I = 5 y J = 2 (se incrementa J en 1)                      | ELEMENTO (5, 2) = 7. |

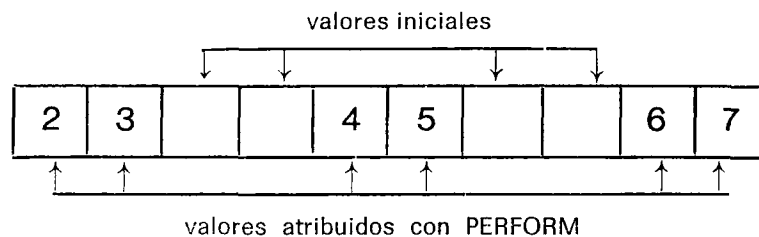
Si, por ejemplo, TABLA es un campo compuesto de diez elementos y cuya definición en la DATA DIVISION es la siguiente:

02 TABLA.

03 FILA OCCURS 5 TIMES.

04 ELEMENTO PICTURE 9 OCCURS 2 TIMES.

después de ejecutarse la orden PERFORM el contenido del campo TABLA será:



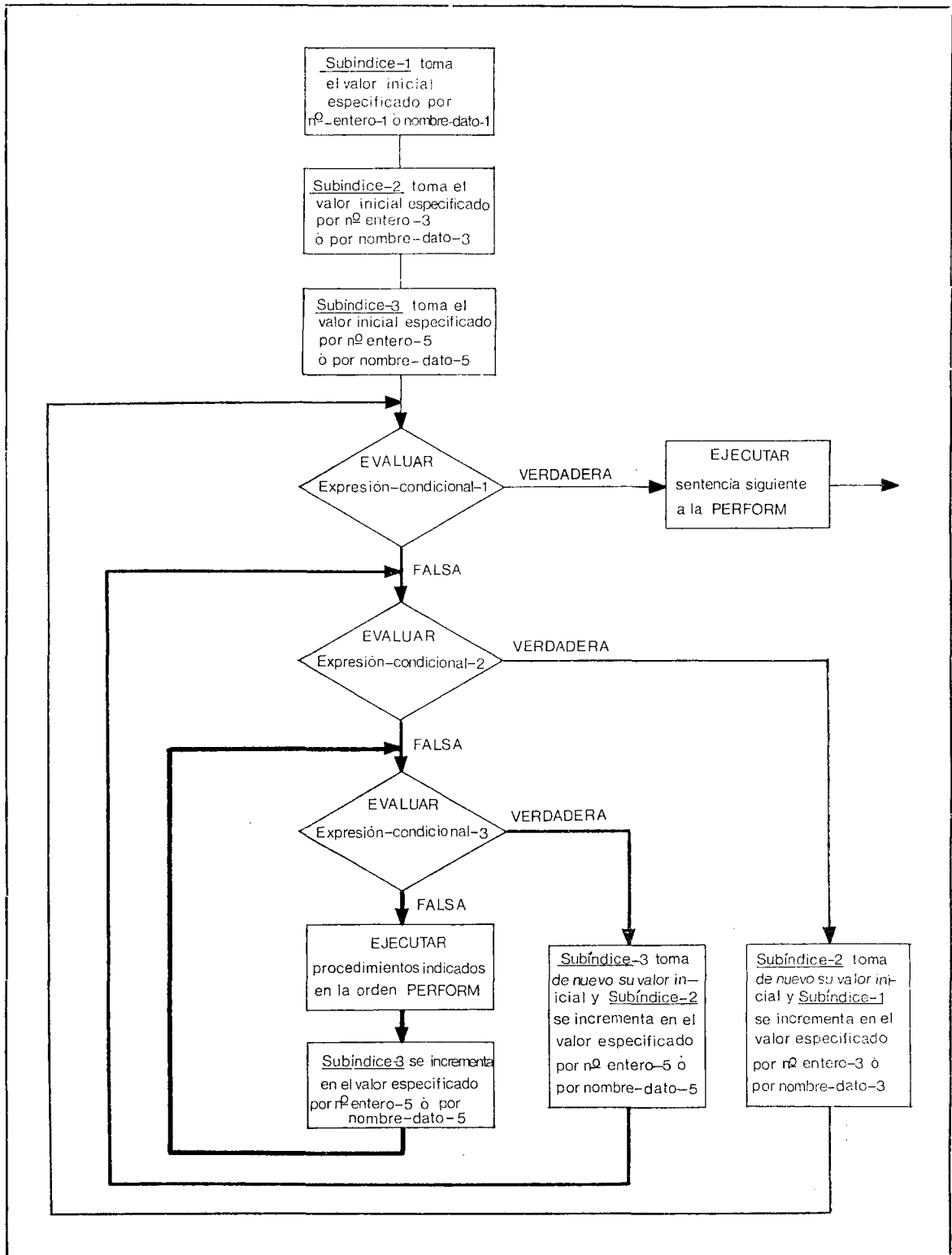
**Con la opción AFTER (2 veces  $\times$  3 subíndices)**

PERFORM nombre-procedimiento-1 [THRU nombre-procedimiento-2] VARYING subíndice-1

FROM { núm.-entero-1 } { nombre-dato-1 } BY { núm.-entero-2 } { nombre-dato-2 } UNTIL expresión-condicional-1 AFTER subíndice-2

FROM { núm.-entero-3 } { nombre-dato-3 } BY { núm.-entero-4 } { nombre-dato-4 } UNTIL expresión-condicional-2 AFTER subíndice-3

FROM { núm.-entero-5 } { nombre-dato-5 } BY { núm.-entero-6 } { nombre-dato-6 } UNTIL expresión-condicional-3



**EJEMPLO-6**

```

PERFORM RUTINA VARYING I FROM 1 BY 2 UNTIL I > 6
AFTER J FROM 1 BY 1 UNTIL J = 3
AFTER K FROM 3 BY -1 UNTIL K < 1.
GO TO FIN.

```

RUTINA.

```

COMPUTE X = K + I - J
MOVE X TO ELEMENTO (I, J, K).

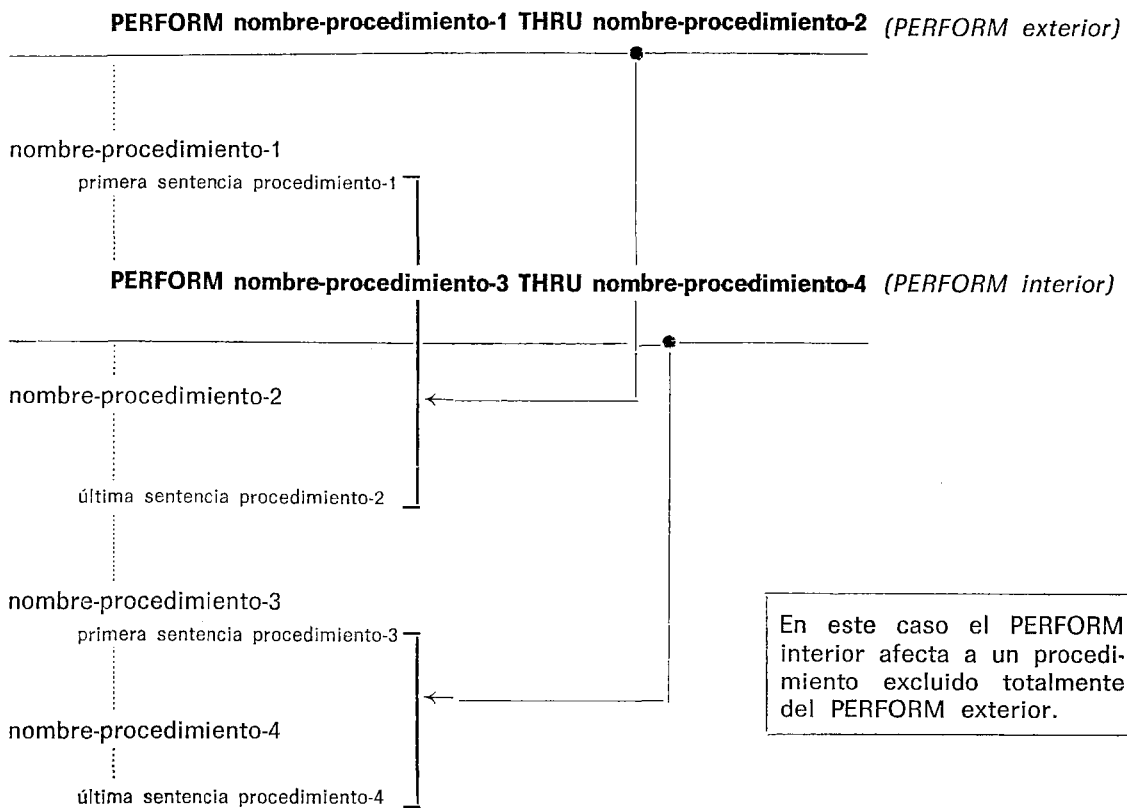
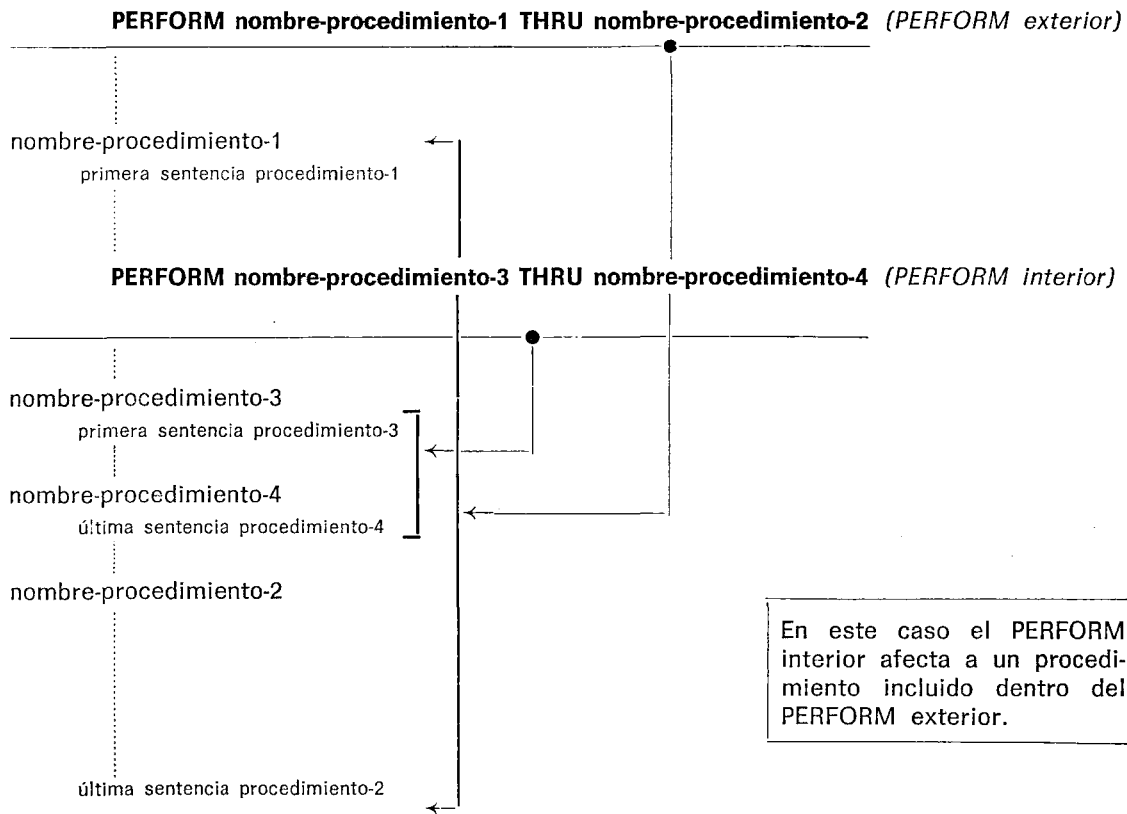
```

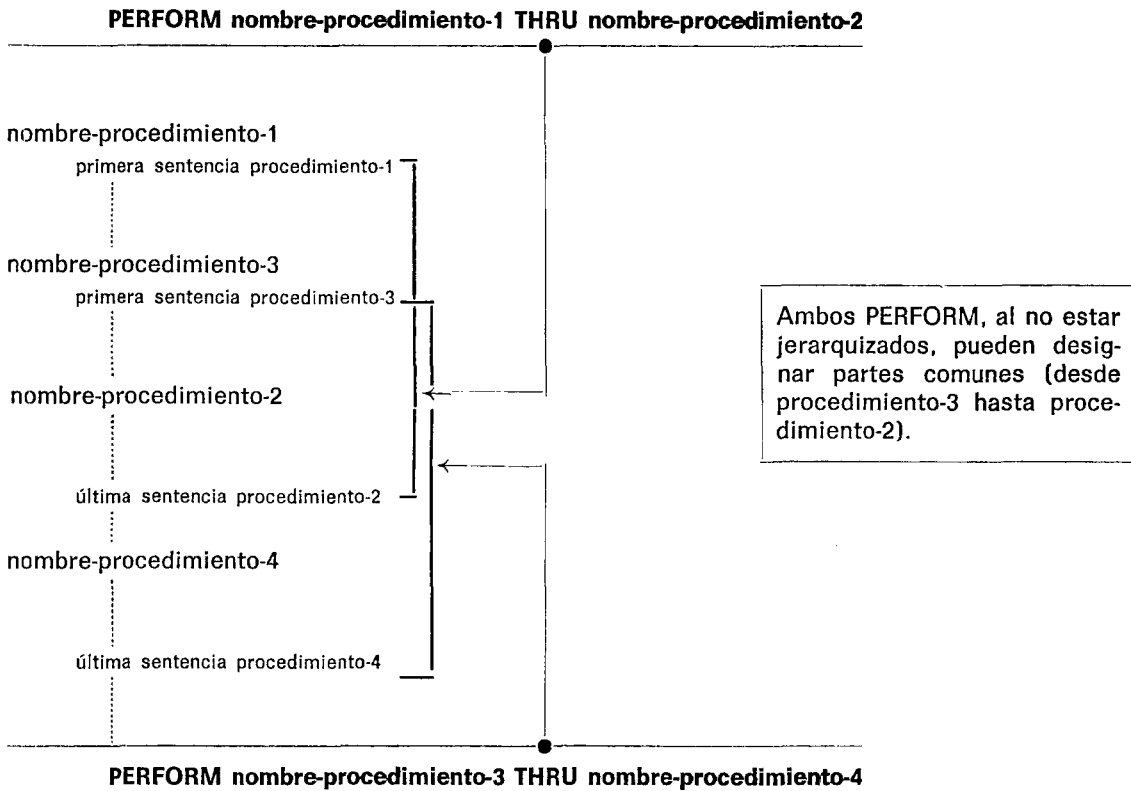
El procedimiento RUTINA se ejecutará 18 veces, que corresponden a los valores sucesivos de I, J y K siguientes:

	Contenido de
1. ^a ) I = 1, J = 1, K = 3 (valores iniciales)	ELEMENTO (1, 1, 3) = 1.
2. ^a ) I = 1, J = 1, K = 2 (se incrementa K en -1)	ELEMENTO (1, 1, 2) = 2.
3. ^a ) I = 1, J = 1, K = 1 (se incrementa K en -1)	ELEMENTO (1, 1, 1) = 1.
4. ^a ) I = 1, J = 2, K = 3 (se incrementa J en 1 y K valor inicial)	ELEMENTO (1, 2, 3) = 2.
5. ^a ) I = 1, J = 2, K = 2 (se incrementa K en -1)	ELEMENTO (1, 2, 2) = 1.
6. ^a ) I = 1, J = 2, K = 1 (se incrementa K en -1)	ELEMENTO (1, 1, 1) = 0.
7. ^a ) I = 3, J = 1, K = 3 (se incrementa I en 2 y J y K valores iniciales)	ELEMENTO (3, 1, 3) = 5.
8. ^a ) I = 3, J = 1, K = 2 (se incrementa K en -1)	ELEMENTO (3, 1, 2) = 4.
9. ^a ) I = 3, J = 1, K = 1 (se incrementa K en -1)	ELEMENTO (3, 1, 1) = 3.
10. ^a ) I = 3, J = 2, K = 3 (se incrementa J en 1 y K valor inicial)	ELEMENTO (3, 2, 3) = 2.
11. ^a ) I = 3, J = 2, K = 2 (se incrementa K en -1)	ELEMENTO (3, 2, 2) = 2.
12. ^a ) I = 3, J = 2, K = 1 (se incrementa K en -1)	ELEMENTO (3, 2, 1) = 2.
13. ^a ) I = 5, J = 1, K = 3 (se incrementa I en 2 y J y K valores iniciales)	ELEMENTO (5, 1, 3) = 7.
14. ^a ) I = 5, J = 1, K = 2 (se incrementa K en -1)	ELEMENTO (5, 1, 2) = 6.
15. ^a ) I = 5, J = 1, K = 1 (se incrementa K en -1)	ELEMENTO (5, 1, 1) = 5.
16. ^a ) I = 5, J = 2, K = 3 (se incrementa J en 1 y K valor inicial)	ELEMENTO (5, 2, 3) = 6.
17. ^a ) I = 5, J = 2, K = 2 (se incrementa K en -1)	ELEMENTO (5, 2, 2) = 5.
18. ^a ) I = 5, J = 2, K = 1 (se incrementa K en -1)	ELEMENTO (5, 2, 1) = 4.

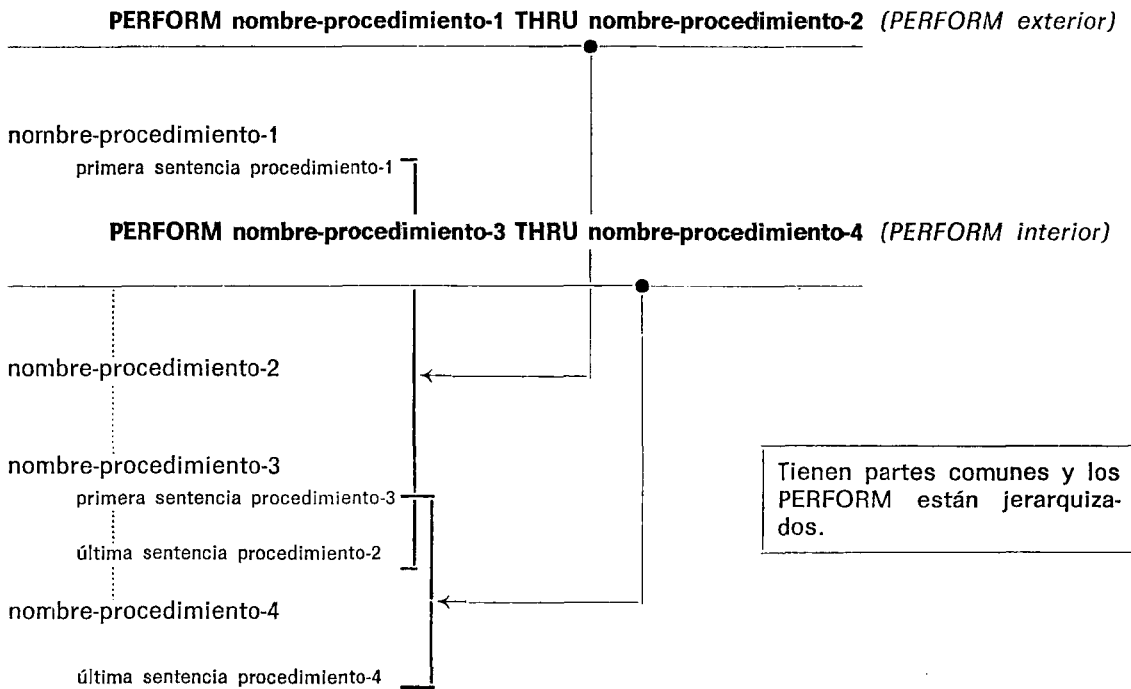








En cambio, sería incorrecto el siguiente supuesto:



## VI-5.4. VERBOS COMPLEMENTARIOS

### VI-5.4.1. Verbo EXIT (1)

**OBJETO**

Este verbo se utiliza como complemento del verbo PERFORM cuando en los procedimientos abarcados por dicha orden PERFORM existe una o varias bifurcaciones.

**FORMATO**

**EXIT**

**PARTICULARIDADES DE SU USO**

La utilización de este verbo obliga a que no exista ningún otro verbo en el párrafo que lo contenga, por consiguiente el verbo EXIT forma, por sí solo, un párrafo identificado por un determinado nombre de párrafo.

Cuando cualquier procedimiento deba ser ejecutado por una orden PERFORM y dicho procedimiento no contiene ninguna bifurcación y, por consiguiente, se ejecuta la última sentencia del último párrafo del procedimiento, no es necesario el empleo del verbo EXIT.

**EJEMPLO**

```
PERFORM RUTINA-B THRU FINAL.  
RUTINA - A.  
  MOVE PTS TO TOTAL.  
  SUBTRACT INTERESES FROM TOTAL GO TO CALCULO-FINAL.  
RUTINA - B.  
  IF PTS < 100 MOVE 100 TOTAL GO TO FINAL.  
RUTINA - C.  
  IF PTS > 200 MOVE 20 TO INTERESES GO TO FINAL.  
  COMPUTE PTS = PTS * .98.  
FINAL.  
  EXIT.  
CALCULO-FINAL.  
  COMPUTE PTS = PTS — INTERES
```

En el ejemplo expuesto, si no existiese la orden complementaria EXIT identificada por el párrafo FINAL, la orden PERFORM que debería haber sido entonces escrita como PERFORM RUTINA-B THRU RUTINA-C, provocaría la ejecución de las sentencias abarcadas por ambos párrafos, es decir, desde IF PTS < 100 ... hasta COMPUTE PTS = PTS * .98.

En el caso de que el valor del campo PTS fuese menor de 100 o mayor de 200, no se ejecutaría la sentencia COMPUTE PTS = PTS * 0.98, y por consiguiente no se realizaría correctamente el retorno a la siguiente sentencia a la PERFORM (MOVE PTS TO TOTAL).

(1) N. A.: Este verbo puede también ser considerado como perteneciente al grupo de verbos del compilador, o como un falso verbo, ya que no origina codificación alguna en el programa-objeto. Por este hecho, algunos manuales tienen esta consideración. Sin embargo, como su función básica es la de servir de complemento a un verbo de control de secuencia, nos ha parecido más lógico su inclusión en este apartado.

Como tanto RUTINA-B como RUTINA-C (abarcadas por PERFORM) contiene órdenes de bifurcación incondicional (GO TO FINAL), es necesario utilizar el verbo EXIT. De esta forma la última sentencia del procedimiento abarcado es precisamente la sentencia EXIT y ya no COMPUTE PTS ... En este caso todas las bifurcaciones se dirigen al párrafo FINAL, que contiene precisamente la última sentencia del procedimiento.

#### VI-5.4.2. Verbo STOP

**OBJETO**

Este verbo se utiliza para interrumpir temporal o definitivamente la ejecución de un programa.

**FORMATO**

**STOP** { literal }  
          { RUN }

**OPCIONES**

Cuando se utiliza en el formato la opción literal, se produce una parada temporal y se emite a través de la consola el valor que se haya establecido para literal. Para proseguir la ejecución de programa el operador debe reanudar el proceso.

Cuando se utiliza la opción RUN se interrumpe definitivamente el proceso de forma automática, quedando liberado el sistema del programa en tratamiento y dispuesto para ejecutar un nuevo programa. Por consiguiente, la orden STOP RUN es la última orden que se ejecuta en un programa, aunque no supone esto que deba ser la instrucción colocada en el último lugar del programa fuente, ya que puede llegarse a ella, por ejemplo, a través de una orden de bifurcación.

VI - 6

VERBOS DEL COMPILADOR



## VI-6. VERBOS DEL COMPILADOR

### VI-6.1. GENERALIDADES

Los verbos del compilador, también denominados directrices del programa traductor son órdenes que se incluyen en el programa fuente para ejercer una determinada acción exclusivamente en la fase de traducción sin tener, por consiguiente, efecto en la fase de ejecución.

Como se ha indicado en el apartado anterior el verbo EXIT puede ser incluido también en este tipo de verbos aunque por las razones expuestas su estudio se ha realizado en los verbos complementarios de los verbos de control de secuencia.

### VI-6.2. VERBO ENTER

#### OBJETO

El verbo ENTER, utilizado en conjunción con otras órdenes COBOL, permite la comunicación entre un programa COBOL y uno o más programas COBOL o programas en otro lenguaje.

#### FORMATO

**ENTER** { especificaciones }

#### PARTICULARIDADES DE SU USO

La forma de empleo del verbo ENTER difiere sensiblemente de un compilador a otro. Por esta causa, se han incluido en el apéndice A los diversos procedimientos de enlace entre programas. El uso de este verbo implica la obligación de utilizar la LINKAGE SECTION o COMMON-STORAGE SECTION (ver capítulo V-6).

### VI-6.3. VERBO NOTE

#### OBJETO

Este verbo permite incluir notas de observaciones aclarativas, dentro del texto, del programa fuente.

#### FORMATO

**NOTE** (cualquier comentario)

#### PARTICULARIDADES DE SU USO

El verbo NOTE debe ser, siempre que se utilice, el primero de la sentencia en la que esté incluido. Por consiguiente pueden ocurrir dos circunstancias:

Que la sentencia en que esté incluido el verbo NOTE sea la primera de un párrafo. En este caso el programa traductor considera la totalidad del párrafo como una nota o comentario.

Que la sentencia en la que esté incluido el verbo NOTE no sea la primera del párrafo. En este caso los comentarios se consideran terminados en el primer punto seguido de un espacio que aparezca en la sentencia.

**EJEMPLOS**

En una parte de un programa se desea efectuar una observación sobre determinados valores de un campo de datos, antes de efectuar su tratamiento.

.....  
.....  
.....

VALORES.

**NOTE LOS VALORES DEL CAMPO MATERIAL TIENEN EL SIGUIENTE SIGNIFICADO.**

**20 ≤ MATERIAL < 30 = HERRAMIENTAS.**

**30 ≤ MATERIAL < 40 = TORNILLERIA.**

**40 ≤ MATERIAL < 50 = MAQUINAS.**

CALCULOS.

IF MATERIAL < 20 OR MATERIAL > 50 NEXT SENTENCE  
ELSE GO TO CALCULO-PRECIO. **NOTE LOS VALORES DE MATERIAL INFERIORES A 20 O SUPERIORES A 50 NO SE TRATAN EN CALCULO-PRECIO.**

IF PRECIO > 40      ADD 20 TO PRECIO.

Con el primer verbo NOTE todo el texto incluido en el párrafo VALORES se lista en el programa fuente en la fase de traducción existan o no otros verbos dentro del párrafo. Con el segundo verbo NOTE el programa traductor considera como primera sentencia la que se inicia con la palabra IF hasta el verbo NOTE y como segunda sentencia la que se inicia con la otra palabra IF, que es la primera que aparece después del punto y espacio que siguen a la palabra CALCULO-PRECIO.



VI - 7

VERBO DE CONTROL DE  
ENTRADA-SALIDA



## VI-7. VERBO DE CONTROL DE ENTRADA-SALIDA

### VI-7.1. GENERALIDADES

En la utilización de los verbos de entrada-salida se exponía que existe una comprobación de etiquetas previa al proceso de lectura de un fichero o una generación de etiquetas en la grabación o escritura de ficheros. El control de estas etiquetas se efectúa por los verbos de entrada-salida salvo cuando la etiqueta está especificada por el usuario. La indicación de este tipo de etiquetas se efectúa en la cláusula LABEL RECORD mediante la correspondiente opción.

Por otra parte, puede suceder que durante el desarrollo de un programa, se detecte un error de paridad sobre una banda magnética, un error de direccionamiento en los ficheros al azar, etc.

Todas las circunstancias anteriores provocan, si se presenta el error y si el programador no ha previsto ninguna acción en el programa que la ejecución del mismo, se cancele.

El verbo USE permite controlar estas situaciones provocando la ejecución de un determinado procedimiento.

### VI-7.2. VERBO USE

#### OBJETO

El verbo USE permite ejecutar determinados procedimientos cuando se presenta un error en una operación de entrada-salida o cuando se especifican etiquetas de usuario en la descripción de un fichero.

#### FORMATOS

Existen dos posibles formatos:

*Para control de etiquetas de usuario*

```
USE FOR { CREATING } [ BEGINNING ] LABEL ON  
          { CHECKING } [ ENDING ]  
{ OUTPUT } nombre-fichero-1 nombre-fichero-2 ...  
{ INPUT }
```

*Para control de errores de entrada-salida*

```
USE AFTER STANDARD ERROR PROCEDURE ON  
nombre-fichero
```

#### PARTICULARIDADES DE SU USO

Para cualquiera de los formatos especificados en el verbo USE, en caso de detección de un error, ejecuta un determinado procedimiento.

Este procedimiento debe venir especificado obligatoriamente dentro de una sección que se denomina declarativa. Como todas las secciones, debe especificarse con un nombre de procedimiento y la palabra SECTION. Las secciones declarativas deben agruparse juntas al comienzo de la PROCEDURE DIVISION e ir precedidas de la palabra clave DECLARATIVES, escrita en el margen A y seguida de las palabras claves END DECLARATIVES, también escritas a partir del margen A.

En el primer formato se utiliza la palabra CREATING conjuntamente con OUTPUT cuando se desee proporcionar un proceso de grabación de etiquetas especificadas por el usuario y la palabra CHECKING conjuntamente con INPUT cuando se desee proporcionar un proceso de lectura de etiquetas de usuario.

Las opciones BEGINNING o ENDING hacen referencia a control de etiquetas de cabecera o cola, respectivamente. La omisión de estas opciones obliga al control de ambas.

En la compilación del verbo USE con el primer formato, el traductor coloca las instrucciones correspondientes al final de la última declaración de la SECCION. Por consiguiente, todas las bifurcaciones lógicas del programa dentro de esta sección, deben dirigirse al final de dicha sección, salvo en el caso de establecer una declaración GO TO MORE-LABELS.

En este caso, la acción transcurre de la forma siguiente:

Para ficheros de entrada se lee la etiqueta de cabecera o cola especificada por el usuario y se retorna a la sección para efectuar una nueva comprobación de etiquetas.

Para ficheros de salida se escribe la etiqueta de cola o cabecera especificada por el usuario y se retorna a la sección para la creación de nuevas etiquetas.

Las etiquetas especificadas por el usuario deben estar contenidas en el área de entrada o en el área de salida. La descripción de estas etiquetas debe realizarse como un campo de datos, elementales o compuestos a la Sección de Unión de la División de Datos y especificarse dicho nombre en la cláusula LABEL RECORDS de la inscripción FD referida al fichero que deba contener o que contenga dicha etiqueta.

El formato segundo obliga a que en las secciones declarativas no se pueda hacer referencia mediante los verbos OPEN, RECORD, WRITE o REWRITE a los ficheros asociados con el verbo USE.

El traductor COBOL proporciona un registro especial denominado ERROR-CODE generado por el sistema de control de entrada-salida para almacenar el tipo de error que se produce y que puede ser consultado, por su nombre, por el programador dentro de las sentencias de las secciones declarativas.

<b>EJEMPLO</b>
----------------

PROCEDURE DIVISION.

DECLARATIVES.

CONTROL SECTION.

USE AFTER ERROR FICH-ENT.

ANULA-ERROR.

    DISPLAY 'REGISTRO ERRONEO' REG-ENT UPON CONSOLE.

END DECLARATIVES.

Con esta sección declarativa se genera sobre la consola un mensaje de error, con el contenido siguiente REGISTRO ERRONEO (contenido del registro) cada vez que en la lectura del fichero denominado FICH-ENT se detecte una anomalía con arreglo a la definición establecida para el mismo. El registro se omite y prosigue la ejecución del programa.



```

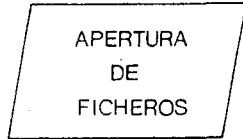
00253 020010 PROCEDURE DIVISION.
00254 020012 DECLARATIVES.
00255 020014 VERIFI SECTION.
00256 020016 USE AFTER ERROR PROCEDURE DISCO.
00257 020017 ABC.
00258 020018 DISPLAY '**EJEMPLO ** BLOQUE ERRONEO ' DIS.
00259 020019 END DECLARATIVES.
-----
00260 020020 000-APERTURA.
00261 020030 OPEN INPUT FICHAS
00262 020040 I-O DISCO
00263 020050 OUTPUT CINTA IMPRESORA.
00264 020060
00265 020070
00266 020080
00267 020090 NOTE *****
00268 020100 APERTURA DE FICHEROS Y TRATAMIENTO FICHA DE
00269 020110 CANCELANDO EL TRABAJO SI ESTA ES ERRONEA
00270 020120 *****
-----
00271
00272
00273
00274 020130 010-FICHA.
00275 020140 ACCEPT FICHA-FECHA.
00276 020150 IF MES-FECHA GREATER 0 AND MES-FECHA LESS 13
00277 020160 GO TO 030-LEERFICHA.
00278 020170 DISPLAY
00279 020180 '**EJEMPLO ** FICHA DE FECHA ERRONEA ' FICHA-FEC
00280 020190 020-CANCELAR.
00281 020200 STOP
00282 020210 '**EJEMPLO ** OPERADOR CANCELE TRABAJO'
00283 020220 GO TO 020-CANCELAR.
00284
-----
00285
00286
00287 021010 NOTE *****
00288 021020 LECTURA FICHERO DE FICHAS, BUSCANDO EN TABL
00289 021030 QUE SE TRATA, TRATAMIENTO DE DEPURACION D
00290 021040 SI PROCEDE TRATAMIENTO DE DISCO
00291 021050 *****
00292 022010 030-LEERFICHA.
00293 022020 READ FICHAS
00294 022030 INTO WORKAFICHA
00295 022040 AT END MOVE HIGH-VALUE TO WORKAFI
00296 022050 GO TO '9999'-FINAL.
00297 022083 MOVE 1 TO X1.
-----
00298 022085 040-BUSCA. GO TO 045-C.
-----
00299 022086 045-C.
00300 022090 IF MES-FECHA = MESTA (X1) MOVE NOMTA (X1) TO
00301 022100 GO TO 050-TRATAMIENTO.
00302 022110 ADD 1 TO X1.
00303 022120 IF X1 GREATER 11
00304 022130 NEXT SENTENCE ELSE
00305 022140 GO TO 045-C.
00306 022150 MOVE NOMTA (X1) TO MESI.
00307 022160 ALTER
00308 022170 040-BUSCA TO PROCEED TO 050-TRATAMIENTO.
00309 022190 050-TRATAMIENTO.
00310 022200 ADD 1 TO FICHAS-LEIDA.
00311 022210 PERFORM 100-RUTINATRATA THRU 500-FINRUTINA.
00312 022220 GO TO 030-LEERFICHA.
00313
00314
00315

```

00025300  
00025400  
00025500  
00025600  
00025700  
00025800  
00025900

Esta sección de DECLARATIVOS contiene un único procedimiento englobado dentro del párrafo ABC; se utiliza para controlar el caso de detectarse un bloque de longitud distinta a la especificada en la inscripción FD del fichero DISCO (por error de dispack o cualquier otra eventualidad). Se emite el mensaje *****EJEMPLO***BLOQUE ERRONEO** y el contenido del bloque erróneo.

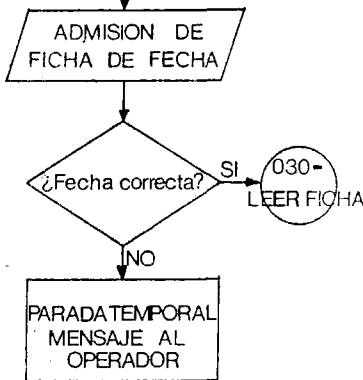
00026000  
00026100  
00026200  
00026300  
00026400  
00026500  
00026600



Con una única orden OPEN se efectúa la apertura de todos los ficheros: uno de entrada por intermedio de INPUT, dos de salida con OUTPUT y uno de entrada-salida con I-O (se lee y se graba en él).

*****00026700  
FECHA 00026800  
*****00026900  
*****00027000

00027100  
00027200  
00027300  
00027400  
00027500  
00027600  
00027700  
00027800  
00027900  
00028000  
00028100  
00028200  
00028300  
00028400



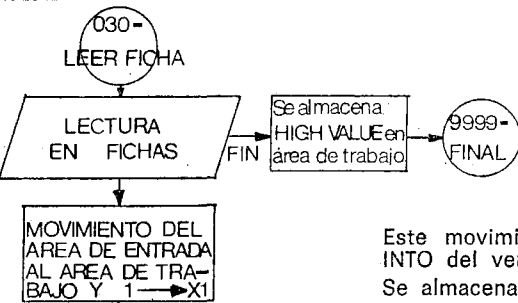
La orden ACCEPT, sin opción FROM, admite una ficha de control desde la lectora de fichas. Debe contener en las columnas 3 y 4 el valor del mes de los datos que debe ser impreso en la línea de censura del fichero IMPRESORA.

Se efectúa el control de validez de la ficha de control:  $0 < \text{valor mes} < 13$ .

En caso de ser correcto se bifurca el párrafo 030-LEER FICHA.

En caso de ser errónea se provoca un «loop» con mensajes intermitentes al operador: FICHA DE FECHA ERRONEA XX ** EJEMPLO ** OPERADOR CANCELE TRABAJO.

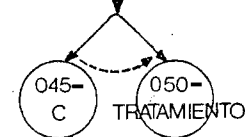
00028500  
00028600  
*****00028700  
EL MES 00028800  
FICHA Y 00028900  
*****00029000



Este movimiento se realiza con la opción INTO del verbo READ.

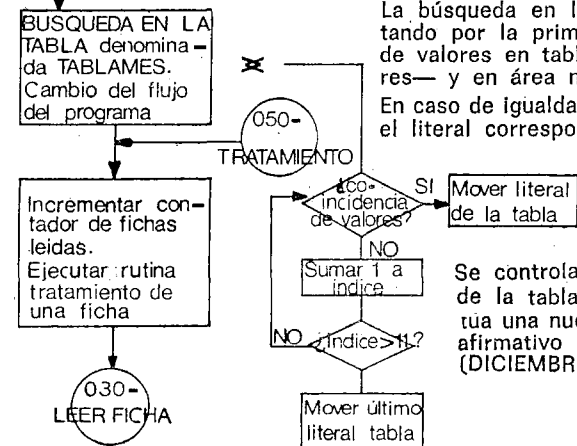
Se almacena el valor 1 en el campo X1.

00029100  
00029200  
00029300  
00029400  
00029500  
00029600  
00029700



Se incluye una bifurcación incondicional que va a ser controlada por un verbo ALTER para alterar el camino del flujo del programa. La primera vez se bifurca al procedimiento 045-C. La segunda y siguientes al procedimiento 050-TRATAMIENTO.

00029800  
00029900  
00030000  
00030100  
00030200  
00030300  
00030400  
00030500  
00030600  
00030700  
00030800  
00030900  
00031000  
00031100  
00031200  
00031300  
00031400  
00031500



La búsqueda en la tabla se realiza preguntando por la primera posición (coincidencia de valores en tabla —dos primeros caracteres— y en área mes-fecha).

En caso de igualdad se mueve al campo MESI el literal correspondiente de la tabla.

En caso de desigualdad se mueve una posición en la tabla (incrementando en 1 el índice).

Se controla si se ha llegado al final de la tabla. En caso negativo se efectúa una nueva comparación. En caso afirmativo se mueve el último literal (DICIEMBRE).



00316

```
00317 024010 100-RUTINATRATA.
00318 024020 EXAMINE TIPO-FICHA REPLACING LEADING ' ' BY '1'.
00319 024030 IF TIPO-FICHA LESS ZERO OR TIPO-FICHA GREATER 3
00320 024040 GO TO 110-LISTA.
00321 024050 TRANSFORM CLAVEFICHA FROM 'OAKR' TO '1237'
00322 024060 IF CLAVEFICHA NUMERIC AND COEFICI-FICHA NUMERIC
00323 024070 GO TO 130-TRATA1
00324 024080 140-TRATA2
00325 024090 180-TRATA3
00326 024100 DEPENDING ON TIPO.
00327
00328
00329 024110 110-LISTA.
00330 024120 PERFORM 1000-LISTA THRU 1500-FINLIS.
00331 024130 GO TO 500-FINRUTINA.
00332 024150 NOTE *****
00333 024160 ACABAMOS DE VERIFICAR EL TIPO Y CLAVE DE FICHA
00334 024170 SI ES ERRONEA,
00335 024180 *****
00336
00337
00338
00339 025010 130-TRATA1.
00340 025020 ADD C-TABLA (1CLA) TO TOTAL-COE.
00341 025025 ADD C-TABLA (2CLA) TO TOTAL-COE.
00342 025030 ADD C-TABLA (3CLA) TO TOTAL-COE.
00343 025035 ADD C-TABLA (4CLA) TO TOTAL-COE.
00344 025060 138-CONTINUA.
00345 025070 DIVIDE TOTAL-COE INTO INFLACION GIVING INDICE-APLI
00346 025080 ON SIZE ERROR
00347 025090 MOVE ZERO TO INDICE-APLICA
00348 025100 IF CLAVEFICHA LESS 0008
00349 025110 COMPUTE PTSCOEFI ROUNDED = INDICE-APLI
00350 025120 (TOTAL-COE * 9)
00351 025130 GO TO 190-TRATADISCO.
00352 025140 IF CLAVEFICHA GREATER 0008
00353 025150 SUBTRACT 1 FROM INDICE-APLICA GIVING P
00354 025160 GO TO 190-TRATADISCO.
00355 025170 MULTIPLY 3 BY INDICE-APLICA GIVING I
00356 025180 GO TO 190-TRATADISCO.
00357 025190 NOTE *****
00358 025200 TRATAMIENTO DE LAS FICHAS DE TIPO 1
00359 025300 *****
00360
00361
00362
00363 026010 140-TRATA2.
00364 026020 PERFORM 130-TRATA1 3 TIMES.
00365 026030 COMPUTE INDICE-APLICA =
00366 026040 TOTAL-COE N INFLACION
00367 026050 ON SIZE ERROR MOVE ZERO TO INDICE-APLICA.
00368 026080 160-CONTINUA.
00369 026090 IF INDICE-APLICA LESS 5
00370 026100 PERFORM 140-TRATA2
00371 026110 UNTIL INDICE-APLICA = 5 OR INDICE-APLICA GRE
00372 026130 COMPUTE PTSCOEFI = ((TOTAL-COE * INDICE-APLICA) *
00373 026180 GO TO 190-TRATADISCO.
00374 026190 170-SUMAR.
00375 026300 ADD C-TABLA (X1) TO PTSCOEFI.
00376 026400
00377 026500 NOTE *****
00378 026600 TRATAMIENTO Y CALCULO DE LA FICHA 2
```

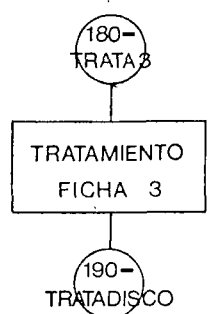
00031600	Los párrafos siguientes, hasta el de nombre de párrafo 500-FINRUTINA, efectúan el tratamiento de cada una de las fichas del fichero FICHAS.
00031700	<p>Las fichas, deben de tener en el campo TIPO-FICHA, los valores 1, 2 y 3 para que sean consideradas correctas. En el caso de no tener ningún valor se transforma en valor 1.</p> <p>La orden EXAMINE sustituye al valor espacio por el valor 1 en el campo TIPO-FICHA. Se controla el valor de TIPO-FICHA. Si es erróneo (∅ &gt; TIPO-FICHA &gt; 3) se bifurca el párrafo 110-LISTA. Si es correcto se transforman los caracteres del campo CLAVEFICHA. El ∅ pasa a ser 1; la A pasa a ser 2; la K pasa a ser 3 y la R pasa a ser 7. Los demás caracteres no cambian. Por último, si los campos CLAVEFICHA y COEFICI-FICHA son numéricos se bifurca al tratamiento individual de cada tipo de ficha (1 al tratamiento 130-TRATA1; 2 al tratamiento 140-TRATA2, y 3 al tratamiento 180-TRATA3).</p>
00031800	
00031900	
00032000	
00032100	
00032300	
00032400	
00032500	
00032600	
00032700	
00032800	
00032900	<p>A este párrafo se llega cuando: TIPO-FICHA es erróneo (línea 024040), o CLAVE-FICHA y/o COEFICI-FICHA no son numéricos (línea 024120).</p> <p>En estos casos se ejecutan los procedimientos de impresión de un registro en el fichero IMPRESORA y se sale de la rutina de tratamiento de fichas.</p>
00033000	
00033100	
00033200	
*****00033300	
A, LISTAND00033400	
00033500	
*****00033600	
00033700	
00033800	
00033900	<p>El tratamiento de las fichas con valor 1 en el campo TIPO-FICHA consiste en:</p> <ol style="list-style-type: none"> <li>Sumar los valores de la tabla denominada T-COEFICIENTE correspondiente a los caracteres del campo CLAVEFICHA (si el carácter es un 1 se suma el primer valor de la tabla, si el carácter es un 2 el 2.º, etc.). La suma de los valores se almacena en el campo TOTAL-COE.</li> <li>Calcular su índice de aplicación $\text{INDICE-APLICACION} = \frac{\text{INFLACION}}{\text{TOTAL-COE}}$ <p>La opción ON SIZE ERROR especifica que cuando TOTAL-COE = ∅ (que daría un valor infinito a INDICE-APLICACION) se almacena el valor ∅ en INDICE-APLICACION.</p> </li> <li>Calcular su pts. de coeficiente. Este cálculo responde a las siguientes fórmulas: $\text{PTS COEFI} = \text{índice-aplicación} + 9 \times \text{total-coe}$ <p>(cuando CLAVE FICHA &lt; 8)</p> $\text{PTS COEFI} = \text{índice-aplicación} - 1$ <p>(cuando CLAVE FICHA &gt; 8)</p> $\text{PTS COEFI} = \text{índice-aplicación} \times 3$ <p>(cuando CLAVE FICHA = 8)</p> </li> <li>Consulta al fichero DISCO.</li> </ol>
CA 00034000	
00034100	
00034110	
00034200	
00034210	
00034500	
CA 00034600	
00034700	
00034800	
00034900	
CA + 00035000	
00035100	
00035200	
00035300	
FSCOEFI 00035400	
00035500	
PTSCOEFI 00035600	
00035700	
*****00035800	
00035900	
*****00036000	
00036100	
00036200	
00036300	<p>El tratamiento de las fichas con valor 2 en el campo TIPO-FICHA consiste en:</p> <ol style="list-style-type: none"> <li>Ejecutar el cálculo de TOTAL-COE de la misma forma que en el caso de TIPO-FICHA = 1. Se realiza con la orden PERFORM ... 3 TIMES.</li> <li>Cálculo de índice-aplicación. $\text{INDICE-APLICACION} = \frac{\text{TOTAL-COE}}{\text{INFLACION}}$ </li> <li>Se controla el valor de índice de aplicación. Si es menor de 5 se efectúan de nuevo los cálculos de los pasos 1.º y 2.º hasta que se consigue un valor de índice-aplicación ≥ 5.</li> <li>Calcular su pts de coeficiente. Este cálculo se realiza con la siguiente fórmula: $\text{PTS-COEFI} = (\text{total-coeficiente} \times \text{índice aplicación}).$ </li> <li>Consulta al fichero DISCO.</li> </ol>
00036400	
00036500	
00036600	
00036700	
00036800	
00037100	
00037200	
00037300	
ITER 5. 00037400	
* 2). 00037500	
00037900	
00038000	
00038200	
00038300	
*****00038400	
00038500	

```

00379 026700 *****
00380
00381
00382
00383 027010 180-TRATA3.
00384 027020 MOVE 1 TO X1.
00385 027030 PERFORM 170-SUMAR VARYING X1 FROM 1 BY 3 UNTIL
00386 027050 NOTE *****
00387 027060 TRATAMIENTO Y CALCULO DE LA FICHA 3
00388 027070 *****
00389
00390
00391
00392 027090 190-TRATADISCO.
00393 027100 MOVE CLAVEFICHA TO CLAVE-ID.
00394 027110 READ DISCO
00395 027120 INVALID KEY GO TO 200-ALTA.
00396 027130 MOVE PTSCOEFI TO COEFICIENTE-D.
00397 027140 REWRITE DIS
00398 027142 MOVE PARADISCO TO CIN.
00399 027144 WRITE CIN.
00400 027146 ADD 1 TO RGTOS-ESCRIT.
00401 027150 MOVE ZERO TO TOTAL-COE INDICE-APLI
00402 027160 GO TO 500-FINRUTINA.
00403
00404
00405 027170
00406 027173 200-ALTA.
00407 027180 MOVE PTSCOEFI TO COEFICIENTE-D.
00408 027190 MOVE CLAVEFICHA TO CLAVE-DISCO.
00409 027200 MOVE TIPO-FICHA TO TIPO-DISCO.
00410 027300 MOVE CORRESPONDING PARADISCO TO DIS.
00411 028010 WRITE DIS
00412 028020 INVALID KEY
00413 028030 GO TO 300-ERROR.
00414 028040 WRITE CIN FROM PARADISCO.
00415 028050 ADD 1 TO RGTOS-ESCRIT MOVE ZERO TO TOTAL-COE
00416 028060 GO TO 500-FINRUTINA.
00417 028070 300-ERROR.
00418 028080 DISPLAY *** EJEMPLO** CLAVE DUPLICADA O ERRONE
00419 028090 UPON CONSOLE.
00420 028130 500-NOTA.
00421 028140 NOTE *****
00422 028150 ACABAMOS DE ACTUALIZAR DISCO Y GRABAR EN CI
00423 028160 LAS FICHAS QUE NO ESTAN EN DISCO
00424 028170 *****
00425 028180 500-FINRUTINA.
00426 028190 EXIT.
00427
00428
00429
00430
00431
00432
00433 029010 NOTE *****
00434 029020 ESCRIBIR LINEA DE DETALLE Y CABECERAS
00435 029030 *****
00436 030010 1000-LISTA.
00437 030020 IF SW1 NOT = ZERO
00438 030030 GO TO 1020-DETALLE.
00439 030040 MOVE 1 TO SW1.
00440 030050 1010-CABECERA.
00441 030060 SUBTRACT 1 FROM HOJA.

```

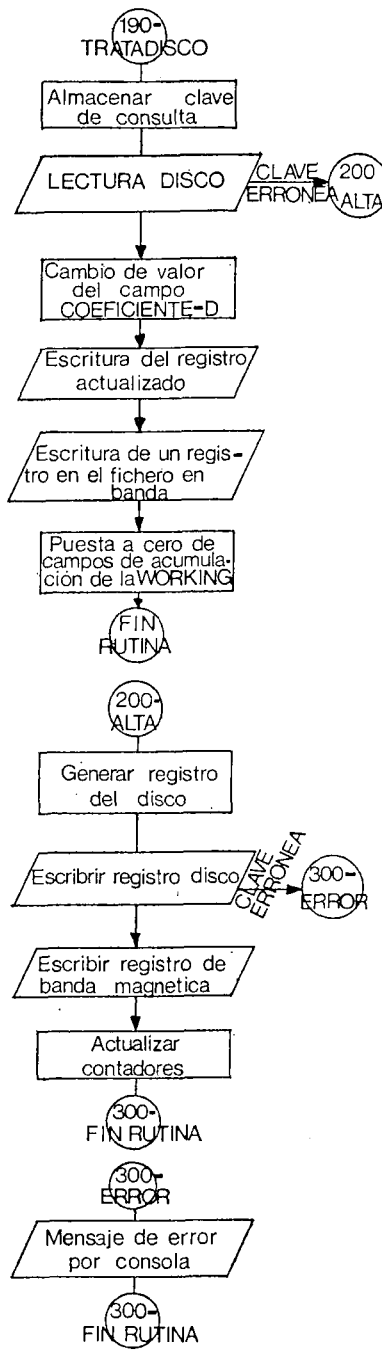
*****.00038600  
 00038700  
 00038800  
 00038900  
 00039000  
 00039100  
 X1 EQUAL 7. 00039200  
 *****00039300  
 00039400  
 *****.00039500  
 00039600  
 00039700



El tratamiento de las fichas con valor 3 en el campo TIPO-FICHA consiste en:

- 1.º) Efectuar el cálculo de PTS COEFI a través de acumular a su valor, los valores de los elementos 1.º, 4.º y 7.º de la TABLA T-COEFICIENTE. Se realiza con el verbo PERFORM.
- 2.º) Consulta al fichero en DISCO por secuencia normal.

00039800  
 00039900  
 00040000  
 00040100  
 00040200  
 00040300  
 00040400  
 00040410  
 00040420  
 00040430  
 CA. 00040500  
 00040600  
 00040700  
 00040800  
 00040900  
 00041000  
 00041100  
 00041200  
 00041300  
 00041400  
 00041500  
 00041600  
 00041700  
 00041800  
 00041900  
 00042000  
 00042100  
 A 00042200  
 00042300  
 00042400  
 *****00042500  
 AITA TODAS 00042600  
 00042700  
 *****.00042800  
 00042900  
 00043000  
 00043100  
 00043200  
 00043300  
 00043400  
 00043500  
 00043600



Si cualquiera de los tres tipos de ficha tratados son correctos, se efectúa una consulta al fichero en disco para actualizar su contenido. La clase de identificación es el contenido del campo CLAVEFICHA.

Si el contenido de dicho campo no coincide con ningún registro del disco se procede a efectuar un alta en el fichero.

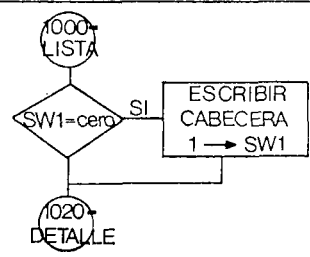
Si existe se sustituye el valor de PTS COEFI calculado por el de COEFICIENTE-D del DISCO y se graba el registro actualizado.

Como las fichas que efectúan consultas al disco son correctas en su contenido se graban en el fichero en banda magnética haciendo previamente un movimiento desde el área de trabajo al área de salida.

Se actualizan los contadores creados en la WORKING y se sale de la rutina para volver a la sentencia siguiente a la que se abandonó la secuencia del programa para ejecutar la rutina de tratamiento de una ficha. En el caso de no encontrarse el registro buscado en el disco se da una alta generando un registro nuevo y moviendo cada uno de los campos de la ficha a los campos del área de entrada-salida del disco (obsérvese que la instrucción MOVE CORRESPONDING solamente mueve el campo CODIGO, que es el que tiene igual nombre en ambos registros). A continuación se escribe un registro en el fichero en banda haciendo, a través de la opción FROM del verbo WRITE el movimiento desde el área de trabajo al área de salida.

Se actualizan los contadores del área de trabajo y se sale de la rutina. En el caso de salida por INVALID KEY se emite un mensaje por consola indicando que la clave es errónea ya que el caso duplicada no debe darse.

*****.00043700  
 00043800  
 *****.00043900  
 00044000  
 00044100  
 00044200  
 00044300  
 00044400  
 00044500



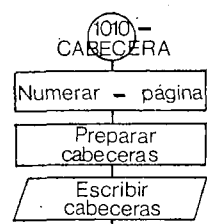
Con este párrafo se controla si se está al comienzo de una página. Si es afirmativo (SW1 = 0) se debe escribir una cabecera de página. En caso contrario se escribe a un registro erróneo del fichero en fichas.

```

00442 030070 MOVE HOJA TO LHOJA.
00443 030080 MOVE LINEA1 TO IMPRE.
00444 030090 WRITE IMPRE AFTER ADVANCING LINNEAL.
00445 030100 WRITE IMPRE FROM LINEA2 AFTER ADVANCING 1.
00446 030110 WRITE IMPRE FROM LINEA3 AFTER ADVANCING 2.
00447 030120 1020-DETALLE.
00448 030122 ADD 2 TO CONTADORLI.
00449 030124 IF CONTADORLI GREATER 56 .
00450 030126 MOVE 2 TO CONTADORLI
00451 030128 GO TO 1010-CABECER
00452 030130 MOVE CLAVEFICHA TO CLAVEI.
00453 030140 MOVE COEFICI-FICHA TO COEFI.
00454 030150 MOVE 'E' TO CODIGI.
00455 030160 MOVE TIPO-FICHA TO TIPOI.
00456 030170 WRITE IMPRE FROM LINEA4 AFTER ADVANCING 2.
00457 030180
00458 030190 1500-FINLIS.
00459 030200 EXIT.
00460
00461
00462
00463
00464
00465
00466
00467
00468 031010 9999-FINAL.
00469 031020 DISPLAY '*** EJEMPLO** FICHAS LEIDAS ' FICH
00470 031030 UPON CONSOLE.
00471 031040 DISPLAY '*** EJEMPLO** REGISTROS ESCRITOS CI
00472 031050 RGTDS-ESCRIT.
00473 031070 CLOSE FICHAS
00474 031080 DISCO
00475 031090 IMPRESORA
00476 031100 CINTA.
00477 031120 GOBACK.

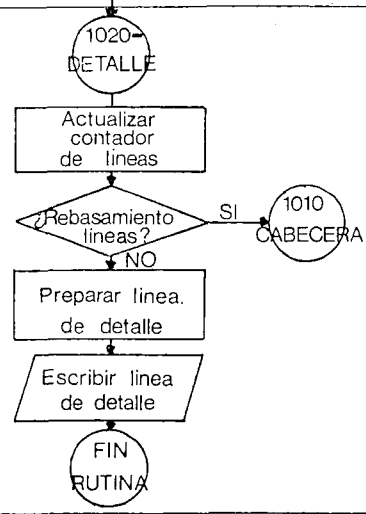
```

00044600  
 00044700  
 00044800  
 00044900  
 00045000



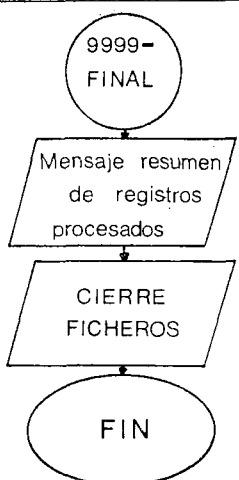
Con este párrafo se escribe una línea de cabecera. La numeración de páginas se realiza creando un valor negativo en el contador de páginas (con el fin de que la PICTURE _ _ _ edite el signo - que se utiliza para enmarcar entre guiones el valor). Se generan los valores variables de las líneas de cabecera y se editan las 3 líneas de cabecera.

00045100  
 00045110  
 00045120  
 00045130  
 00045140  
 00045200  
 00045300  
 00045400  
 00045500  
 00045600  
 00045700  
 00045800  
 00045900  
 00046000  
 00046100  
 00046200



Se actualiza el contador de líneas impresas. Antes de imprimir una línea de detalle se consulta el valor. Si supera el máximo número admitido, 56, se editan previamente las líneas de cabecera. Se genera el registro de detalle moviendo los campos de la ficha a los campos del área de salida del fichero IMPRESORA.

00046300  
 00046400  
 00046500  
 00046600  
 00046700  
 00046800  
 00046900  
 00047000  
 00047100  
 00047200  
 00047300  
 00047400  
 00047500  
 00047600  
 00047700



FIN DEL FICHERO DE ENTRADA


Esta rutina se ejecuta al detectarse el final del fichero en fichas. Se emiten dos mensajes sobre la consola:  
 ** EJEMPLO ** FICHAS LEIDAS ****  
 ** EJEMPLO ** REGISTROS ESCRITOS CINTA ****  
 Con el contenido de los contadores respectivos se cierran los ficheros que intervienen en el proceso y se detiene el proceso GOBACK X STOP RUN.

LEIDA




# CAPITULO VII

DEPURACION DE UN PROGRAMA  
COBOL









**CONTENIDO**

VII-1. MENSAJES  
DEL COMPILADOR

VII-2. LENGUAJE DEPURADOR





VII - 1

MENSAJES DEL COMPILADOR



## VII-1. MENSAJES DEL COMPILADOR

Los programas compiladores o traductores exploran las sentencias, elemento por elemento, para determinar si las palabras están combinadas de forma significativa. Basándose en los elementos que ya han sido explorados, existen únicamente ciertas palabras o elementos que pueden combinarse correctamente.

Si los elementos previstos no se encuentran, se produce un mensaje de diagnóstico. Algunos de estos errores no pueden descubrirse hasta que se combina la información procedente de varias partes del programa y se denota la discrepancia. Los errores que se descubren de esta forma pueden dar origen a un formato de mensajes ligeramente diferente a los descubiertos cuando todavía está disponible el texto fuente. El mensaje que aparece como único mediante este determinado error puede que no tenga, por ejemplo, la sentencia fuente real que originó el error. Sin embargo, la referencia de posición y secuencia indica el lugar en que se descubrió el error. Los errores que puedan parecer idénticos pueden ser diagnosticados de una forma distinta, dependiendo de dónde los encontró el compilador y de cómo encajan en el contexto de la sintaxis válida. Por ejemplo, un punto que falte al final de una cláusula `WORKING-STORAGE SECTION` se diagnostica específicamente como ausencia de un punto necesario.

En el apéndice E se detallan todos los mensajes que puede emitir uno de los compiladores a los que se ha hecho referencia en el estudio del lenguaje. Ante la imposibilidad material de incluir todos los catálogos de mensajes de diagnósticos de los compiladores existentes en el mercado, hemos seleccionado uno de los que pueden tener mayor utilidad para los alumnos del Instituto de Informática.

La corrección, por parte del programador, de aquellos errores que el compilador no puede automáticamente subsanar, obligará a efectuar una nueva compilación del programa y así sucesivamente hasta que no se produzca ningún diagnóstico o mensaje de error. En estas circunstancias, el paso siguiente consiste en efectuar pruebas con los adecuados juegos de ensayo, para comprobar, ejecutándose el programa, que el programa realiza exactamente las operaciones que se desean.



VII - 2

LENGUAJE DEPURADOR





## VII-2. EL LENGUAJE DEPURADOR, EN FASE DE PRUEBAS

Cuando un programa ha sido totalmente depurado de los posibles errores de sintaxis o morfología, detectados en la etapa de compilación, a través de los mensajes proporcionados por el programa traductor, incorporando las palabras necesarias o modificando las existentes, viene la etapa más difícil de la puesta a punto del programa. Esta es la denominada depuración lógica, o dicho de otra forma, la comprobación de que el programa realiza exactamente lo que se pretende que efectúe.

Normalmente, en la primera o primeras pruebas que se efectúan, se comprueba, desoladoramente, que el programa no realiza todo lo que de él se requiere.

¿A qué puede deberse, si no existen errores gramaticales? Las causas pueden ser muy variadas. Desde la más simple, que sin embargo cuesta mucho trabajo detectar, como es la omisión de un punto gramatical entre dos sentencias —que el compilador no puede detectar, pues las considera como únicas— hasta la más compleja, como puede ser que el programa se introduzca en un bucle y no progrese en su ejecución.

Para poder resolver la situación el programador no cuenta generalmente más que con su propia intuición y su conocimiento del problema.

Sin embargo, algunos compiladores COBOL proporcionan una pequeña herramienta que, a veces, resulta de una eficacia extraordinaria.

Esta herramienta son unos verbos nuevos, que pueden incorporarse al programa fuente en determinados puntos del mismo y que en general realizan una misión de chequeo de la estructura del programa.

De entre los compiladores que poseen verbos de depuración, posiblemente los sistemas de IBM son los que ofrecen mayor variedad. De entre todos, vamos a exponer los siguientes:

TRACE  
EXHIBIT  
ON

### VII-2.1. FUNCION DEL VERBO TRACE

El verbo TRACE (marcar) tiene como objetivo efectuar trazas o marcas de los párrafos o secciones del programa por los cuales está pasando en su fase de ejecución.

El formato de este verbo es:

<u>READY</u> <u>RESET</u>	}	<u>TRACE</u>
------------------------------	---	--------------

Para que comience a ejercer su función debe incluirse una declaración **READY TRACE** en un punto del programa a partir del cual se desea que se efectúe la marca a traza indicada. Con esta declaración se logra, por consiguiente, que a partir de dicho punto del programa y cada vez que se comience la ejecución de un párrafo o de una sección se escriba o se imprima a través del dispositivo de salida lógico el nombre del párrafo o el nombre de la sección que se está ejecutando, facilitándose así una pista sobre el camino que se ha seguido durante la ejecución del programa.

Es obvio indicar que estas marcas deberán incluirse en partes del programa donde existen posibles bifurcaciones, ya que si se incluyen en partes secuenciales del programa no se logrará ninguna pista del camino seguido, ya que en las zonas secuenciales, y como es lógico, el programa siempre se ejecuta secuencialmente.

Para indicar la «marca» final de la serie a chequear, es decir, para delimitar la parte del programa que no interesa continuar chequeando, bastará incluir una declaración **RESET TRACE**, con lo que terminarán las funciones de la declaración **READY TRACE** previamente incluida.

## VII-2.2. FUNCION DEL VERBO EXHIBIT

De la misma forma que el verbo anteriormente estudiado imprime los nombres de los párrafos por los que pasa el flujo del programa, en su ejecución, el verbo **EXHIBIT** se utiliza para imprimir los valores que va teniendo un nombre de datos o varios nombres de datos en los distintos momentos de la ejecución del programa.

El formato de este verbo es el siguiente:

<b>EXHIBIT</b> { <table style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px 5px;"><b>NAMED</b></td> <td rowspan="3" style="font-size: 2em; vertical-align: middle;">}</td> </tr> <tr> <td style="padding: 2px 5px;"><b>CHANGED</b></td> </tr> <tr> <td style="padding: 2px 5px;"><b>CHANGED NAMED</b></td> </tr> </table> <table style="display: inline-table; vertical-align: middle; margin-left: 10px;"> <tr> <td style="padding: 2px 5px;">{ nombre-dato-1</td> <td rowspan="2" style="font-size: 2em; vertical-align: middle;">}</td> <td rowspan="2" style="padding: 2px 5px;">{ ...</td> </tr> <tr> <td style="padding: 2px 5px;">{ literal-no-numérico-1</td> </tr> </table>	<b>NAMED</b>	}	<b>CHANGED</b>	<b>CHANGED NAMED</b>	{ nombre-dato-1	}	{ ...	{ literal-no-numérico-1
<b>NAMED</b>	}							
<b>CHANGED</b>								
<b>CHANGED NAMED</b>								
{ nombre-dato-1	}	{ ...						
{ literal-no-numérico-1								

Como puede observarse por el formato, este verbo tiene tres posibles opciones en cuanto a función:

**NAMED**  
**CHANGED NAMED**  
**CHANGED**

y dos posibles opciones en cuanto al campo o campos a exhibir:

**NOMBRE DE DATOS**  
**LITERAL-NO-NUMERICO**

En cualquiera de las opciones el nombre de datos o los nombres de datos exhibidos pueden ser un campo elemental o un campo compuesto, pero en este último caso los campos elementales que lo forman y cuya **USAGE** no sea **DISPLAY**, no se convierten en su exhibición en formato imprimible.

También, para cualquiera de las opciones, el verbo **EXHIBIT** tiene que ser la última declaración de la sentencia de la cual forma parte.

### VII-2.2.1. Opción **NAMED**

Cuando se ejecuta una orden **EXHIBIT NAMED** se produce la representación impresa de los nombres de datos o literales numéricos que figuren en el formato de la declaración.

El formato más generalizado en la representación del nombre de datos es el siguiente:

nombre-dato-1 = contenido del dato-1 nombre-dato-2 = contenido del dato-2 etc. ...

En el caso de utilizar la opción literal-no-numérico, ésta aparecerá siempre precedida de un espacio.

Se pueden utilizar varios nombres de datos con tal que la suma de los caracteres totalmente expuestos, incluidos espacios y los caracteres del propio nombre de datos, no supere la longitud máxima de la línea de impresión.

#### VII-2.2.1.1. Ejemplo de uso de la opción EXHIBIT NAMED

Supongamos que se ha definido un registro en la DATA DIVISION, en la siguiente forma:

```
01 REGISTRO-1.  
  02 CAMPO-1 PICTURE 999.  
  02 CAMPO-2.  
    03 CAMPO-3 PICTURE 99.  
    03 CAMPO-4 PICTURE 99.
```

y en la WORKING-STORAGE SECTION:

```
77 CAMPO-5 PICTURE 999 VALUE 325.  
77 CAMPO-6 PICTURE XXX VALUE 'ABC'.
```

el empleo de la declaración:

```
EXHIBIT NAMED CAMPO-1 CAMPO-6.
```

provocaría, si en este momento del programa el valor del campo 1 es 030, la siguiente línea de impresión.

```
CAMPO-1 = 030 CAMPO-6 = 'ABC'
```

#### VII-2.2.2. Opción CHANGED

Con esta opción el verbo EXHIBIT hace que se representen únicamente los campos (y no todas las veces) cuando el valor de los mismos cambia con respecto al valor que tenían cuando se ejecutó la declaración EXHIBIT la vez anterior.

Siempre se representarán la primera vez que se ejecute esta orden, ya que se considera que han cambiado de valor.

En el ejemplo del epígrafe anterior, la declaración EXHIBIT CAMPO-6 solamente dará representación impresa la primera vez que se ejecuta, pero las siguientes veces no se imprimirá nada, ya que el valor del CAMPO-5 es invariable, a no ser que en la PROCEDURE DIVISION exista alguna instrucción que modifique su valor.

Con esta opción solamente puede representarse un nombre de datos. Si fuese necesario representar más nombre de datos habría que utilizar más órdenes EXHIBIT CHANGED.

En el caso de que en diferentes partes del programa se introduzcan dos o más verbos EXHIBIT CHANGED referidos al mismo nombre de datos queda asociado el valor del dato en el momento de la ejecución a cada una de las distintas declaraciones. Es decir, que siempre que cambie el valor de nombre de dato de una declaración a otra no se exhibirá su valor si no ha habido también un cambio de valor en relación a la última vez que se ejecutó la orden correspondiente.

**Ejemplo:**

Consideremos los datos del ejemplo anterior y supongamos que la PROCEDURE DIVISION contiene dos sentencias EXHIBIT CHANGED:

PROCEDURE DIVISION.

_____  
_____  
_____  
_____  
_____

EXHIBIT CHANGED CAMPO-1.

_____  
_____  
_____

EXHIBIT CHANGED CAMPO-1.

_____  
_____

Primera ejecución CAMPO-1 vale 020 y no varía.

Se representa el valor del CAMPO-1 020 para la primera orden.

Se representa el valor del CAMPO-1 = 020 para la segunda orden.

Ahora cambia CAMPO-1 a valor 030.

Se representa CAMPO-1 = 030 para la primera orden.

Vuelve a cambiar CAMPO-1 a valor 020.

No se representa CAMPO-1 para la segunda orden, ya que aunque ha cambiado de valor (030 pasa a 020) no lo ha hecho con respecto al valor que tenía cuando se ejecutó por primera vez en la segunda declaración.

**VII-2.2.3. Opción CHANGED NAMED**

Esta opción es una combinación de las dos anteriores; es decir, representa los nombre de los datos cuando el valor de éstos cambia, como ocurre en la opción CHANGED, y representa también el nombre del dato acompañando a su valor, como ocurre con la opción NAMED.

### VII-2.3. FUNCION DEL VERBO ON

El verbo ON se utiliza para especificar cuándo tienen que ejecutarse determinadas declaraciones.

El formato de este verbo es:

<b>ON</b> entero-1 [ <b>AND EVERY</b> entero-2 ] [ <b>UNTIL</b> entero-3 ] { declaración-imperativa-1 } NEXT SENTENCE
{ ELSE } { declaración-imperativa-2 } { OTHERWISE } { NEXT SENTENCE }

Cada vez que el flujo del programa llega a una declaración ON se incrementa en una unidad un contador que ha sido generado por el compilador, con un valor inicial igual a cero.

Cuando el valor del contador es igual a  $\text{entero-1} + K \times \text{entero-2}$  ( $K$  entero positivo) y menor que  $\text{entero-3}$  se ejecutan las declaraciones imperativas-1 o la próxima sentencia, dependiendo de lo especificado. En caso contrario se ejecutan las declaraciones que siguen a ELSE.

Como se observa, este verbo se ejecuta de forma totalmente análoga a la expuesta para IF.

Si no se especifica la opción *AND EVERY* entero-2 y si UNTIL entero-3 se presupone entero-2 con valor 1.

Si no se especifica ninguna de las dos opciones, la declaración imperativa se ejecuta una vez

#### **Ejemplo:**

```
ON 3 AND EVERY 4 UNTIL 12 DISPLAY 'CONTROL-1' ELSE DISPLAY 'CONTROL-2'
```

Se exhibirá el mensaje CONTROL-1 la tercera, séptima y undécima vez que el flujo del programa pase por la orden ON. El mensaje CONTROL-2 se exhibirá las restantes veces.





# ANEXO A

RELACION DE PALABRAS  
RESERVADAS COBOL







PALABRA RESERVADA	SIGNIFICADO	CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
ABOUT	aproximadamente	S	N	N	N	N	N	N	S
ACCEPT	aceptar	S	S	S	S	S	S	S	S
ACCESS	acceso	S	S	S	S	S	S	S	S
ACTUAL	real	S	S	S	S	S	S	S	S
ADD	sumar	S	S	S	S	S	S	S	S
ADDRESS	dirección	S	N	S	N	N	N	S	S
ADVANCING	avanzando	S	S	S	S	S	S	S	S
AFTER	después	S	S	S	S	S	S	S	S
ALL	todo	S	S	S	S	S	S	S	S
ALPHABETIC	alfabético	S	S	S	S	S	S	S	S
ALPHANUMERIC	alfanumérico	S	N	N	N	N	N	S	S
ALTER	alterar	S	S	S	N	S	N	S	S
ALTERNATE	alternativa	S	S	S	N	S	S	S	S
AN	alfanumérico	S	N	N	N	N	N	N	S
AND	y	S	S	S	S	S	S	S	S
APPLY	aplicar	S	S	N	S	S	S	N	S
ARE	son	S	S	S	S	S	S	S	S
AREA(S)	área	S	S	S	S	S	S	S	S
ASCENDING	ascendente	S	S	S	S	S	S	S	S
ASSIGN	asignar	S	S	S	S	S	S	S	S
AT	a	S	S	S	S	S	S	S	S
AUTHOR	autor	S	S	S	S	S	S	S	S
BCD	decimal codificado en binario	S	N	N	N	N	N	N	S
BEFORE	antes	S	S	S	S	S	S	S	S
BEGINNING	comienzo	S	S	S	N	S	S	N	S
BEGINNING-FILE-LABEL	etiqueta de comienzo de fichero	S	N	N	N	N	N	N	S
BEGINNING-TAPE-LABEL	etiqueta de comienzo volumen	S	N	N	N	N	N	N	S
BINARY	binario	S	N	N	N	N	N	N	S
BIT	bit	S	N	N	N	N	N	N	S
BLANK	espacio	S	S	S	S	S	S	S	S
BLOCK	bloque	S	S	S	S	S	N	S	S
BY	por	S	S	S	S	S	N	S	S
CALL	llamar	S	S	N	N	S	N	S	S
CANCEL	cancelar	S	N	N	N	N	N	S	S
CARD-PUNCH	perforadora de fichas	N	N	N	N	S	N	S	N
CARD-READER	lectora de fichas	N	N	N	N	S	N	S	N

S: Indica palabra implementada en el compilador del sistema.  
N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO								
		CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	H-W-B sistema GE-600
CF	control de pie de página	S	S	S	N	S	N	S	S
CH	control de cabecera de página	S	S	S	S	S	N	S	S
CHANGED	cambiado	N	S	N	N	S	N	N	N
CHARACTERS	caracteres	S	S	S	N	S	N	S	S
CHEK	chequear	S	N	N	N	N	N	N	S
CHECKING	comprobar	N	S	N	N	S	N	N	N
CLASS	clase	S	S	N	N	S	N	S	S
CLOCK-UNITS	unidad de tipo	S	S	S	N	S	N	S	S
CLOSE	cerrar	S	S	S	S	S	S	S	S
COBOL	lenguaje común orientado a problemas comerciales	S	S	S	S	S	S	S	S
CODE	código	S	S	S	N	S	N	S	S
COLUMN	columna	S	S	S	S	S	N	S	S
COMMA	coma	S	S	S	S	S	N	S	S
COMMON-STORAGE	almacenamiento común	S	N	N	N	N	N	N	S
COMP	cálculo	S	N	S	N	N	S	S	S
COMP-1	cálculo-1	S	N	N	N	N	S	N	S
COMP-2	cálculo-2	S	N	N	N	N	S	N	S
COMP-3	cálculo-3	N	N	N	N	N	S	S	S
COMPUTATIONAL	para cálculo	S	S	S	S	S	S	S	S
COMPUTATIONAL-1	para cálculo-1	S	S	N	S	S	S	N	S
COMPUTATIONAL-2	para cálculo-2	S	S	N	S	S	S	N	S
COMPUTATIONAL-3	para cálculo-3	N	S	N	N	S	S	S	S
COMPUTE	calcular	S	S	S	N	S	S	S	S
CONFIGURATION	configuración	S	S	S	S	S	S	S	S
CONSOLE	consola	S	S	N	N	S	S	N	S
CONSTANT	constante	S	N	N	N	N	N	N	S
CONTAINS	contiene	S	S	S	S	N	S	S	S
CONTROL(S)	control	S	S	S	N	S	N	S	S
CONVERSION	conversión	S	N	N	S	N	S	N	N
COPY	copiar	S	S	S	N	S	N	S	S
CORR	correspondiendo	S	N	S	N	N	N	S	S
CORRESPONDIGN	correspondiendo	S	S	S	N	S	N	S	S
CREATING	creación	N	S	N	N	S	S	S	N
CURRENCY	moneda	S	N	S	S	N	S	S	S
CYCLES	ciclos	N	S	N	N	S	N	N	N

S: Indica palabra implementada en el compilador del sistema.  
N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO								
		GDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
DATA	dato	S	S	S	S	S	S	S	S
DATE	fecha	S	N	N	N	N	S	N	S
DATE-COMPILED	fecha de compilación	S	S	S	N	S	S	S	S
DATE-TODAY (TODAY-DATE)	fecha del día	S	N	N	S	N	N	N	N
DATE-WRITTEN	fecha de escritura	S	S	S	S	S	S	S	S
DAY	día	S	N	N	N	N	S	N	S
DE	detalle	S	S	S	S	S	N	S	S
DECIMAL	decimal	S	N	N	N	N	N	N	S
DECIMAL-POINT	punto decimal	S	S	S	S	S	S	S	S
DECLARATIVES	declarativas	S	S	S	S	S	S	S	S
DELETE	borrar	S	N	N	S	N	S	N	N
DENSITY	densidad	S	N	N	S	N	N	N	S
DEPENDING	dependiendo	S	S	S	S	S	S	S	S
DESCENDING	descendiendo	S	S	S	N	S	S	S	S
DETAIL	detalle	S	S	S	S	S	N	S	S
DIGITS	dígitos	S	N	N	N	N	N	N	S
DIRECT	directo	S	S	N	N	S	S	N	N
DIRECT-ACCESS	acceso directo	N	S	N	N	S	S	N	N
DISPLAY	exhibir	S	S	S	S	S	S	S	S
DISPLAY-ST	exhibir-libras	N	S	N	N	S	N	N	N
DIVIDE	dividir	S	S	S	S	S	S	S	S
DIVIDED	dividido	S	N	N	N	N	N	N	S
DIVISION	división	S	S	S	S	S	S	S	S
DOLLAR	dólar	S	S	N	N	N	N	N	S
DOWN	bajo	S	N	S	N	S	S	S	S
ELSE	en caso contrario	S	S	S	S	S	S	S	S
END	fin	S	S	S	S	S	S	S	S
END-OF-PAGE	fin de ficha	S	N	N	N	N	N	N	S
ENDING	terminación (etiqueta de cola)	S	S	S	N	S	S	S	S
ENDING-FILE-LABEL	etiqueta fin de fichero	S	N	N	N	N	N	N	S
ENDING-TAPE-LABEL	etiqueta fin de fichero volumen	S	N	N	N	N	N	N	S
ENTER	introducir	S	S	S	S	S	S	S	S
ENTRY	entrada	S	S	N	N	S	S	N	S
ENVIRONMENT	características	S	S	S	S	S	S	S	S
EQUAL(S)	igual	S	S	S	S	S	S	S	S

S: Indica palabra implementada en el compilador del sistema.  
N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO	CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
ERROR	error	S	S	S	S	S	S	S	S
ERROR-CODE	códigos de error	S	N	N	S	N	N	N	N
EVEN	incluso	S	N	N	N	N	N	S	S
EVERY	cada	S	S	S	S	S	S	S	S
EXAMINE	examinar	S	S	S	S	S	S	S	S
EXCEEDS	exceder	S	N	N	N	N	N	S	S
EXHIBIT	exhibir	S	S	N	N	S	N	N	N
EXIT	salida	S	S	S	S	S	S	S	S
FD	descripción de fichero	S	S	S	S	S	S	S	S
FILE(S)	fichero	S	S	S	S	S	S	S	S
FILE-CONTROL	control de ficheros	S	S	S	S	S	S	S	S
FILE-LIMIT(S)	límite de fichero	S	S	S	S	S	S	S	S
FILLER	relleno	S	S	S	S	S	S	S	S
FINAL	final	S	S	S	N	S	N	S	S
FIRST	primero	S	S	S	S	S	S	S	S
FLOAT	flotante	S	S	N	N	N	N	N	S
FOOTING	pie de página	S	S	S	N	S	N	S	S
FOR	para	S	S	S	S	S	S	S	S
FORM-OVERFLOW	capacidad excedida de formulario	N	S	N	N	S	S	N	N
FORMAT	formato	N	N	N	S	N	N	N	S
FROM	de	S	S	S	S	S	S	S	S
GENERATE	general	S	S	S	S	S	N	S	S
GIVING	dado	S	S	S	S	S	S	S	S
GO	ir	S	S	S	S	S	S	S	S
GREATER	mayor	S	S	S	S	S	S	S	S
GROUP	grupo	S	S	S	S	S	N	S	S
HEADING	encabezamiento	S	S	S	N	S	N	S	S
HIGH-VALUE(S)	altos valores	S	S	S	S	S	S	S	S
HOLD	retener	S	S	S	N	S	N	N	S
ID	identificar	S	N	N	N	S	N	N	S
IDENTIFICATION	identificación	S	S	S	S	S	S	S	S
IF	sí	S	S	S	S	S	S	S	S
IN	en	S	S	S	N	S	S	S	S
INCLUDE	incluir	S	S	N	N	S	S	N	N
INDEX	índice	S	N	S	S	S	N	S	S
INDEXED	índices	S	S	S	S	S	N	S	S

S: Indica palabra implementada en el compilador del sistema.  
N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO								
		CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
INDICATE	indicar	S	S	S	N	S	N	S	S
INITIATE	iniciar	S	S	S	S	S	N	S	S
INPUT	entrada	S	S	S	S	S	S	S	S
INPUT-OUTPUT	entrada-salida	S	S	S	S	S	S	S	S
INSTALLATION	instalación	S	S	S	S	N	S	S	S
INTO	en, de	S	S	S	S	S	S	S	S
INVALID	inválido	S	S	S	S	S	S	S	S
I-O	entrada-salida	S	S	S	S	S	S	S	S
I-O-CONTROL	control de entrada-salida	S	S	S	S	S	S	S	S
IS	es	S	S	S	S	S	S	S	S
JUSTIFIED	alineado	S	S	S	S	S	S	S	S
KEY	clave	S	S	S	S	S	S	S	S
KEYS	claves	S	S	S	N	N	N	S	S
LABEL	etiqueta	S	S	S	S	S	S	S	S
LABELS	etiquetas	S	S	N	N	S	S	S	S
LAST	último	S	S	S	S	S	N	S	S
LEADING	que aparecen al principio	S	S	S	S	S	S	S	S
LEAVING	dejando	S	S	N	N	N	N	N	S
LEFT	izquierda	S	S	S	S	S	S	S	S
LENGTH	longitud	N	N	N	S	N	N	N	S
LESS	menor	S	S	S	S	S	S	S	S
LIBRARY	biblioteca	S	S	N	N	S	N	N	S
LIMIT	límite	S	N	S	S	S	N	S	S
LIMITS	límites	S	N	S	S	S	N	S	S
LINE(S)	línea(s)	S	S	S	S	S	S	S	S
LINE-COUNTER	contador de líneas	S	S	S	N	S	N	S	S
LINKAGE	unión	S	S	N	S	S	S	S	S
LOCATION	ubicación	S	N	N	N	S	N	N	S
LOCK	bloqueo	S	S	S	N	S	S	S	S
LOOP	bucle	N	N	N	S	N	N	N	S
LOW	bajo	S	N	N	S	N	N	N	N
LOW-VALUE(S)	bajos niveles	S	S	S	S	S	S	S	S
MEMORY	memoria	S	N	S	S	N	N	S	S
MINUS	menos	S	N	N	N	N	N	N	S
MODE	modo	S	S	S	S	S	S	S	S
MODULES	módulos	S	N	N	N	N	N	S	S

S: Indica palabra implementada en el compilador del sistema.

N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO								
		CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRI-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
MORE-LABEL	más etiquetas	S	S	S	N	S	S	N	N
MULTIPLE	múltiples	S	N	S	S	S	N	S	S
MULTIPLIED	multiplicado	S	N	N	N	N	N	N	S
MULTIPLY	multiplicar	S	S	S	S	S	S	S	S
NAMED	nombrado	N	S	N	N	S	N	N	N
NEGATIVE	negativo	S	S	S	S	S	S	S	S
NEXT	próximo	S	S	S	S	S	S	S	S
RANDOM	al azar	S	S	S	S	S	S	S	S
RANGE	escala	S	N	N	N	N	N	N	S
RD	descripción de informe	S	S	S	S	S	N	S	S
READ	leer	S	S	S	S	S	S	S	S
READY	listo	S	S	N	N	S	N	N	N
RECORD	registro	S	S	S	S	S	S	S	S
RECORDING	registrar	S	S	N	N	S	S	S	S
RECORDS	registros	S	S	S	S	S	S	S	S
REDEFINES	redefine	S	S	S	S	S	S	S	S
REEL	carrete	S	S	S	S	S	S	S	S
REEL-NUMBER	número de carrete	S	N	N	N	N	N	N	S
RELATIVE	relativa	S	S	N	N	S	N	N	N
REALEASE	facilitar	S	S	S	N	S	S	S	S
REMAINDER	resto	S	N	N	N	N	S	N	S
REMARKS	observaciones	S	S	S	S	S	S	S	S
RENAMES	renombrar	S	N	S	N	N	S	S	S
RENAMING	renombrando	S	N	N	N	N	N	N	S
REPLACING	sustituyendo	S	S	S	S	S	S	S	S
REPORT	informe	S	N	S	S	S	N	S	S
REPORTING	producir grupo de informe	S	S	S	N	S	N	N	S
REPORTS	informes	S	S	S	S	S	N	N	S
RERUN	pasada	S	S	S	S	S	S	S	S
RESERVE	reservar	S	S	S	N	S	S	S	S
RESET	restaurar	S	S	S	N	S	N	S	S
RESTRICTED	limitado	N	S	N	N	S	S	N	N
RETENTION-PERIOD	período de retención	N	N	N	S	N	N	N	S
RETURN	volver	S	S	S	N	S	S	S	S
REVERSED	invertido	S	S	S	N	S	S	S	S
REWIND	rebobinar	S	S	S	S	S	S	S	S

S: Indica palabra implementada en el compilador del sistema.  
N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO	CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
REWRITE	volver a grabar	S	S	N	S	S	S	N	N
RF	informe pie de página	S	S	S	S	S	N	S	S
RH	informe cabecera de página	S	S	S	S	S	N	S	S
RIGHT	derecha	S	S	S	S	S	S	S	S
ROUNDED	redondeado	S	S	S	S	S	S	S	S
RUN	funcionamiento	S	S	S	S	S	N	N	S
SAME	misma	S	S	S	S	S	S	S	S
SD	descripción de archivo de clasificación	S	S	S	N	S	S	S	S
SEARCH	buscar	S	S	S	N	S	S	S	S
SECTION	sección	S	S	S	S	S	S	S	S
SECURITY	seguridad	S	S	S	S	S	N	S	S
SEEK	buscar	S	N	S	S	N	S	S	S
SEGMENT-LIMIT	límite del segmento	S	N	S	N	N	N	S	S
SELECT	seleccionar	S	S	S	S	S	S	S	S
SENTENCE	sentencia	S	S	S	S	S	S	S	S
SEQUENCED	consecuencia	S	N	N	N	N	N	N	S
SEQUENTIAL	secuencial	S	S	S	S	S	S	S	S
SET	fijar	S	N	S	N	S	N	S	S
SIGN	signo	S	N	S	S	S	S	S	S
SIGNED	con signo	S	S	S	N	N	N	N	S
SIZE	tamaño	S	S	S	S	S	S	S	S
SORT	clasificar	S	S	S	N	S	S	S	S
SOURCE	fuelle	S	S	S	S	S	N	S	S
SOURCE-COMPUTER	ordenador-fuelle	S	S	S	S	S	S	S	S
SPACE(S)	espacio(s)	S	S	S	S	S	S	S	S
SPECIAL-NAMES	nombres-especiales	S	S	S	S	S	S	S	S
STANDARD	estándar	S	S	S	S	S	S	S	S
STATUS	estado (de un switch)	S	N	S	S	N	N	S	S
STOP	parar	S	S	S	S	S	S	S	S
SUBTRACT	restar	S	S	S	S	S	S	S	S
SUM	sumar	S	S	S	N	S	S	S	S
SUPERVISOR	supervisor	S	N	N	N	N	N	N	S
SUPRESS	suprimir	S	N	N	N	N	N	N	S
SWITCH	switch (conmutador)	S	N	N	N	N	N	N	S
SYMBOLIC	simbólico	S	S	N	S	S	S	N	N
SYNC	sincronizado	S	N	S	N	N	S	S	S
SYNCHRONIZED	sincronizado	S	N	S	S	N	S	S	S

S: Indica palabra implementada en el compilador del sistema.

N: Indica palabra no implementada en el compilador del sistema.

PALABRA RESERVADA	SIGNIFICADO								
		CDC sistema 6.000	IBM sistemas 360/370	UNIVAC sistema 1108	HW-B sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sist. 9200/9300	HW-B sistema GE-600
SYSIN	entrada lógica	N	N	N	N	S	S	N	N
SYSOUT	salida lógica	S	N	S	S	N	N	S	N
SYSPUNCH	salida de perforación	N	S	N	N	S	S	N	S
TALLY	contador	S	S	S	S	S	S	S	S
TALLYNG	cuenta	S	S	S	S	S	S	S	S
TAPE	banda magnética	S	N	S	N	S	S	S	S
TERMINATE	terminar	S	S	S	S	S	S	S	S
THAN	que	S	S	S	S	S	S	S	S
THEN	entonces	S	S	N	N	S	S	S	S
THROUGH	hasta	S	N	S	N	N	S	S	S
THRU	hasta	S	S	S	S	S	S	S	S
TIMES	veces	S	S	S	S	S	S	S	S
TO	a, hasta	S	S	S	S	S	S	S	S
TRACE	seguir	N	S	N	N	S	N	N	N
TRACK-AREA	área de pista	N	S	N	N	S	S	N	N
TRACKS	pistas	N	S	N	N	S	S	N	N
TRANSFORM	transformar	N	S	N	N	S	N	N	N
TRY	intentar	N	S	N	N	S	N	N	N
TYPE	tipo	S	S	S	S	S	S	S	S
UNEQUAL	distinto, desigual	S	N	N	N	N	N	S	S
UNIT	unidad	S	S	S	S	S	S	S	S
UP	arriba	S	N	S	N	S	N	S	S
UNIT-RECORD	registro-unitario	N	S	N	N	S	S	N	N
UNITS	unidades	N	N	N	N	S	S	N	N
UNTIL	hasta	S	N	S	S	S	S	S	S
UPON	en, según, a través de	S	S	S	N	S	S	S	S
USAGE	uso	S	S	S	S	S	S	S	S
USE	utilizar	S	S	S	S	S	S	S	S
USING	utilizando	S	S	S	S	S	S	S	S
UTILITY	utilidad	N	S	N	N	S	S	N	N
VALUE(S)	valor(es)	S	S	S	S	S	S	S	S
VARYING	variando	S	S	S	S	S	S	S	S
WHEN	cuando	S	S	S	S	S	S	S	N
WITH	con	S	S	S	S	S	N	S	S
WORDS	palabras	S	N	S	N	N	N	S	S
WORKING-STORAGE	memoria de trabajo	S	S	S	S	S	S	S	S
WRITE	escribir	S	S	S	S	S	S	S	S
ZERO(S)	cero(s)	S	S	S	S	S	S	S	S
ZEROES	ceros	S	S	S	S	S	S	S	S

S: Indica palabra implementada en el compilador del sistema.

N: Indica palabra no implementada en el compilador del sistema.

No se han figurado en esta lista algunas palabras, que son ampliación implementada solamente en uno de los sistemas.





# ANEXO B

SECUENCIAS DE CLASIFICACION





**SECUENCIAS DE CLASIFICACION (COLLATING SEQUENCE) PARA LOS 51  
CARACTERES BASICOS DEL COBOL EN LOS DISTINTOS SISTEMAS  
(de menor a mayor)**

N.º orden	CDC sistema 6000	IBM sistemas 360 y 370	UNIVAC sistema 1108	HW-B sistema GE-100	SIEMENS sistema 4004	CII sistema IRIS 50	UNIVAC sistemas 9200-9300	HW-B sistema GB-600
1	∅	∅	∅	∅	∅	∅	∅	∅
2	>	.	A	1	.	.	.	1
3	.	<	B	2	<	<	<	2
4	)	(	C	3	(	(	(	3
5	:	+	D	4	+	+	+	4
6	+	\$	E	5	\$	\$	\$	5
7	\$	*	F	6	*	*	*	6
8	*	)	G	7	)	)	)	7
9	-	;	H	8	;	;	;	8
10	/	-	I	9	-	-	-	9
11	/	/	J	>	/	/	/	>
12	(	.	K	∅	.	.	.	∅
13	=	>	L	A	>	>	>	A
14	.	=	M	B	=	=	=	B
15	<	.	N	C	.	.	.	C
16	A	A	O	D	A	A	A	D
17	B	B	P	E	B	B	B	E
18	C	C	Q	F	C	C	C	F
19	D	D	R	G	D	D	D	G
20	E	E	S	H	E	E	E	H
21	F	F	T	I	F	F	F	I
22	G	G	U	.	G	G	G	.
23	H	H	V	(	H	H	H	(
24	I	I	W	<	I	I	I	<
25	J	J	X	J	J	J	J	J
26	K	K	Y	K	K	K	K	K
27	L	L	Z	L	L	L	L	L
28	M	M	)	M	M	M	M	)
29	N	N	-	N	N	N	N	-
30	O	O	+	O	O	O	O	+
31	P	P	<	P	P	P	P	<
32	Q	Q	=	Q	Q	Q	Q	=
33	R	R	>	R	R	R	R	>
34	S	S	\$	-	S	S	S	-
35	T	T	*	\$	T	T	T	\$
36	U	U	(	*	U	U	U	*
37	V	V	.	)	V	V	V	)
38	W	W	∅	:	W	W	W	:
39	X	X	1	.	X	X	X	.
40	Y	Y	2	+	Y	Y	Y	+
41	Z	Z	3	/	Z	Z	Z	/
42	∅	∅	4	S	∅	∅	∅	S
43	1	1	5	T	1	1	1	T
44	2	2	6	U	2	2	2	U
45	3	3	7	V	3	3	3	V
46	4	4	8	W	4	4	4	W
47	5	5	9	X	5	5	5	X
48	6	6	.	Y	6	6	6	Y
49	7	7	:	Z	7	7	7	Z
50	8	8	/	,	8	8	8	,
51	9	9	.	=	9	9	9	=



---

## ANEXO C

OPCIONES DE LA CLAUSULA  
USAGE Y REPRESENTACION  
INTERNA DE LOS DATOS

---



## OPCIONES DE LA CLAUSULA USAGE Y REPRESENTACION INTERNA PARA CADA SISTEMA

OPCION	CDC sistema 6000	IBM sistemas 360 y 370	UNIVAC sistema 1108	HW-BULL sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sistemas 9200-9300	HW-BULL sistema GE-600
								1 palabra=36 bits
DISPLAY	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres	Código binario de n caracteres
DISPLAY-1								Código binario de n caracteres
DISPLAY-2								Código binario de n caracteres (especial)
COMPUTATIONAL (COMP)	Código binario de n caracteres	$1 \leq n \leq 4$ , código binario sobre 2 octetos $5 \leq n \leq 9$ , código binario sobre 4 octetos $10 \leq n \leq 18$ , código binario sobre 8 octetos	Código binario de n caracteres	Código binario de n caracteres	$1 \leq n \leq 4$ , código binario sobre 2 octetos $5 \leq n \leq 9$ , código binario sobre 4 octetos $10 \leq n \leq 18$ , código binario sobre 8 octetos	$1 \leq n \leq 4$ , código binario sobre 2 octetos $5 \leq n \leq 9$ , código binario sobre 4 octetos $10 \leq n \leq 18$ , código binario sobre 8 octetos	decimal empaquetado sobre parte entera de $\left[ \frac{n}{2} + 1 \right]$	$n < 18$ ( $d = n.$ º de cifras decimales) 1 palabra máquina cuando $n - d < 9 - x$ 2 palabras máquina cuando $n - d > 9 - x$ siendo x valor superior a d, más a cercano a 0,3.508

OPCION	CDC sistema 6000	IBM sistemas 360 y 370	UNIVAC sistema 1108	HW-BULL sistema GE 100	SIEMENS sistema 4004	CII sistema IRIS-50	UNIVAC sistemas 9200-9300	HW-BULL sistema GE-600
								1 palabra=36 bits
COMPUTATIONAL-1 (COMP-1)	$n \leq 14$ coma flotante simple precisión $15 \leq n \leq 18$ coma flotante doble precisión	coma flotante simple precisión sobre 4 bytes (se debe omitir n)	$n < 18$ en x bits $\left[ x = 6 \frac{4n}{7} + 1 \right]$		coma flotante simple precisión sobre 4 bytes (se debe omitir n)	coma flotante simple precisión sobre 4 bytes (se debe omitir n)		$n < 9$ binario fijo sobre una palabra $9 \leq n \leq 18$ binario fijo sobre dos palabras
COMPUTATIONAL-2 (COMP-2)	$n \leq 18$ u omisión coma flotante simple precisión	coma flotante doble precisión sobre 8 bytes (se debe omitir n)			coma flotante doble precisión sobre 8 bytes (se debe omitir n)	coma flotante doble precisión sobre 8 bytes (se debe omitir n)		$n < 9$ binario flotante sobre dos palabras $9 \leq n \leq 18$ binario flotante sobre dos palabras
COMPUTATIONAL-3 (COMP-3)		decimal empaquetado sobre parte entera de $\left[ \frac{n}{2} + 1 \right]$			decimal empaquetado sobre parte entera de $\left[ \frac{n}{2} + 1 \right]$	decimal empaquetado sobre parte entera de $\left[ \frac{n}{2} + 1 \right]$	decimal empaquetado sobre parte entera de $\left[ \frac{n}{2} + 1 \right]$	$n \leq 10$ binario fijo sobre una palabra máquina

n especifica el tamaño del campo definido por las clausulas SIZE o PICTURE.



---

# ANEXO D

TABLAS RESUMEN  
DE COMPARACIONES,  
MOVIMIENTOS Y SERIES DE  
CARACTERES PERMISIBLES

---



## EXPRESIONES ARITMETICAS

(Pares de símbolos consecutivos permitidos en una expresión aritmética)

2.º símbolo → 1.º símbolo ↓	Campo de datos o literal	* / ó **	+ ó -	(	)
Campo de datos o literal	N	P	P	N	P
* / ó **	P	N	P <small>como 1.º carácter de un literal</small>	P	N
+ ó -	P	N	N	P	N
(	P	N	P	P	N
)	N	P	P	N	P

P significa secuencia permitida.  
 N significa secuencia no permitida.

## EXPRESIONES CONDICIONALES COMPUESTAS

(Pares de símbolos consecutivos permitidos en una expresión condicional compuesta)

2.º símbolo → 1.º símbolo ↓	Expresión condicional simple	OR	AND	NOT	(	)
Expresión condicional simple	N	P	P	N	N	P
OR	P	N	N	P	P	N
AND	P	N	N	P	P	N
NOT	P <small>si la expresión condicional simple no contiene NOT</small>	N	N	N	P	N
(	P	N	N	P	P	N
)	N	P	P	N	N	P

**COMPARACIONES PERMITIDAS ENTRE DOS CAMPOS,  
SEGUN SU DEFINICION**

2.º campo → 1.º campo ↓	Compuesto	Alfabético	Alfanumérico	Decimal externo	Decimal interno	Binario	Punto flotante externo	Punto flotante interno	Editado	Constante figurativa
Compuesto	P	P	P	P	P	P	P	P	P	P
Alfabético	P	P	P	N	N	N	N	N	N	P con SPACE y ALL 'carácter'
Alfanumérico	P	P	P	P en enteros	N	N	N	N	P	P
Decimal externo	P	N	P en enteros	P	P	P	P	P	N	P con ZERO y ALL 'carácter'
Decimal interno	P	N	N	P	P	P	P	P	N	P con zero
Binario	P	N	N	P	P	P	P	P	N	P con zero
Punto flotante externo	P	N	N	P	P	P	P	P	N	P con zero
Punto flotante interno	P	N	N	P	P	P	P	P	N	P con zero
Editado	P	N	P	N	N	N	N	N	N	P salvo QUOTE
Constante figurativa	P	P con SPACE y ALL 'carácter'	P	P con ZERO y ALL 'carácter'	P con ZERO y ALL 'carácter'	P con zero	P con zero	P con zero	P salvo QUOTE	

P significa comparación permitida.  
N significa comparación no permitida.

## MOVIMIENTOS PERMISIBLES ENTRE CAMPOS, SEGUN SU DEFINICION

Campo destino → Campo origen ↓	Compuesto	Alfabético	Decimal externo	Decimal interno	Binario	Punto flotante externo	Punto flotante interno	Editado
Compuesto	P	P	P	N	N	N	N	N
Alfabetico	P	P	P	N	N	N	N	N
Decimal externo	P	P	P	N	N	N	N	N
Decimal interno	P	N	P	P	P	P	P	P
Binario	P	N	P	P	P	P	P	P
Punto flotante externo	P	N	N	P	P	P	P	P
Punto flotante interno	P	N	N	P	P	P	P	P
Editado	P	N	P	N	N	N	N	N
ZEROS	P	N	P	P	P	P	P	P
SPACES	P	P	P	N	N	N	N	N
ALL 'carácter' HIGH-VALUE LOW- VALUE y QUOTES	P	N	P	N	N	N	N	N

P significa movimiento permitido.

N significa movimiento no permitido.

## ORDEN DE SUCESION O PRECEDENCIA DE LOS CARACTERES DE UNA PICTURE

		INSERCIÓN FIJA							OTROS SÍMBOLOS												
		B	O	,	.	{±}	{±}	{CR DB}	\$	A X	P	P	S	V	{Z*}	{Z*}	9	{±}	{±}	\$	\$
INSERCIÓN FIJA	B	*	*	*	*	*		*	*	*			*	*	*	*	*	*	*	*	*
	O	*	*	*	*	*		*	*	*			*	*	*	*	*	*	*	*	*
	,	*	*	*	*	*		*		*			*	*	*	*	*	*	*	*	*
	.	*	*	*		*		*					*		*	*	*	*		*	
	{±}									*			*								
	{±}	*	*	*	*			*		*			*	*	*	*	*			*	*
	{CR DB}	*	*	*	*			*		*			*	*	*	*	*				
	\$					*				*	*		*								
OTROS SÍMBOLOS	A X	*	*						*		*					*					
	P									*		*	*								
	P	*	*	*		*		*				*		*		*	*		*	*	
	S																			*	
	V	*	*	*		*		*			*	*		*		*	*		*	*	
	{Z*}	*	*	*		*		*					*		*					*	
	{Z*}	*	*	*	*	*		*		*	*		*	*	*					*	
	9	*	*	*	*	*		*	*	*		*	*	*		*	*		*	*	
	{±}	*	*	*				*										*			
	{±}	*	*	*	*			*		*	*		*				*	*			
	\$	*	*	*		*														*	*
	\$	*	*	*	*	*					*			*						*	*

Este cuadro muestra el orden de sucesión o precedencia en una PICTURE. El * en una intersección indica que el símbolo o símbolos de la parte superior de la columna pueden preceder, en una determinada serie de caracteres, al símbolo, o los símbolos, que están a la izquierda de la fila. Una llave significa que los caracteres que se hallan en ella se excluyen mutuamente en la misma PICTURE. El carácter P y los caracteres de inserción + y - figuran dos veces en la tabla, una para su utilización a la izquierda de los caracteres numéricos y otra para su empleo a la derecha de los caracteres numéricos de la PICTURE. Por lo menos, uno de los símbolos A, X, Z, 9, *, o por lo menos dos de los símbolos +, - ó \$ deben estar presentes en una serie PICTURE. Z *, y la inserción no fija de \$, + y -, aparecen dos veces en la tabla, una para su utilización antes del signo decimal y otra para su uso después del signo decimal.



# APENDICE A

ENLACE ENTRE PROGRAMAS







El enlace entre dos programas consiste en la orden emanada de un programa que se esté ejecutando para que se comience a ejecutar otro programa. La justificación de esta forma de operar está determinada cuando en una instalación diversos programas realizan rutinas comunes.

Para evitar que un cambio en la estructuración de dichas rutinas comunes provoque una reprogramación de cada uno de los programas que las emplean es conveniente que éstas no formen parte de cada uno de dichos programas, sino que formen por sí solas un programa independiente, y que cada uno de los programas que lo necesiten se enlacen con él en el momento de su ejecución.

Estas rutinas de uso común se denominan subprogramas. Estos subprogramas se escriben con nombres de datos correspondientes a variables ficticias denominadas argumentos, los cuales son reemplazados por los nombres de datos reales correspondientes, en el momento de la ejecución.

El programa que efectúa la llamada a otro u otros subprogramas se denomina programa de llamada o principal. Un subprograma puede a su vez efectuar llamadas a otros subprogramas pero no a sí mismo.

Las órdenes necesarias para poder enlazar un programa principal y un subprograma difieren sensiblemente de un sistema a otro.

En las siguientes páginas se expone de manera muy sintetizada la forma e instrucciones utilizadas en los distintos sistemas utilizados en los ejemplos de esta obra.

## CODIFICACIONES NECESARIAS EN LAS DIVISIONES DATA Y PROCEDURE

SISTEMA	En el programa de llamada	En los subprogramas	Observaciones
CDC, sistema 6000	DATA DIVISION COMMON-STORAGE SECTION definición de argumentos de enlace-1 PROCEDURE DIVISION <u>ENTER LINKAGE</u> { <u>CALL</u> } [nombre-lenguaje] { <u>ENTER</u> } { nombre-punto-entrada } { nombre-subprograma } <u>USING</u> lista de argumentos] <u>ENTER COBOL</u>	DATA DIVISION COMMON-STORAGE SECTION definición de argumentos de enlace-2 punto de entrada PROCEDURE DIVISION. <u>ENTER LINKAGE</u> <u>ENTRY</u> nombre-punto-entrada [ <u>USING</u> lista de argumentos ] <u>ENTER COBOL.</u> ===== (rutina común) retorno al programa de llamada <u>ENTER LINKAGE</u> { <u>RETURN.</u> } { <u>EXIT PROGRAM</u> } <u>ENTER COBOL.</u>	<b>nombre-punto-entrada</b> es un nombre de 1 a 30 caracteres alfanuméricos, el primero de los cuales debe ser alfabético. Argumentos de enlace-1 y argumentos de enlace-2 ocupan un emplazamiento único en memoria central. La asimilación de ambos puede realizarse en cualquier parte con cláusulas <b>REDEFINES nombre-lenguaje</b> no se utiliza cuando ambos programas son COBOL.
IBM, sistemas 360 y 370	PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>CALL</u> nombre-punto de entrada <u>USING</u> lista de argumentos] <u>ENTER COBOL</u>	LINKAGE SECTION definición de argumentos PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>ENTER</u> nombre-punto-entrada [ <u>USING</u> lista de argumento ] <u>ENTER COBOL</u> ===== rutina común retorno al programa de llamada <u>ENTER LINKAGE</u> <u>RETURN</u> <u>ENTER COBOL</u>	

SISTEMA	En el programa de llamada	En los subprogramas	Observaciones
UNIVAC, sistema 1108	<p><b><u>ENTER</u></b> nombre-subprograma <b><u>SUBROUTINE</u></b></p> <p><b><u>REFERENCING</u></b> { literal-1                           nombre-dato-1                           nombre-fichero-1 }</p> <p>[ literal-2   nombre-dato-2   nombre-fichero-2 ]</p>	<p>COMMON-STORAGE SECTION definición de argumentos</p> <p>PROCEDURE DIVISION DECLARATIVES nombre-sección SECTION</p> <p><b><u>USE FOR ENTRY POINTS</u></b> nombre-procedimiento-1 [nombre-procedimiento-2]</p> <p>END DECLARATIVES</p> <p><b><u>GO TO</u></b> nombre-procedimiento-1</p> <p><b><u>PERFORM</u></b> nombre-procedimiento-2 [<b><u>THRU</u></b> nombre-procedimiento-3]</p> <p><b><u>EXECUTE</u></b> nombre-procedimiento-4 [<b><u>THRU</u></b> nombre-procedimiento-5]</p>	<p>nombre-procedimiento-1, ... siguen las reglas habituales de formación, pero el compilador sólo toma en cuenta los seis primeros caracteres. Por consiguiente, éstos no deben ser iguales. La orden EXECUTE es análoga a la orden PERFORM, pero un procedimiento utilizado en una orden PROCEDURE no puede ser utilizada en otra orden COBOL <b>nombre-subprograma</b>, debe estar formado por seis caracteres alfanuméricos, el primero de los cuales debe ser obligatoriamente alfabético.</p>
HW-BULL, sistema GE 100	<p>PROCEDURE DIVISION</p> <p><b><u>ENTER APS</u></b> [nombre rutina]</p> <p>===== (rutina en APS) =====</p> <p><b><u>ENTER COBOL</u></b></p>		<p>En este sistema el verbo ENTER permite introducir en COBOL instrucciones en otro lenguaje.</p> <p>nombre-rutina, si se utiliza, sólo suma efectos informativos.</p> <p>En la rutina APS no se pueden utilizar las instrucciones ENS e INS.</p> <p>Solamente pueden utilizarse las macroinstrucciones COLL y SUB.</p>

SISTEMA	En el programa de llamada	En los subprogramas	Observaciones
SIEMENS, sistema 4004	PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>CALL</u> nombre-punto-entrada <u>[USING</u> lista de argumentos...] <u>ENTER COBOL</u>	LINKAGE SECTION definición de argumentos PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>ENTRY</u> nombre-punto-entrada <u>[USING</u> lista de argumentos] <u>ENTER COBOL</u> ===== ===== (rutina común) ===== <div style="border: 1px solid black; padding: 2px; display: inline-block;">retorno al programa de llamada</div> <u>ENTER LINKAGE</u> <u>RETURN</u> <u>ENTER COBOL</u>	Los nombres de datos u otros argumentos especificados en una declaración CALL pueden ir calificados o afectados de subíndices.
CII, sistema IRIS-50	PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>CALL</u> nombre-punto-entrada <u>[USING</u> lista de argumentos...] <u>ENTER COBOL</u>	LINKAGE SECTION definición de argumentos PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>ENTRY</u> nombre-punto-entrada <u>[USING</u> lista de argumentos] <u>ENTER COBOL</u> ===== ===== (rutina común) ===== <div style="border: 1px solid black; padding: 2px; display: inline-block;">retorno al programa de llamada</div> <u>ENTER LINKAGE</u> <u>RETURN</u> <u>ENTER COBOL</u>	

SISTEMA	En el programa de llamada	En los subprogramas	Observaciones
UNIVAC 9200-9300	PROCEDURE DIVISION <u>ENTER LINKAGE</u> <u>CALL</u> nombre-punto-entrada <u>[ USING [argumento] ... ]</u> <u>ENTER COBOL</u>	(sólo BAL) nombre-punto-entrada <u>ENTRY</u> nombre-punto-entrada ===== (rutina) ===== ===== END	Solamente puede establecerse enlace entre un programa COBOL y un programa BAL (Basic Assembly Language).
HW-BULL, sistema GE-600	PROCEDURE DIVISION <u>ENTER LINKAGE</u> MODE <u>CALL</u> nombre-punto de entrada <u>[ USING lista argumentos ]</u> <u>ENTER COBOL</u>	PROCEDURE DIVISION <u>ENTER LINKAGE</u> MODE <u>ENTRY</u> POINT IS nombre-punto-entrada <u>[ USING lista argumentos-1 ]</u> <u>[ GIVING lista argumentos-2 ]</u> <u>ENTER COBOL</u> ===== (rutina común) ===== ===== EXIT	Lista argumentos-1 se refiere a los nombres de datos dados. Lista argumentos-2 se refiere a los nombres de datos de los resultados.



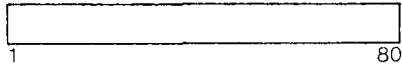
001010 IDENTIFICATION DIVISION.  
001020 PROGRAM-ID. 'PROGLLAM'  
001030 REMARKS. SE ILUSTRÁ, COMO EJEMPLO, LA UNIÓN ENTRE ESTE  
001040 PROGRAMA Y EL PROGRAMA DENOMINADO JOBPROGM CODIFICADO  
001110 EN LA PARTE INFERIOR DE ESTA SEPARATA.  
001120 ENVIRONMENT DIVISION.  
001130 CONFIGURATION SECTION.  
001140 SOURCE-COMPUTER. IBM-360 F40.  
001150 OBJECT-COMPUTER. IBM-360 F40.  
001160 INPUT-OUTPUT SECTION.  
001170 FILE-CONTROL.  
001180 SELECT FICHERO-A ASSIGN TO 'SYS004' UTILITY 2400 UNITS.  
001190 SELECT FICHERO-B ASSIGN TO 'SYS006' UNIT-RECORD 2540R  
001200 RESERVE NO ALTERNATE AREA.  
001210 DATA DIVISION.  
001220 FILE SECTION.  
001230 FD FICHERO-A DATA RECORD IS REGISTRO-A LABEL RECORD ARE  
001240 STANDARD BLOCK CONTAINS 5 RECORDS RECORDING MODE IS F.  
001250 01 REGISTRO-A.  
002010 02 CAMPO-1 PICTURE X(40).  
002020 02 CAMPO-2 PICTURE X(30).  
002030 02 FILLER PICTURE X(10).  
002040 FD FICHERO-B DATA RECORD IS REGISTRO-B LABEL RECORD IS  
002050 OMITTED.  
002060 01 REGISTRO-B PICTURE X(80)  
002070 PROCEDURE DIVISION.  
002080 APERTURA.  
002090 OPEN INPUT FICHERO-B OUTPUT FICHERO-A. _____  
002100 LECTURA.  
002110 READ FICHERO-B AT END GO TO FIN. _____  
002120 ENTER LINKAGE.  
002130 CALL 'SUBPROGR' USING REGISTR-C-B. _____  
002140 ENTER COBOL.  
002150 WRITE REGISTR-C-A FROM REGISTR-C-B. _____  
002160 GO TO LECTURA. _____  
002170 FIN.  
002180 CLOSE FICHERO-A FICHERO-B STOP RUN.

---

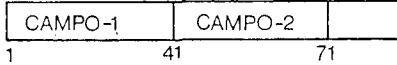
001010 IDENTIFICATION DIVISION.  
001020 PROGRAM-ID. 'SUBPROGM'.  
001030 REMARKS. ESTE PROGRAMA ES LLAMADO POR EL PROGRAMA PROGLLAM  
001040 UTILIZÁNDOSE COMO ÁREA DE ENLACE LA DEFINIDA EN  
001050 LINKAGE SECTION.  
001060 ENVIRONMENT DIVISION.  
001070 CONFIGURATION SECTION.  
001080 SOURCE-COMPUTER. IBM-360 F40.  
001090 OBJECT-COMPUTER. IBM-360 F40.  
001100 DATA DIVISION.  
001110 WORKING-STORAGE SECTION.  
001150 77 CAMBIO PICTURE X(30) VALUE 'ESTE REGISTRO SE HA MODIFICADO'  
001160 LINKAGE SECTION.  
001170 01 AREA-COMUN.  
001180 02 PARTE-1 PICTURE X(40).  
001190 02 PARTE-2 PICTURE X(30).  
001200 02 FILLER PICTURE X(10).  
001210 PROCEDURE DIVISION.  
001220 ENTRY 'SUBPROGR' USING AREA-COMUN. _____  
001230 ENTER COBOL.  
001240 MODIFICACION.  
001250 MOVE CAMBIO TO PARTE-2. _____  
002010 ENTER LINKAGE.  
002020 RETURN.

Este programa, principal, efectúa la lectura del fichero denominado FICHERO-B para generar otro fichero denominado FICHERO-A, cuyos registros tienen los siguientes formatos:

REGISTRO-B



REGISTRO-A



Se efectúa la apertura de ambos ficheros.

Se efectúa la lectura de un registro del fichero de entrada que se almacena en el área de entrada.

A continuación se procede a llamar al programa SUBPROGR, utilizándose como parámetro de enlace el propio registro-B o área de entrada. Se pasa a ejecutar la PROCEDURE de dicho programa

Se graba un registro del fichero de salida, después de hacer un movimiento desde el área de entrada (opción FROM).

Se repite todo el proceso con un nuevo registro.

En este programa existen dos áreas en la memoria central:  
En el área de trabajo, con el siguiente contenido:

CAMBIO ESTE REGISTRO SE HA MODIFICADO

AREA COMUN



Se establece la correspondencia de AREA-COMUN y REGISTRO-B (el mismo área de memoria es compartida por ambos campos)

Se almacena en el campo PARTE-2 de AREA-COMUN el contenido del campo CAMBIO y se devuelve el control al programa principal (esta operación ha permitido almacenar en el área de entrada del programa principal, en las posiciones 41 a 70, el contenido de PARTE-2).





# APENDICE B

CLASIFICACION INCLUIDA  
EN UN PROGRAMA COBOL





El objeto de un módulo de clasificación del lenguaje COBOL es el de permitir redactar en COBOL secuencias especiales de tratamiento durante el primer y último paso de una CLASIFICACION sin tener que utilizar explícitamente el sistema de enlace entre un programa COBOL y los programas escritos en otro lenguaje.

El COBOL introduce la noción de fichero intermedio de clasificación. Este fichero no se presenta nunca en realidad durante la ejecución del programa en la forma de un fichero COBOL ordinario, de hecho se le puede definir como el conjunto de registros tratados por la clasificación.

Sin embargo, y desde un punto de vista funcional, todo ocurre en un programa COBOL que llame a la clasificación como si se desarrollaran sucesivamente las tres operaciones siguientes:

- Grabación del fichero intermedio de clasificación.
- Clasificación de los registros de este fichero en el orden necesario y especificado.
- Lectura del fichero clasificado.

En la primera operación se utiliza la instrucción **RELEASE**, análoga al verbo **WRITE**; la última operación se realiza a través de la orden **RETURN**, análoga al verbo **READ**.

El conjunto de las tres operaciones lo gobierna el verbo de clasificación **SORT**.

El verbo **SORT** opera de forma distinta según que el programador desee o no constituir y explotar el fichero intermedio de clasificación.

En el caso de que todos los registros a clasificar se encuentren ya agrupados en un fichero único, el programador solamente tiene que indicar el nombre de fichero, que constituye la entrada de clasificación, por medio de la opción **USING**.

En los demás casos el programador puede seleccionar registros, incluso de diferentes ficheros, y conformar los registros a clasificar.

Las codificaciones obligatorias que se deben incluir dentro del programa COBOL para efectuar una clasificación, independientemente de las propias del programa, son las siguientes:

ENVIRONMENT DIVISION.  
FILE-CONTROL.

**SELECT** nombre-fichero-intermedio de-clasificación .....

DATA DIVISION.

**SD** nombre-fichero **DATA** { RECORD IS  
RECORDS ARE } nombre-registro-1

[nombre-registro-2] ...

**inscripciones de descripción de registro-1 [registro-2] .....**

PROCEDURE DIVISION.

_____  
_____  
_____

**RELEASE** nombre-de-registro **FROM** [nombre-dato]

Esta instrucción proporciona los registros, designados por nombre de registro, a la fase inicial de la clasificación. Solamente puede emplearse dentro del procedimiento abarcado por la opción INPUT PROCEDURE de la orden SORT.

La opción FROM opera de forma análoga a la misma opción del verbo WRITE. Después de la ejecución del procedimiento en el cual está contenida la orden RELEASE se ha generado el fichero intermedio de clasificación y se puede ejecutar la orden SORT, cuyo formato es:

**SORT** nombre-fichero ON { DESCENDING  
ASCENDING }

KEY nombre-dato-1 [nombre-dato-2] ...

[ ON { DESCENDING  
ASCENDING } KEY nombre-dato-3 [nombre-dato-4] ... ]...

{ **INPUT PROCEDURE** IS nombre-procedimiento-1 [ **THRU**  
USING nombre-fichero-1

nombre-procedimiento-2] }

{ **OUTPUT PROCEDURE** IS nombre-procedimiento-3 [ **THRU**  
GIVING nombre-fichero-2

nombre-procedimiento-4] }

Nombre-fichero, debe ser el nombre establecido en la inscripción SD de la DATA DIVISION.

Nombre-dato-1, ... deben ser campos de datos de los registros que constituyen nombre-fichero.

Nombre-procedimiento-1, ... designan los procedimientos a establecer en la fase previa a la clasificación y nombre-procedimiento-3, ... los procedimientos a ejecutar después de efectuar la clasificación de los registros del fichero intermedio. Todos los nombres de procedimientos deben ser nombres de sección.

La operación de clasificar consiste en ordenar los registros del fichero intermedio según los valores de los campos definidos por nombre-dato-1, ...

La ordenación según los valores indicados, puede realizarse para cada campo en orden a su valor creciente o ascendente o a su valor decreciente o descendente. Se utilizan las palabras reservadas ASCENDING o DESCENDING, respectivamente, para cada necesidad.

Por ejemplo: la orden ASCENDING CAMPO-1 DESCENDING CAMPO-2 ASCENDING CAMPO-3 CAMPO-4 indicará que la clasificación del fichero se realiza en primer lugar según los valores ascendentes de CAMPO-1; a igualdad de éstos, para valores descendentes de CAMPO-2; a igualdad de valores de CAMPO-1 y CAMPO-2, para valores ascendentes de CAMPO-4, y, por último, a igualdad de todos los anteriores, para valores ascendentes de CAMPO-4.

REG. NUM.	Contenido de			
	CAMPO-1	CAMPO-2	CAMPO-3	CAMPO-4
1	1	2	4	5
2	1	2	3	7
3	1	3	2	1
4	1	3	2	3
5	1	2	4	6
6	2	4	1	6
7	1	2	4	9
8	1	3	5	8
9	1	2	4	3

Suponiendo que el contenido de CAMPO-1, CAMPO-2, CAMPO-3 y CAMPO-4 sea el figurado en el cuadro anterior, el resultado de la aplicación de la opción anteriormente comentada daría el siguiente orden de clasificación de los registros:

- 1.º) Registro número 3.
- 2.º) » » 4.
- 3.º) » » 8.
- 4.º) » » 2.
- 5.º) » » 9.
- 6.º) » » 1.
- 7.º) » » 5.
- 8.º) » » 7.
- 9.º) » » 6.

USING nombre-fichero-1 especifica que el fichero a clasificar se crea automáticamente a partir de un fichero definido en la DATA en una inscripción a nivel FD nombre-fichero-1. En caso de desearse una preselección de determinados registros y/o selección de determinados campos, es necesario establecer un procedimiento, el cual se especifica por INPUT PROCEDURE ... interviene la orden RELEASE estudiada anteriormente.

GIVING nombre-fichero-2, especifica que el fichero intermedio ya clasificado queda disponible para su tratamiento en el área definida por nombre-fichero-2, cuyas inscripciones se deben haber realizado a nivel FD nombre-fichero-2. En caso de desearse una preselección de determinados registros y/o selección de determinados campos del fichero designado, es necesario establecer un procedimiento, el cual se especifica por OUTPUT PROCEDURE ... En este caso debe intervenir la orden RETURN, cuyo formato es:

RETURN nombre-fichero-intermedio RECORD INTO nombre-dato  
AT END declaración-imperativa.

Esta orden RETURN es totalmente análoga en su función a la orden READ. La declaración imperativa del formato se ejecuta cuando el fichero intermedio de clasificación se ha leído totalmente.

---

# APENDICE C

ANSI COBOL

---





El American National Standards Institute (ANSI), anteriormente denominado USA Standard Institute (USASI) ha sido el responsable de establecer un COBOL estandarizado para los usuarios USA. El objetivo de esta revisión del COBOL ha sido el de conseguir una total estandarización y eliminar de esta forma las pequeñas o grandes diferencias existentes entre los compiladores. Este COBOL, totalmente estandarizado, se denomina COBOL ANSI.

La estructura básica del COBOL no ha variado en la versión ANSI. Existen, sin embargo, diferencias mínimas entre algunas opciones de las cláusulas estudiadas en el texto, que no incluimos en este apéndice.

En cambio exponemos las diferencias más importantes que afectan a cada una de las Divisiones.

<b>IDENTIFICATION DIVISION</b>	
<p><b><u>VERSION COBOL no ANSI</u></b> PROGRAM-ID. Gran variedad para cada uno de los compiladores.</p> <p>DATE-COMPILED, debe indicarse alguna información.</p>	<p><b><u>COBOL-ANSI</u></b> PROGRAM-ID</p> <ul style="list-style-type: none"> <li>— No debe incluirse el nombre del programa entre comillas.</li> <li>— Se debe formar como un nombre de dato.</li> <li>— Se toman en consideración solamente los ocho primeros escritores.</li> </ul> <p>DATE-COMPILED. Se debe omitir cualquier información. El compilador incluye la fecha de la compilación dentro del listado fuente.</p>
<b>ENVIRONMENT DIVISION</b>	
<p><b><u>VERSION COBOL no ANSI</u></b> Los párrafos SOURCE-COMPUTER y OBJECT-COMPUTER incluyen el tipo y modelo separados por un guión (el modelo puede ser optativo, pero debe incluirse el guión).</p> <p>Clausula SELECT</p> <p>Son variables las opciones para cada compilador.</p>	<p><b><u>COBOL-ANSI</u></b> Solamente es obligatorio indicar el tipo de máquina y no el modelo.</p> <p>El formato de la clausula SELECT es SELECT nombre-fichero ASSIGN TO número sistema-clase dispositivo-número de dispositivo-modo de acceso; es decir, debe incluirse un guión entre cada especificación; el número de sistema no debe incluirse entre comillas; la clase de dispositivos debe ser: UR, registro unitario; UT, utilidad; DA, acceso directo. El modo de acceso debe ser: S para secuencial; D para directo, e I para secuencial-indexado. No existe la palabra UNIT de los compiladores no ANSI.</p>

## DATA DIVISION

### VERSION COBOL no ANSI

Cláusula RECORDING MODE. En muchos compiladores puede ser una cláusula optativa. Su omisión supone modo de registro variable.

### COBOL-ANSI

Cláusula RECORDING MODE. Es siempre una cláusula opcional. Su omisión indica modo de registro fijo.

## PROCEDURE DIVISION

### VERSION COBOL no ANSI

#### Expresiones condicionales compuestas

No pueden utilizarse en todos los compiladores sujetos implícitos.

#### Declaración IF

No admiten todos los compiladores el uso de las palabras reservadas NEXT SENTENCE detrás de la opción ELSE.

#### Verbos

No se admiten varios campos receptores con la opción TO en algunos compiladores. No se admite la palabra BY en el formato del verbo DIVIDE de algunos compiladores.

#### Verbo ALTER

La palabra PROCEED es clave.

#### Verbo WRITE

No es válida en todos los compiladores la opción BEFORE.

#### AFTER ADVANCING Ø LINES

Produce un salto a nueva página.

Se utiliza la cláusula APPLY FORM-OVER FLOW en algunos compiladores para el control de fin de página.

### COBOL-ANSI

#### Expresiones condicionales compuestas

Es válido siempre el uso de operadores implícitos. Es correcto, pues:

IF A > 4 OR < 3 AND > — 5

#### Declaración IF

Siempre es válido el uso de NEXT SENTENCE detrás de ELSE.

#### Verbos

Se admiten siempre varios campos receptores de la opción TO.

Se admite siempre la opción BY del formato del verbo DIVIDE.

#### Verbo ALTER

La palabra PROCEED es optativa.

#### Verbo WRITE

Son válidas las opciones BEFORE y AFTER.

No está permitido el uso del número entero Ø para saltar a nueva página. Debe utilizarse un nombre de dato para expresar el salto a nueva página a través de un nombre nemotécnico en SPECIAL-NAMES.

No está permitido el uso de la cláusula APPLY FORM-OVER FLOW.

Queda sustituida por una opción nueva AT 

END-OF-PAGE
EOP

 declaración-imperativa en el formato del verbo WRITE.

El uso de esta opción obliga a utilizar la cláusula RESERVE NO ALTERNATE AREA para el fichero especificado de salida.

<p>VERBO STOP RUN</p> <p><b><u>FICHEROS EN DISCO</u></b></p> <p>Se utiliza la cláusula ORGANIZACION para indicar el tipo de organización de un fichero en disco.</p> <p>Existe la cláusula SYMBOLIC KEY.</p>	<p>Queda sustituido por GOBAG.</p> <p><b><u>FICHEROS EN DISCO</u></b></p> <p>Se elimina esta cláusula y se incluye una nueva opción en la cláusula</p> <p>ASSIGN { I } I significa secuencial-indexado                { S } S significa secuencial                { D } D significa directa</p> <p>Se sustituye por la cláusula NOMINAL KEY.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



---

# APENDICE D

BIBLIOTECA DE PROGRAMAS  
FUENTE

---



Como se indicaba en el epígrafe I-2.1.1.6., muchas veces en un programa es necesario escribir determinadas secuencias de instrucciones idénticas a las escritas en otro programa. A fin de evitar, al programador, la escritura de estas secuencias es posible incluirlas en un fichero especial del sistema, denominado LIBRERIA e insertarlas en el nuevo programa a partir de dicho fichero. Para poderlo realizar basta con haber asignado a cada una de las secuencias que se vayan a insertar, un nombre simbólico que se denomina nombre-biblioteca.

La extracción de estas rutinas o secuencias de instrucciones de la biblioteca, para incluirlas en el programa fuente, se realiza a través de los verbos COPY e INCLUDE.

Estos verbos son tenidos en cuenta por el compilador en el momento de la compilación y permiten insertar en el entorno en donde se encuentra uno u otro de estos verbos, una o varias instrucciones del fichero especial LIBRERIA.

COPY puede ser utilizado para ciertos párrafos de la ENVIRONMENT DIVISION y para descripciones de ficheros y registros de la DATA DIVISION.

El verbo INCLUDE permite insertar grupos de instrucciones en la PROCEDURE DIVISION. La escritura de estos verbos varía de un compilador a otro, aunque la más generalizada es

*COPY* nombre-biblioteca.

*INCLUDE* nombre-biblioteca

Nombre-biblioteca permite encontrar sobre el fichero LIBRERIA la secuencia solicitada.

Los lugares donde debe incluirse el verbo COPY son:

Dentro de INPUT-OUTPUT SECTION

{ FILE-CONTROL }  
{ I-O-CONTROL } COPY nombre-biblioteca

Dentro de FILE-CONTROL

SELECT nombre-fichero COPY nombre-biblioteca.

Dentro de una inscripción de descripción de registro o dentro de la WORKING-STORAGE o LINKAGE SECTION.

01 nombre-dato COPY nombre-biblioteca.

Dentro de la WORKING-STORAGE o LINKAGE SECTION

77 nombre-dato COPY nombre-biblioteca.

Dentro de FILE SECTION

FD nombre-fichero COPY nombre-biblioteca.





---

# APENDICE E

MENSAJES DE ERROR DE UN  
COMPILADOR COBOL

---



## DIAGNOSTICOS DE ERRORES DETECTADOS POR EL COMPILADOR

Se incluye en este apéndice, y como ejemplo, los diagnósticos de errores que detecta el compilador COBOL del sistema UNIVAC 9200/9300 (1). El incluir en esta obra los mensajes del compilador COBOL de los restantes sistemas comentados en el texto no conduciría sino a aumentar innecesariamente el volumen de páginas del mismo. Se ha elegido como muestra el compilador mencionado por ser uno de los sistemas que los alumnos del Instituto de Informática pueden utilizar en los ejercicios prácticos.

El compilador COBOL del sistema UNIVAC 9200/9300 comprueba si la codificación del programa fuente es inadecuada, incompatible o dudosa, produciéndose en tal caso una línea de diagnóstico al imprimirse el listado del programa fuente.

El formato de los diagnósticos del listado fuente consta de un *número de línea* de cinco dígitos y un *número de diagnóstico* de tres dígitos.

El número de línea va asociado a cada línea del programa fuente y se imprime a la izquierda de la misma en el listado del programa fuente. En algunos casos el número de línea del diagnóstico puede ser diferente al número de línea de la codificación aplicable; por ejemplo, si un verbo y uno de sus operandos están en líneas diferentes y el operando es incorrecto se incluye en el diagnóstico el número de línea para el verbo y no para el operando.

Existen tres tipos de diagnósticos:

- P, que indica precaución. Indica que el código fuente se desvía del formato ordinario y puede ser invalidado. El diagnóstico se imprime como aviso, pero la compilación continúa y la codificación se procesa.
- C, que indica error corregible. Se escribe este mensaje cuando el compilador detecta la omisión de un carácter, palabra, cláusula o proposición o cuando se han empleado incorrectamente. Para compensar tal error se hace una suposición razonable en el compilador. La compilación continúa y la codificación se procesa; sin embargo, se pueden producir otros errores si la suposición no es la adecuada.
- U, que indica error incorregible. Se emite este mensaje cuando el compilador detecta la omisión de un carácter, palabra, cláusula o preposición o cuando se han empleado incorrectamente, pero a diferencia del error C el compilador no puede hacer ninguna suposición razonable. La compilación continúa, pero no se procesa la codificación incorrecta o incompatible. Para corregir el programa debe intervenir el programador y procederse a una nueva compilación.

En algunos casos se imprimen varios diagnósticos para la misma codificación fuente. Esto se debe a que diferentes partes del compilador ejecutan generalmente pruebas afines sobre la codificación, incluso después de que se haya visto que ésta es inadecuada, incompatible o dudo-

---

(1) Los conceptos de este apéndice están tomados del manual de programación COBOL UPE-7543. Rev. 3 de UNIVAC para los sistemas 9200 II/9300/9300 II.

sa. También muchas declaraciones aparentemente básicas se descomponen en el compilador en varias más básicas. Por ejemplo, la declaración `ADD A TO B` se procesa como dos declaraciones: `ADD A, B GIVING SUM` y `MOVE SUM TO B`. Si `B` no está definida o está definida incorrectamente, se imprimen dos diagnósticos idénticos para una sola sentencia fuente.

En los siguientes cuadros se incluye un catálogo de los diagnósticos, dispuestos por orden ascendente de número de diagnóstico para mayor facilidad de localización. Para cada número de diagnóstico el catálogo contiene el tipo de diagnóstico, una breve explicación y la causa del error.

## DIAGNOSTICOS COMUNES A TODAS LAS DIVISIONES

N.º	Tipo	Mensaje (traducción) y explicación del error
001	P	<p><b>MISSING HEADER OR INVALID SINTAX FOLLOWING A PERIOD</b> (Cabecera omitida o sintaxis inválida a continuación de un punto)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Falta una cabecera de división o de sección. Si falta la cabecera de DATA DIVISION o la de PROCEDURE DIVISION no se procesa la división correspondiente. La compilación no queda afectada si falta cualquiera otra cabecera de división o de sección.</li> <li>(2) La sintaxis que sigue a un punto no es válida. Se salta la palabra, así como las que le sigan hasta que se detecte la siguiente palabra reservada válida o nombre-de-procedimiento.</li> </ol>
002	U	<p><b>INVALID SINTAX</b> (Sintesis inválida)</p> <p>Se detecta una de las siguientes condiciones y se salta la proposición o cláusula:</p> <ol style="list-style-type: none"> <li>(1) Falta o no es válida la primera palabra requerida de una frase opcional.</li> <li>(2) Falta un punto necesario para la sintaxis de una proposición o cláusula.</li> </ol>
003	U	<p><b>MISSING OR MISSPELLED RESERVED WORD</b> (Palabra reservada omitida o mal escrita)</p> <p>Se ha saltado la palabra. Si la palabra es clave, se salta toda la proposición.</p>
004	P	<p><b>CARD SEQUENCE ERROR</b> (Error de secuencia en las fichas fuente)</p> <p>El número de secuencia de una ficha (columnas 1 a 6) va en orden descendente o falta. Se ignora esta condición y se acepta la ficha.</p>
005	U	<p><b>ERRONEUS CHARACTER</b> (Carácter erróneo)</p> <p>Se ha detectado un carácter que no forma parte del repertorio de caracteres de COBOL. Se salta y se reduce la serie de caracteres. Ejemplo: el nombre-de-dato AB%C. De esta condición resulta también un diagnóstico 013. El carácter erróneo puede ser también un carácter no imprimible y no se detectará en el listado fuente.</p>
006	C	<p><b>EXCESIVE CHARACTER-STRING LENGTH</b> (Longitud excesiva en una cadena de caracteres)</p> <p>Un literal no-numérico tiene más de 132 caracteres, o cualquier otra serie de caracteres tiene más de 30. Los caracteres de exceso se saltan hasta que se detecta el primer apóstrofo (o comillas), punto o espacio.</p>

## DIAGNOSTICOS COMUNES A TODAS LAS DIVISIONES

N.º	Tipo	Mensaje (traducción) y explicación del error
007	C	<p><b>SPACE FOLLOWS LEFT PARENTHESIS</b> <i>(Espacios a continuación de un paréntesis izquierdo)</i></p> <p>Se saltan los espacios.</p>
008	C	<p><b>INVALID CONTINUATION OF NONNUMERIC LITERAL</b> <i>(Continuación errónea de un literal no-numérico)</i></p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Un literal no-numérico que se está continuando (guión en la columna 7 de la ficha de continuación) no empieza con un apóstrofo o comillas. Se acepta como continuación del literal no-numérico que empieza con el primer carácter no blanco.</li> <li>(2) Un literal no-numérico parece haber terminado en una ficha porque su apóstrofo (o comillas) de final va seguido por un punto, coma o espacio; sin embargo, la ficha siguiente contiene un guión en la columna 7, indicando continuación. Se ignora la notación de continuación.</li> </ol>
009	C	<p><b>FORMAT ERROR IN COLUMNS 7-11</b> <i>(Error de formato en las columnas 7-11)</i></p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) La columna 7 no contiene guión, asterisco o espacio. El carácter de la columna 7 se acepta como parte de la serie de caracteres, bajo la suposición de que la serie comenzó erróneamente a la izquierda de la ficha fuente.</li> <li>(2) Se continúa una serie de caracteres en las columnas 8 a 11, en lugar de 12 a 72. Se aceptan los caracteres de continuación.</li> </ol>
010	C	<p><b>INVALID BEGINNING OR ENDING OF NONNUMERIC LITERAL</b> <i>(Comienzo o fin inválido de un literal no-numérico)</i></p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Hay un carácter no blanco precediendo a un apóstrofo de comienzo (o comillas). Se supone espacio.</li> <li>(2) El apóstrofo (o comillas) de final va seguido por un carácter distinto de punto, coma o espacio; o el punto o la coma no van seguidos de espacio. Se acepta el literal como si fuese seguido de punto y espacio o de coma y espacio.</li> </ol> <p>A uno de los siguientes caracteres especiales no le sigue espacio: &lt; = &gt; /. Se ignora el carácter especial.</p>
011	U	<p><b>ERRONEUS CHARACTER-STRING</b> <i>(Cadena de caracteres errónea)</i></p> <p>Una serie de caracteres es errónea porque no sigue las normas de COBOL para palabras, literales o «pictures» (imágenes). Se ignora la serie.</p>

## DIAGNOSTICOS COMUNES A TODAS LAS DIVISIONES

N.º	Tipo	Mensaje (traducción) y explicación del error
012	C	<p><b>SPECIAL CHARACTER NOT PRECEDED BY SPACE</b> (Carácter especial no precedido de un espacio)</p> <p>Uno de los siguientes caracteres especiales: &lt; = &gt; / no va precedido de espacio. Se ignora el carácter especial.</p>
013	P	<p><b>COMMA IN NUMERIC LITERAL</b> (Como ortográfica en un literal numérico)</p> <p>Se suprimen las comas y se reduce el literal.</p>
014	U	<p><b>MISSING OR INVALID STATUS SWITCH CONDICION NAME</b> (Nombre de condición con switch de estado omitido o inválido)</p> <p>Una cláusula ON STATUS u OFF STATUS no contienen en nombre-de-condición válido. La cláusula se salta.</p>
015	U	<p><b>MISSING OR INVALID STATUS SWITCH NMEMONIC-NAME</b> (Nombre nemotécnico con switch de estado omitido o inválido)</p> <p>Esta cláusula se salta.</p>
016	U	<p><b>MISSING OR INVALID ON-OFF STATUS CLAUSE</b> (Cláusula de estado on-off omitido o inválido)</p> <p>La proposición SYSSWCH no tiene una cláusula OFF ni ON STATUS. La proposición se salta.</p>
017	P	<p><b>SYSSWCH WITH MORE THAN ONE ON STATUS OR OFF STANS</b> (SYSSWCH con más de un estado ON u OF)</p> <p>Se acepta la primera situación de STATUS. Las siguientes se saltan.</p>
018	U	<p><b>INVALID SYSSWCH INTEGER</b> (Número entero de SYSSWCH inválido)</p> <p>El número SYSSWCH encontrado no está dentro del ámbito 0 a 7. Toda la proposición SYSSWCH se salta.</p>

## DIAGNOSTICOS DE LA IDENTIFICATION y ENVIRONMENT DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
019	C	<p><b>MISSING OR INVALID PROGRAM-ID</b> (PROGRAM-ID omitido o inválido)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Falta la palabra PROGRAM-ID. Se asume dicha ficha.</li> <li>(2) Falta el nombre-de-programa. Se asigna NOTNAMED como nombre-de-programa.</li> <li>(3) El nombre-de-programa pasa de los 30 caracteres. Se utilizan los ocho primeros caracteres; el resto de los caracteres se trunca.</li> <li>(4) El nombre-de-programa contiene caracteres que no son ni alfabéticos ni numéricos. Se asigna el nombre-de-programa NOTNAMED.</li> </ol>
020	C	<p><b>SOURCE OR OBJECT COMPUTER NOT UNIVAC 9200/9300</b> (Ordenador fuente u objeto no UNIVAC 9200/9300)</p> <p>Se asume UNIVAC 9200 II/9300/9300 II como ordenador fuente u objeto.</p>
021	C	<p><b>INVALID MEMORY SIZE CLAUSE</b> (Cláusula MEMORY SIZE inválida)</p> <p>Se asigna un tamaño de memoria de 32.768 octetos al ordenador objeto.</p>
022	U	<p><b>INVALID CURRENCY SIGN CHARACTER</b> (Carácter de la cláusula CURRENCY SIGN inválido)</p> <p>Se salta la cláusula CURRENCY SIGN.</p>
023	U	<p><b>MISSING OR INVALID SELECT OR RERUN FILE-NAME</b> (Nombre de fichero en cláusula SELECT o RERUN omitida o inválida)</p> <p>En una SELECT o RERUN no se especifica un nombre-de-fichero o se describe erróneamente. La anotación se salta.</p>
024	U	<p><b>INCORRECT HARDWARE-NAME OR INVALID HARDWARE DESIGNATION</b> (Nombre de hardware incorrecto o inválido)</p> <p>Se salta la anotación.</p>
025	U	<p><b>MISSING ASSIGN CLAUSE</b> (Cláusula ASSIGN omitida)</p> <p>No se asocia ninguna cláusula ASSIGN con una proposición SELECT. La proposición se salta.</p>



## DIAGNOSTICOS DE LA IDENTIFICATION y ENVIRONMENT DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
026	J	<p><b>INVALID ALTERNATE AREA CLAUSE</b> (Cláusula ALTERNATE AREA inválida)</p> <p>Para un archivo que no es de cinta se especifica una cláusula ALTERNATE AREA. Se salta la cláusula.</p>
027	U	<p><b>INVALID RESERVE ALTERNATE AREA CLAUSE</b> (Cláusula RESERVE ALTERNATE AREA inválida)</p> <p>Se emplea una cláusula RESERVE ALTERNATE AREA, pero no se especifica ninguna de las áreas alternativas. La cláusula se salta.</p>
028	U	<p><b>MISSING OR INVALID RERUN CLAUSE</b> (Cláusula RERUN omitida o inválida)</p> <p>Falta el entero de una cláusula RERUN. Se salta la cláusula.</p>
029	U	<p><b>MISSING OR INVALID SAME AREA FILE-NAME</b> (Nombre-fichero de una SAME AREA omitido o inválido)</p> <p>Una cláusula SAME AREA no contiene dos o tres nombres de archivo válidos. Se salta la cláusula.</p>
030	U	<p><b>EXCESSIVE SELECT ENTRIES</b> (Número de inscripciones SELECT excesivas)</p> <p>El número de anotaciones SELECT excede de los límites del compilador. Se saltan las anotaciones sobrantes.</p>
031	U	<p><b>FILE-NAME HAS NO VALID SELECT ENTRY</b> (Un nombre de fichero tiene una inscripción SELECT no válida)</p> <p>En una proposición SELECT correcta no está definido un nombre-de-archivo referenciado por una anotación SAME AREA, RERUN o FD. Se salta la anotación correspondiente.</p>
032	U	<p><b>FILE-NAME SELECTED MORE THAN ONCE</b> (Nombre de fichero referenciado por más de una cláusula)</p> <p>El último nombre-de-fichero es referenciado por más de una cláusula SELECT. Se acepta la primera cláusula SELECT; las demás se saltan.</p>
033	C	<p><b>INVALID LOGICAL UNIT</b> (Unidad lógica inválida)</p> <p>A un nombre-de-dispositivo se le asigna un código no válido de unidad lógica. El código se pone a cero para todos los dispositivos, excepto para cinta; el código de unidad lógica para cinta se pone a 1.</p>

## DIAGNOSTICOS DE LA IDENTIFICATION y ENVIRONMENT DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
034	C	<p><b>INVALID PUNCH CODE</b> (Código de perforación inválido)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Petición de perforadora a filas para perforar un número de columnas que no sea 80. El número de columnas a perforar se fija en 80.</li> <li>(2) Petición de perforadora normal para perforar menos de 2 y más de 80 columnas. El número de columnas a perforar se fija en 80.</li> <li>(3) Se pide la perforadora normal para perforar un número impar de columnas. El número de columnas a perforar se incrementa en 1.</li> </ol>
035	U	<p><b>INVALID RERUN</b> (Cláusula RERUN inválida)</p> <p>El nombre-de-dispositivo especificado en una cláusula RERUN no es TAPE. Se salta la cláusula RERUN.</p>
036	C	<p><b>DUPLICATE ENVIRONMENT DIVISION CLAUSE</b> (Cláusulas duplicadas en la ENVIRONMENT DIVISION)</p> <p>Una proposición de la ENVIRONMENT DIVISION contiene cláusulas duplicadas. Se acepta la primera cláusula.</p>
037	C	<p><b>INVALID SELECT STATEMENT</b> (Inscripción SELECT inválida)</p> <p>Una proposición SELECT contiene una sintaxis irreconocible o le falta un punto. Se procesa la información anterior al punto de la sintaxis irreconocible o al punto que falta. El resto de la información se salta hasta que se encuentre el siguiente punto o cabecera de división.</p>
038	P	<p><b>INCONSISTENT CHANNEL NUMBER IN SPECIAL-NAMES CLAUSE</b> (Número de canal incompatible en una cláusula SPECIAL-NAMES)</p> <p>Aparece un número de canal incompatible en la cláusula SPECIAL-NAMES. Se supone SYSCHAN-n.</p>
039	P	<p><b>INVALID LITERAL IN SELECT STATEMENT</b> (Literal inválido en una inscripción SELECT)</p> <p>Una proposición SELECT contiene un literal no ANSI después del nombre-de-implementador. El COBOL 9200/9300 requiere y procesa el literal.</p>
040	U	<p><b>MISSING RECORD KEY OR ACTUAL KEY CLAUSE</b> (Omisión de cláusula RECORD KEY o ACTUAL KEY)</p> <p>Un archivo definido como ORGANIZATION IS INDEXED no tiene una cláusula RECORD KEY y/o ACTUAL KEY asociada.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
041	U	<p><b>FD ENTRY NOT IN FILE SECTION</b> (Inscripción FD fuera de la FILE SECTION)</p> <p>Se ha encontrado una anotación FD en sitio distinto a la FILE SECTION. Se salta la anotación FD.</p>
042	C	<p><b>INCONSISTENT FD ENTRY</b> (Inscripción FD incompatible)</p> <p>Un fichero no asignado a cinta contiene, en la anotación FD, una cláusula RECORDING MODE o LABEL RECORDS, que se salta.</p>
043	U	<p><b>INVALID DATA RECORDS CLAUSE</b> (Cláusula DATA RECORDS inválida)</p> <p>El nombre-de-dato de la cláusula DATA RECORDS no es válido. Se salta toda la anotación FD, y todas las descripciones de registro subordinadas se atribuyen a la anotación FD válida anterior.</p>
044	U	<p><b>INVALID BLOCK CONTAINS CLAUSE</b> (Cláusula BLOCK CONTAINS inválida)</p> <p>En una cláusula BLOCK CONTAINS no se incluye un literal numérico o el valor del literal numérico pasa de 4096. Se salta toda la anotación FD y todas las descripciones de registro subordinadas se atribuyen a la anotación FD válida anterior.</p>
045	C	<p><b>INVALID BLOCK LENGTH</b> (Longitud de bloque inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) La cláusula BLOCK CONTAINS especifica menos caracteres que una descripción de registro del archivo. La longitud del mayor registro se utiliza para calcular la longitud del bloque.</li> <li>(2) El tamaño del bloque se ha calculado para ser mayor de 4096 para cintas o 3625 para discos. La longitud se fija en 4096 para cintas o 3625 para discos.</li> </ol>
046	C	<p><b>UNEQUAL RECORD LENGTHS</b> (Longitud de registro diferente)</p> <p>Los registros de un archivo que no es de cinta o que, siéndolo, lleva RECORDING MODE es F, son de longitud desigual. Para calcular la longitud del bloque se utiliza la del registro mayor.</p>
047	C	<p><b>INVALID BLOCK CONTAINS</b> (BLOCK CONTAINS inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Un archivo que no es de cinta o discos tiene una cláusula BLOCK CONTAINS, que se salta.</li> <li>(2) Una cláusula BLOCK CONTAINS especifica una longitud de bloque que excede de 4096 octetos para cintas o 3625 para discos. La longitud se fija en 4096 para cintas o 3625 para discos.</li> </ol>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
048	U	<p><b>SAME AREA CONFLICT</b> (Conflicto en una cláusula SAME AREA)</p> <p>La memoria asignada a este archivo es de diferente longitud que la asignada al primer archivo que utilice la misma área. Si la longitud de este archivo es mayor, los resultados son imprevisibles; si la longitud es menor, este diagnóstico es simplemente de precaución.</p>
049	C	<p><b>INVALID STANDARD LABEL INFORMATION</b> (Información de etiqueta estándar inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) Falta un parámetro de etiqueta estándar. El parámetro se pone a un valor asignado.</li> <li>(2) El valor asignado a uno de los parámetros de etiqueta estándar es excesivo. El parámetro se trunca.</li> </ol>
050	C	<p><b>DUPLICATE STANDARD LABEL PARAMETER</b> (Parámetro de etiqueta estándar duplicado)</p> <p>Las etiquetas de fichero utilizadas están duplicadas. La compilación continúa y se utiliza la información facilitada para la primera etiqueta.</p>
051	C	<p><b>UNDEFINED STANDARD LABEL PARAMETER</b> (Parámetro de etiqueta estándar indefinido)</p> <p>Las definiciones de etiquetas de fichero son incompletas. La compilación continúa sin el parámetro de etiquetas estándar.</p>
052	U	<p><b>MISSING OR INVALID VALUE OR FILE-ID CLAUSE</b> (Valor de cláusula FILE-ID omitido o inválido)</p> <p>FILE-ID no está descrito como nombre-de-dato, literal o constante figurativa. Se salta toda la anotación FD y todas las descripciones de registro subordinadas se atribuyen a la anterior anotación FD válida.</p>
053	U	<p><b>INVALID CREATION-DATE CLAUSE</b> (Cláusula CREATION-DATE inválida)</p> <p>CREATION-DATE no está descrita como nombre-de-dato, literal o constante figurativa. Se salta toda la anotación FD y sus descripciones de registro se atribuyen a la anterior FD válida.</p>
054	U	<p><b>INVALID EXPIRATION-DATE CLAUSE</b> (Cláusula EXPIRATION-DATE inválida)</p> <p>EXPIRATION-DATE no está descrita como nombre-de-dato, literal o constante figurativa. Se salta toda la anotación FD y todas las descripciones de registro subordinadas se atribuyen a la anterior FD válida.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
055	U	<p><b>INVALID GENERATION-NO CLAUSE</b> (Cláusula <i>GENERATION-NO</i> inválida)</p> <p>GENERATION-NO no está descrita como nombre-de-dato, literal o constante figurativa. Se salta toda la anotación FD y las descripciones de registro subordinadas se atribuyen a la anterior FD válida.</p>
056	U	<p><b>INVALID VOLUME-NO CLAUSE</b> (Cláusula <i>VOLUME-NO</i> inválida)</p> <p>VOLUME-NO no está descrita como nombre-de-dato, literal o constante figurativa. Se salta toda la anotación FD y las descripciones de registro subordinadas se atribuyen a la anterior FD válida.</p>
057	U	<p><b>INVALID LABEL RECORDS CLAUSE</b> (Cláusula <i>LABEL RECORDS</i> inválida)</p> <p>LABEL RECORDS no se describe como STANDARD u OMITTED. Se salta la cláusula LABEL RECORDS y se supone STANDARD para cintas o discos.</p>
058	U	<p><b>INVALID BLANK WHEN ZERO ITEM</b> (Campo de <i>BLANK WHEN ZERO</i> inválido)</p> <p>Un campo con la cláusula BLANK WHEN ZERO pasa de 30 caracteres. Se salta la cláusula BLANK WHEN ZERO.</p>
059	U	<p><b>INVALID BLANK WHEN ZERO CLAUSE</b> (Cláusula <i>BLANK WHEN ZERO</i> inválida)</p> <p>Falta la palabra ZERO en la cláusula BLANK WHEN ZERO. Se salta toda la descripción de datos.</p>
060	U	<p><b>LIMIT EXCEEDED FOR ITEM DESCRIPTIONS SUBORDINATE TO 01 LEVEL</b> (Límite excedido para descripciones de campos subordinados a un nivel 01)</p> <p>Se ha excedido el límite para las anotaciones de descripción de datos subordinados a un campo de nivel 01. Se saltan todos los siguientes nombres-de-dato subordinados.</p>
061	U	<p><b>MISSING FILE-NAME</b> (Nombre de fichero omitido)</p> <p>Una anotación FD no contiene nombre-de-fichero. La anotación FD se salta y todas las descripciones de registro subordinadas se atribuyen a la anterior FD válida.</p>
062	U	<p><b>MISSING RECORD DESCRIPTION</b> (Descripción de registro omitida)</p> <p>Una anotación FD no va seguida de las anotaciones de descripción de datos. Se salta toda la anotación FD y todas las descripciones de registro subordinadas se atribuyen a la anterior anotación FD válida.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
063	U	<p><b>MISSING DATA-NAME</b> <i>(Nombre de dato omitido)</i></p> <p>Un número-de-nivel no va seguido del nombre-de-dato facilitado por el usuario o de FILLER. Se salta la anotación de descripción de datos.</p>
064	P	<p><b>DUPLICATE DATA-NAME</b> <i>(Nombre de dato duplicado)</i></p> <p>Más de una anotación tienen el mismo nombre y el mismo origen. Se procesan todas las anotaciones.</p>
065	P	<p><b>FILLER OR PICTURE ON GROUP LEVEL</b> <i>(FILLER o PICTURE en un nivel de grupo)</i></p> <p>Se procesa la anotación. Esta condición no es reconocida hasta que se reconoce el final del grupo. El número de línea indica el final del grupo.</p>
066	U	<p><b>INVALID LEVEL-NUMBER</b> <i>(Número de nivel inválido)</i></p> <p>El número-de-nivel es uno de los siguientes: 0, 50 a 76, 78 a 87 u 89 a 99. Se salta toda la anotación.</p>
067	U	<p><b>INVALID LEVEL 77 ENTRY</b> <i>(Inscripción de nivel 77 inválida)</i></p> <p>En la FILE SECTION hay una anotación de nivel 77. Se salta la anotación.</p>
068	U	<p><b>LEVEL-NUMBER TOO LARGE</b> <i>(Número de nivel demasiado elevado)</i></p> <p>Un número-de-nivel tiene más de dos dígitos. Se salta la anotación de descripción de datos.</p>
069	C	<p><b>INVALID HIERARCHY</b> <i>(Jerarquía inválida)</i></p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) La primera proposición de la FILE SECTION no es un campo FD. Se crea una FD simulada.</li> <li>(2) Un campo con número-de-nivel entre 02 y 49 no tiene un origen de nivel 01. Se crea una anotación de nivel 01 simulada.</li> <li>(3) El primer campo de un registro es un nivel 88. La anotación se salta.</li> </ol>
070	U	<p><b>INVALID SIZE CLAUSE</b> <i>(Cláusula SIZE inválida)</i></p> <p>La palabra SIZE no va seguida de un entero. Se salta toda la descripción de datos.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
071	C	<p><b>INVALID OCCURS CLAUSE</b> (Cláusula OCCURS inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <p>(1) Un campo que está subordinado al menos a tres cláusulas OCCURS tiene una cláusula OCCURS. Se salta la cláusula OCCURS.</p> <p>(2) En una anotación de nivel 01 ó 77 se utiliza una cláusula OCCURS. La cláusula OCCURS se salta.</p>
072	U	<p><b>MISSING OR INVALID OCCURS INTEGER</b> (Número entero de cláusula OCCURS omitido o inválido)</p> <p>Una cláusula OCCURS no va seguida de un entero válido. La cláusula OCCURS se salta.</p>
073	C	<p><b>EDITED ITEM EXCEEDS 30 CHARACTERS</b> (Campo editado excede 30 caracteres)</p> <p>Un campo descrito como editado por una PICTURE o por cláusulas de edición tiene más de 30 caracteres de longitud. La longitud se establece en 30.</p>
074	U	<p><b>PICTURE LENGTH EXCEEDS 4096 OR 3625</b> (Longitud de la PICTURE superior a 4096 ó 3625)</p> <p>La longitud de un campo es mayor de 4096 caracteres para archivos de cintas o 3625 caracteres para archivos de discos. La longitud se fija en 4096 ó 3625, respectivamente.</p>
075	U	<p><b>EXCESSIVE LENGTH INDICATED BY PICTURE</b> (Longitud indicada por PICTURE es excesiva)</p> <p>Se detecta una de las siguientes condiciones y la longitud se fija a cero:</p> <p>(1) El balance, cuando se indica por varias P en una PICTURE, pasa de 17 si todos son enteros, o de 12 si todos son fracciones.</p> <p>(2) La longitud de la fracción es mayor que 13.</p> <p>(3) Se indica una fracción y la longitud del entero es mayor que 18.</p>
076	U	<p><b>MISSING LENGTH SPECIFICATION</b> (Omitida especificación de longitud)</p> <p>Un campo elemental no tiene cláusulas PICTURE o MAP (SIZE). Para fines de compilación, el campo todavía puede ser referenciado en la PROCEDURE DIVISION: sin embargo, es un campo alfanumérico con longitud cero.</p>
077	C	<p><b>STATED LENGTH NOT EQUAL TO PICTURE LENGTH</b> (Longitud especificada no es igual a la longitud de la PICTURE)</p> <p>Un campo tiene una cláusula MAP (SIZE) y una PICTURE, pero especifican diferentes longitudes. Se utiliza la longitud de la PICTURE.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
078	U	<p><b>INVALID RECORD LENGTH</b> (Longitud de registro inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) La longitud del registro de un archivo asignado a la lectora de fichas no es de 80. Los resultados son imprevisibles.</li> <li>(2) La longitud del registro de un archivo asignado a la perforadora de fichas a filas no es de 80.</li> <li>(3) La longitud del registro de un archivo asignado a la perforadora de fichas normal no es el número de columnas requeridas para que sean perforadas o no es 80 si no se especifica el número de columnas a perforar.</li> <li>(4) La longitud del registro de un archivo asignado a la impresora es mayor que 132.</li> </ol>
079	C	<p><b>LENGTH CONFLICT</b> (Conflicto de longitud)</p> <p>La longitud de un campo de grupo, como se especifica por su cláusula MAP (SIZE), no es igual a la suma de las longitudes de sus campos subordinados. Se ignora la longitud de grupo especificada.</p>
080	C	<p><b>INVALID ITEM LENGTH</b> (Longitud de campo inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) La longitud del campo es mayor de 4096 caracteres para archivos de cintas o 3625 caracteres para archivos de discos. Se fijan 4096 ó 3625, respectivamente.</li> <li>(2) La longitud del entero es mayor que 17 para un campo computacional. Se fija en 17.</li> </ol>
081	C	<p><b>INVALID LENGTH OF ITEM WITH VALUE</b> (Longitud de campo con cláusula VALUE inválida)</p> <p>La longitud de un campo con una cláusula VALUE es mayor que 132. Se establece en 132.</p>
082	C	<p><b>USAGE/EDITING CONFLICT</b> (Conflicto entre cláusula USAGE/Edición)</p> <p>Para editar un campo de clase alfabética o de uso computacional se utiliza la cláusula de edición BLANK WHEN ZERO. Se salta la cláusula de edición.</p>
083	U	<p><b>USAGE FOR CLASS CONFLICT</b> (Conflicto de cláusula USAGE para CLASS)</p> <p>Un campo de grupo tiene una clase o uso que está en desacuerdo con el de un campo subordinado. Se utiliza la clase o uso del campo subordinado.</p>



## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
084	C	<b>PICTURE/USAGE CONFLICT</b> <i>(Conflicto entre cláusulas PICTURE y USAGE)</i>
		Las cláusulas USAGE y PICTURE de una anotación de descripción de datos están en desacuerdo. Tiene preferencia la cláusula PICTURE.
085	U	<b>INVALID JUSTIFIED CLAUSE</b> <i>(Cláusula JUSTIFIED inválida)</i>
		La palabra JUSTIFIED no va seguida de la palabra RIGHT. Se salta toda la anotación de descripción de datos.
086	C	<b>INVALID JUSTIFIED NUMERIC</b> <i>(Campo numérico con cláusula JUSTIFIED inválida)</i>
		Un campo numérico tiene una cláusula JUSTIFIED. Se ignora la cláusula.
087	C	<b>GROUP FRACTION, SIGN, EDITING, OR SCALING</b> <i>(Campo de grupo con fracción, signo, edición o balance)</i>
		Un campo de grupo tiene una fracción, signo, edición o balance que se ignoran.
088	C	<b>UNSIGNED ITEM HAS SIGNED VALUE</b> <i>(Campo sin signo tiene un valor con signo)</i>
		El valor queda sin signo.
089	U	<b>NON ZERO FIGURATIVE CONSTANT VALUE WITH NUMERIC ITEM</b> <i>(Campo numérico con valor de constante figurativa no cero)</i>
		La cláusula VALUE se salta.
090	C	<b>ERRONEUS CHARACTER IN PICTURE</b> <i>(Carácter erróneo en cláusula PICTURE)</i>
		La longitud se fija a cero y la clase que se asume es la alfanumérica.
091	U	<b>ERRONEUS PICTURE CHARACTER-STRING</b> <i>(Cadena de caracteres errónea en cláusula PICTURE)</i>
		Una PICTURE contiene una combinación no válida de caracteres válidos para PICTURE. La longitud se pone a cero y la clase que se asume es la alfanumérica.
092	P	<b>NOT MINIMUM PICTURE</b> <i>(Cláusula PICTURE con no mínimos)</i>
		Una PICTURE no contiene dos símbolos monetarios o por lo menos uno de los siguientes signos: A X 9 Z *. La longitud del campo de datos se pone a cero, y es clasificado como no válido por el compilador. Las referencias a este campo de datos pueden causar otros diagnósticos.

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
093	U	<p><b>INVALID PICTURE DESCRIPTION</b> (Descripción inválida de una cláusula PICTURE)</p> <p>Los caracteres de una cláusula PICTURE no comprenden una combinación válida. La longitud del campo de datos se pone a cero. El compilador lo clasifica como no válido y las referencias a este campo de datos pueden producir otros diagnósticos.</p>
094	C	<p><b>REDEFINED ITEM LENGTH CONFLICT</b> (Conflicto en la longitud de un campo redefinido)</p> <p>La longitud de un campo redefinido no es igual a la longitud del campo redefinido. La longitud redefinida es la que se toma como longitud del campo.</p>
095	U	<p><b>INVALID REDEFINES</b> (Redefinición inválida)</p> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) El nivel de REDEFINES es mayor que cinco. Se salta la cláusula REDEFINES.</li> <li>(2) Un campo redefinido y su redefinidor no están inmediatamente subordinados al mismo campo de datos. La cláusula REDEFINES se salta.</li> <li>(3) Un número-de-nivel de un campo redefinido no es igual al número-de-nivel del campo redefinidor. Se salta la cláusula REDEFINES.</li> <li>(4) El nombre-de-dato redefinido no es válido. Se redefine el campo anterior de igual nivel.</li> </ol>
096	U	<p><b>INVALID REDEFINES CLAUSE</b> (Cláusula REDEFINES inválida)</p> <p>REDEFINES no va seguido de un nombre-de-dato. Se salta toda la anotación de descripción de datos.</p>
097	U	<p><b>INVALID VALUE</b> (Valor inválido)</p> <p>El valor especificado no es ni un literal ni una constante figurativa. Toda la descripción de datos se salta.</p>
098	C	<p><b>EDITED LITERAL IN VALUE OF CLAUSE</b> (Literal editado en cláusula VALUE OF)</p> <p>En una cláusula VALUE OF hay literales editados. El campo se procesa como literal numérico con los puntos y las comas suprimidos.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
099	C	<p><b>VALUE LENGTH GREATER THAN ITEM LENGTH</b> (Longitud del valor mayor que la del campo)</p> <p>La longitud de un valor es mayor que la longitud del campo. El valor se trunca. Si el valor no es numérico, se trunca por la derecha a menos que esté ajustado a la derecha. Si el valor es numérico y la parte entera es demasiado grande, el valor entero se trunca por la izquierda; si la parte fraccionaria es demasiado grande, se trunca por la derecha.</p>
100	C	<p><b>NUMERIC VALUE ASSIGNED TO NONNUMERIC ITEM</b> (Valor numérico asignado a campo no numérico)</p> <p>El campo no es numérico, pero el valor es numérico. El valor se trunca por la derecha y no se lleva a efecto la alineación por la coma decimal.</p>
101	U	<p><b>INVALID VALUE CLAUSE</b> (Cláusula VALUE inválida)</p> <p>Se detecta una de las siguientes condiciones y se salta la cláusula VALUE:</p> <ol style="list-style-type: none"> <li>(1) En la misma anotación de descripción de datos hay una cláusula VALUE y una cláusula OCCURS.</li> <li>(2) En una anotación de descripción de datos subordinada a otra que tiene una cláusula OCCURS, REDEFINES o VALUE, hay una cláusula VALUE.</li> </ol>
102	U	<p><b>INVALID RECORDING MODE CLAUSE</b> (Cláusula RECORDING MODE inválida)</p> <p>Como modo de registro se designa otro diferente a U, F o V. Se salta toda la anotación FD y todas las descripciones de registro subordinadas se atribuyen a la anterior anotación FD válida.</p>
103	U	<p><b>INVALID SYNCHRONIZED CLAUSE</b> (Cláusula SYNCHRONIZED inválida)</p> <p>La palabra LEFT o RIGHT falta o está mal escrita. Se salta toda la descripción de datos.</p>
104	U	<p><b>INVALID USAGE CLAUSE</b> (Cláusula USAGE inválida)</p> <p>USAGE se especifica diferente a DISPLAY, COMPUTATIONAL, COMPUTATIONAL-H, COMP o COMP-H. La descripción de datos se salta.</p>
106	C	<p><b>DUPLICATE CLAUSES IN FD OR DATA DESCRIPTION</b> (Cláusulas duplicadas en descripciones FD o de datos)</p> <p>Están duplicadas las cláusulas de una FD o descripción de datos. Se salta la primera aparición de la cláusula.</p>

## DIAGNOSTICOS DE LA DATA DIVISION

N.º	Tipo	Mensaje (traducción) y explicación del error
107	U	<p><b>INVALID INTEGER IN BLOCK CONTAINS, SIZE OR OCCURS CLAUSE</b> (Entero inválido en cláusulas <i>BLOCK CONTAINS</i>, <i>SIZE</i> u <i>OCCURS</i>)</p> <p>El entero de una cláusula <i>BLOCK CONTAINS</i>, <i>SIZE</i> u <i>OCCURS</i> tiene más de cuatro dígitos. Se salta toda la anotación de descripción de datos.</p>
108	P	<p><b>INCORRECT MARGIN</b> (Margen incorrecto)</p> <p>Un campo de nivel 01 ó 77 no empieza en el margen A (columnas 8 a 11).</p>
109	C	<p><b>NUMERIC LITERAL CONTAINS MORE THAN ONE DECIMAL POINT</b> (Literal numérico contiene más de un punto decimal)</p> <p>Un literal numérico editado contiene más de una coma decimal. Los enteros que siguen a la primera coma decimal se consideran como parte fraccionaria del literal. El resto de esta fracción aparente se suprime. Esta condición de error puede surgir cuando se usa una proposición <i>DECIMAL POINT IS COMMA</i> y un literal numérico editado contiene más de un punto.</p>
110	P	<p><b>ILLEGAL VALUE IS CLAUSE</b> (Cláusula <i>VALUE IS</i> ilegal)</p> <p>No está permitida la cláusula <i>VALUE IS</i> en la <i>FILE SECTION</i>, ya que no se ajusta al COBOL ANSI. Sin embargo, se procesa.</p>
111	C	<p><b>INVALID DISC VOLUME NUMBER</b> (Número de volumen de un disco inválido)</p> <p>El número del volumen del disco tiene más de un octeto.</p>
112	C	<p><b>INVALID DISC VOLUME NUMBER</b> (Número de volumen de un disco inválido)</p> <p>El valor del número del volumen del disco pasa de 8.</p>
113	C	<p><b>ACCES MODE AND ORGANIZATION CLAUSE ARE MISSING</b> (Cláusulas <i>ACCES MODE</i> y <i>ORGANIZATION</i> omitidas)</p> <p>Se supone <i>SEQUENTIAL</i> para ambas cláusulas y continúa el proceso.</p>
114	U	<p><b>ACTUAL KEY OR RECORD KEY CLAUSES ARE PRESENT FOR A SEQUENTIAL FILE</b> (Para un fichero secuencial están presentes las cláusulas <i>ACTUAL KEY</i> o <i>RECORD KEY</i>)</p> <p>No se utilizan las cláusulas <i>ACTUAL KEY</i> y/o <i>RECORD KEY</i> y se saltan; el resultado es imprevisible.</p>
115	P	<p><b>MISSING LABEL RECORD CLAUSE</b> (Cláusula <i>LABEL RECORD</i> omitida)</p> <p>El COBOL ANSI requiere la cláusula <i>LABEL RECORD</i>. Sin embargo, la FD se procesa sin ella.</p>

<b>DIAGNOSTICOS DE LA PROCEDURE DIVISION (referidos a verbos aritméticos)</b>		
N.º	Tipo	Mensaje (traducción) y explicación del error
121	U	<b>INVALID ARITHMETIC OPERAND</b> <i>(Operando aritmético inválido)</i>
		El operando de un verbo aritmético no es un literal numérico, constante figurativa o campo de datos elemental numérico de longitud permitida. La proposición se salta.
122	U	<b>INVALID ARITHMETIC RECEIVING OPERAND</b> <i>(Operando receptor de un campo aritmético inválido)</i>
		El operando receptor de un verbo aritmético no es un campo de datos elemental numérico de longitud permitida. Se salta la proposición.
123	U	<b>MISSING OR INVALID OPERAND</b> <i>(Operando inválido u omitido)</i>
		Se salta la proposición.
124	U	<b>ONLY 1 SUBJECT IN ADD GIVING STATEMENT</b> <i>(Solamente un sujeto en verbo ADD con opción GIVING)</i>
		Se salta la proposición.
125	U	<b>ON SIZE ERROR CLAUSE MISSING OR INVALID</b> <i>(Cláusula SIZE ERROR omitida o inválida)</i>
		La opción ON SIZE ERROR no contiene una proposición imperativa. La opción se salta.
126	U	<b>MISSING OR INVALID SUBTRACT OPERAND</b> <i>(Operando en verbo SUBTRACT omitido o inválido)</i>
		Se salta la proposición.
127	U	<b>MISSING OR INVALID MULTIPLY OR DIVIDE OPERAND</b> <i>(Operando en verbo MULTIPLY O DIVIDE inválido u omitido)</i>
		Se salta la proposición.
128	U	<b>DIVISION BY FIGURATIVE CONSTANT ZERO</b> <i>(División por la constante figurativa ZERO)</i>
		Se salta la proposición.
129	P	<b>MORE THAN IS DIGITS IN DIVISOR</b> <i>(Más de 15 dígitos en el divisor)</i>
		Se emplean los 15 dígitos más significativos.
130	P	<b>MORE THAN IS DIGITS IN MULTIPLY OPERAND</b> <i>(Más de 15 dígitos en un operando del verbo MULTIPLY)</i>
		Se emplean los 15 dígitos más significativos.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a la declaración condicional IF)**

N.º	Tipo	Mensaje (traducción) y explicación del error
134	C	<p><b>INVALID RELATION TEST OPERAND</b> (Operando de prueba de relación inválido)</p> <p>No es válida una prueba de relación porque no son válidos uno o más operandos para la prueba. Se asume una prueba de relación alfanumérica.</p>
135	P	<p><b>THEN NOT FOLLOWED BY VERB</b> (THEN no seguido por un verbo)</p> <p>La palabra THEN se salta.</p>
136	U	<p><b>IMPROPERLY DEFINED IF OPERAND</b> (Operando de IF definido impropriamente)</p> <p>La proposición se salta.</p>
137	U	<p><b>IF WITH INVALID OR MISSING SUBJECT</b> (IF con sujeto inválido u omitido)</p> <p>La palabra IF no va seguida por un operando válido. La proposición se salta.</p>
138	U	<p><b>IF WITH INVALID OR MISSING OBJECT</b> (IF con objeto inválido u omitido)</p> <p>El operando siguiente a un operador de relación no es válido o falta. La proposición se salta.</p>
139	U	<p><b>NO IMPERATIVE VERB IN IF STATEMENT</b> (Verbo no imperativo en sentencia IF)</p> <p>La prueba de condición de una proposición IF no va seguida de un verbo imperativo. La proposición se salta.</p>
140	U	<p><b>NO ELSE VERB IN IF STATEMENT</b> (Palabra ELSE en una sentencia IF no seguida de un verbo)</p> <p>La palabra ELSE de una proposición IF no va seguida de un verbo imperativo. La proposición se salta.</p>

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a la declaración condicional IF)**

N.º	Tipo	Mensaje (traducción) y explicación del error
141	U	<p><b>INVALID IF OR IF NOT STATEMENT</b> <i>(Sentencia IF o IF NOT inválida)</i></p> <hr/> <p>Un operador de condición de clase o de relación no sigue a la palabra NOT, o el operando siguiente a un operador de relación no es válido. La proposición se salta.</p>
142	C	<p><b>INVALID CLASS TEST</b> <i>(Prueba de clase inválida)</i></p> <hr/> <p>Se detecta una de las siguientes condiciones:</p> <ol style="list-style-type: none"> <li>(1) En una prueba ALPHABETIC, el campo está descrito como numérico.</li> <li>(2) En una prueba NUMERIC, el campo está descrito como alfabético.</li> <li>(3) El campo está descrito como computational.</li> </ol> <p>El campo se considera alfanumérico y se acepta la prueba de clase.</p>

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
(referidos a los verbos de entrada/salida)

N.º	Tipo	Mensaje (traducción) y explicación del error
144	U	<b>SEEK WITH MISSING OR INVALID FILE-NAME</b> (Verbo <i>SEEK</i> con nombre-fichero omitido o inválido)
		Falta o no es válido el nombre-de-archivo en una proposición <i>SEEK</i> . La proposición se salta.
145	C	<b>INVALID ADVANCING OPTION</b> (Opción <i>ADVANCING</i> inválida)
		Para un archivo que no es de impresora se especifica la opción <i>ADVANCING</i> del verbo <i>WRITE</i> . Se ignora la opción <i>ADVANCING</i> .
146	U	<b>INVALID ACCEPT OR DISPLAY OPERAND</b> (Operando de <i>ACCEPT</i> o <i>DISPLAY</i> inválido)
		La proposición se salta.
147	U	<b>MISSING OR INVALID CLOSE FILE-NAME</b> (Verbo <i>CLOSE</i> con nombre-fichero omitido o inválido)
		La proposición se salta.
148	U	<b>MISSING OR INVALID AT END OR INVALID KEY CLAUSE</b> (Cláusula <i>AT END</i> o <i>INVALID KEY</i> omitida o inválida)
		Falta la cláusula <i>AT END</i> o <i>INVALID KEY</i> de una proposición <i>READ</i> , o no contiene una proposición imperativa. La cláusula <i>AT END</i> o <i>INVALID KEY</i> se salta.
149	U	<b>MISSING OR INVALID OPEN FILE-NAME</b> (Verbo <i>OPEN</i> con nombre-fichero omitido o inválido)
		Si la proposición no contiene operandos válidos, se salta toda ella.
150	U	<b>MISSING OR INVALID OPERAND IN ADVANCING OPTION</b> (Operando en opción <i>ADVANCING</i> omitido o inválido)
		En una proposición <i>WRITE</i> , a la palabra <i>ADVANCING</i> no le sigue un operando válido. La proposición <i>WRITE</i> se salta.
153	U	<b>MISSING OR INVALID READ FILE-NAME</b> (Verbo <i>READ</i> con nombre-fichero omitido o inválido)
		La proposición se salta.
154	U	<b>MISSING OR INVALID WRITE RECORD-NAME</b> (Verbo <i>WRITE</i> con nombre de registro omitido o inválido)
		La proposición se salta.



**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a los verbos de entrada/salida)**

N.º	Tipo	Mensaje (traducción) y explicación del error
157	U	<b>OPEN WITHOUT INPUT OR OUTPUT</b> (Verbo OPEN sin opción INPUT o OUTPUT)
		La palabra OPEN no va seguida inmediatamente por la palabra INPUT o OUTPUT. La proposición se salta.
158	U	<b>OUTPUT DEVICE AS INPUT</b> (Dispositivo de salida como entrada)
		Se salta la proposición.
159	U	<b>INPUT DEVICE OPENED AS OUTPUT</b> (Dispositivo de entrada abierto como salida)
		Se salta la proposición.
160	U	<b>ATTEMPTED READ ON OUTPUT DEVICE</b> (Intento de lectura sobre dispositivo de salida)
		Se salta la proposición.
161	U	<b>ATTEMPTED WRITE ON INPUT DEVICE</b> (Intento de escritura sobre dispositivo de entrada)
		Se salta la proposición.
162	U	<b>FRACTIONAL WRITE ADVANCING LITERAL</b> (Literal fraccionario en opción ADVANCING del verbo WRITE)
		Se salta la proposición de avance y se genera un avance de una línea.
163	C	<b>CLOSE OPTION INVALID FOR NON-TAPE FILE</b> (Opción de cierre inválida para ficheros no en banda magnética)
		Se ignora la opción.
164	C	<b>DISPLAY EXCEEDS 132 CHARACTERS</b> (DISPLAY excede 132 caracteres)
		Un campo de datos DISPLAY pasa de 132 caracteres, o en un DISPLAY múltiple una serie de literales para DISPLAY y el campo de datos pasan de 132 caracteres.
165	U	<b>INVALID LITERAL IN WRITE STATEMENT</b> (Literal inválido en sentencia WRITE)
		Se han utilizado caracteres no numéricos para el literal de la opción ADVANCING de la proposición WRITE. El literal debe ser numérico.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a los verbos de entrada/salida)**

N.º	Tipo	Mensaje (traducción) y explicación del error
166	U	<b>INVALID OPEN STATEMENT</b> (Sentencia OPEN inválida)
		La palabra INPUT u OUTPUT se utiliza más de una vez en una proposición OPEN. La proposición se salta.
167	U	<b>INVALID OPEN STATEMENT</b> (Sentencia OPEN inválida)
		Se abre un dispositivo de entrada o salida como I-O.
168	U	<b>MISSING OR INVALID ACTUAL KEY CLAUSE</b> (Cláusula ACTUAL KEY omitida o inválida)
169	U	<b>MISSING OR INVALID AREA CLAUSE</b> (Cláusula AREA omitida o inválida)
170	U	<b>INVALID READ STATEMENT</b> (Sentencia READ inválida)
		Al verbo READ de la proposición READ le sigue una cláusula incorrecta.
171	U	<b>MISSING OR INVALID KEY CLAUSE</b> (Cláusula KEY omitida o inválida)
172	P	<b>INVALID KEY CLAUSE</b> (Cláusula KEY inválida)
		Se detecta una cláusula KEY no válida. Se salta la proposición.
173	U	<b>INVALID INSERT STATEMENT</b> (Sentencia INSERT inválida)
174	U	<b>INVALID FILE-NAME ORGANIZATION</b> (Organización del fichero inválida)
		La organización de nombre-de-archivo debe ser indexada.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
(referidos a los verbos de manipulación de datos)

N.º	Tipo	Mensaje (traducción) y explicación del error
175	C	<b>ALPHANUMERIC EDITED MOVE TRUNCATION</b> (Truncamiento de un campo alfanumérico editado con el verbo MOVE)
		El campo de envío de una transferencia alfanumérica editada es mayor que el campo receptor. El campo de envío se trunca por la derecha.
176	U	<b>ATTEMPTING TO MOVE NON ZERO FIGURATIVE CONSTANT TO COMPUTATIONAL ITEM</b> (Intento de mover una constante figurativa no cero a un campo COMPUTATIONAL)
		La proposición se salta.
177	P	<b>NUMERIC MOVE TRUNCATION</b> (Truncamiento en un movimiento de un campo numérico)
		El campo de envío de una proposición MOVE se describe para contener más dígitos que el campo receptor. La transferencia se ejecuta con alineación de coma decimal (los campos de cómputo que contienen un número par de dígitos tienen asignado un dígito adicional por el compilador, lo cual puede originar este diagnóstico).
178	U	<b>MISSING NON NUMERIC LITERAL OR FIGURATIVE CONSTANT</b> (Omisión de un literal no numérico o de una constante figurativa)
		La palabra ALL del verbo EXAMINE no va seguida por una constante figurativa o un literal no numérico. La proposición se salta.
179	U	<b>MISSING OR INVALID MOVE OPERAND</b> (Operando de un verbo MOVE omitido o inválido)
		Se salta la proposición.
181	C	<b>NUMERIC WITH FRACTION AND NONNUMERIC IN MOVE</b> (Movimiento de un campo numérico decimal a un campo no numérico)
		Un campo numérico cuya coma decimal implícita no va inmediatamente a la derecha del dígito de la derecha se transfiere a un campo no numérico o es el campo receptor de un campo no numérico. Los campos se alinean por los caracteres de la izquierda.
182	C	<b>INDEX-NAME IN MOVE</b> (Nombre-índice en verbo MOVE)
		Un operando de una proposición MOVE es un nombre-índice o un campo-de-datos índice. Se genera una MOVE de alfanumérico a alfanumérico.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
(referidos a los verbos de manipulación de datos)

N.º	Tipo	Mensaje (traducción) y explicación del error
183	C	<b>MOVE NUMERIC TO ALPHABETIC</b> (Movimiento de un campo numérico a alfabético)
		Una proposición MOVE tiene un campo de envío numérico y uno receptor alfabético. Se genera una MOVE de alfanumérico o alfanumérico.
184	U	<b>MISSING OR INVALID EXAMINE OPERAND</b> (Operando del verbo EXAMINE omitido o inválido)
		Se salta la proposición.
185	U	<b>MISSING OR INVALID EXAMINE OPTION</b> (Opción del verbo EXAMINE omitida o inválida)
		La palabra TALLYING o REPLACING no va seguida por palabra o palabras permitidas. La proposición se salta.
186	U	<b>INVALID EXAMINE LITERAL</b> (Literal del verbo EXAMINE inválido)
		El literal de una proposición EXAMINE tiene más de un carácter, no es una constante figurativa o pertenece a una clase que no es compatible con la del operando. La proposición se salta.
187	U	<b>EXAMINE OPERAND IS COMPUTATIONAL</b> (Operando del verbo EXAMINE es COMPUTATIONAL)
		El operando de una proposición EXAMINE se describe como computacional. La proposición se salta.
188	C	<b>MOVE NON NUMERIC TO NUMERIC</b> (Movimiento de un campo no numérico a numérico)
		Una proposición MOVE tiene un campo de envío no numérico y un campo receptor numérico. Se genera una MOVE de alfanumérico a alfanumérico.
189	C	<b>EDITED SENDING ITEM</b> (Campo emisor editado)
		Una proposición MOVE tiene un campo emisor editado. Se genera una MOVE de alfanumérico a alfanumérico.
190	U	<b>INVALID FIGURATIVE CONSTANT OR LITERAL</b> (Constante figurativa o literal inválida)
		Se detecta una de las siguientes condiciones, que hacen que se salte la proposición: (1) Una proposición condicional contiene una constante figurativa o literal en su sujeto y en su objeto. (2) En una proposición aritmética hay una constante figurativa que no es ZERO.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
**(referidos a los verbos de manipulación de datos)**

N.º	Tipo	Mensaje (traducción) y explicación del error
191	P	<p><b>ALPHANUMERIC MOVE TRUNCATION</b>  <i>(Truncamiento en un movimiento de un campo alfanumérico)</i></p> <p>El campo emisor de una proposición MOVE es mayor que el campo receptor. El campo emisor se trunca por la derecha.</p>
192	C	<p><b>ALPHANUMERIC OR NUMERIC TO NUMERIC MOVE TRUNCATION</b>  <i>(Truncamiento en un movimiento de un campo alfanumérico o numérico a numérico)</i></p> <p>Se genera una MOVE numérica, y la truncación se realiza por la coma decimal.</p>
193	C	<p><b>SENDING AND RECEIVING FIELDS</b>  <i>(Campos emisores y receptores)</i></p> <p>Un campo editado o alfabético se transfiere a un campo numérico. Si el campo receptor no es editado, se genera una transferencia de alfanumérico a alfanumérico; si el campo receptor es editado, se genera una transferencia de alfanumérico a numérico editado.</p>

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
(referidos a los verbos de control de secuencia)

N.º	Tipo	Mensaje (traducción) y explicación del error
194	U	<b>MAXIMUM NUMBER OF CONDITIONAL STATEMENT EXCEEDED</b> (Se excede el máximo número de sentencias condicionales)
		El número de proposiciones condicionales pasa de 400 (sólo para el compilador en discos). Se salta la proposición.
195	P	<b>INVALID PERFORM STATEMENT</b> (Sentencia PERFORM inválida)
		Una proposición PERFORM no ANSI intenta cruzar segmentos secundarios. Se produce una indicación de la proposición para compatibilidad y se ejecuta.
196	P	<b>INVALID ALTER OF PARAGRAPH</b> (Párrafo del verbo ALTER inválida)
		Se ha alterado un párrafo dentro del mismo segmento secundario. El párrafo alterado permanece en su último estado alterado si se vuelve a cargar el segmento.
197	P	<b>ALTERED SEGMENT NOT RELOADED IN INITIAL STATE</b> (Segmento alterado no vuelto a cargar en su estado Inicial)
		Este diagnóstico es para compatibilidad del COBOL ANSI. El segmento se ejecuta en la forma escrita.
198	U	<b>ARGUMENTS ON CALL LINE EXCEED 30 ITEMS</b> (Número de argumentos en CALL exceden a 30)
		Los nombres y argumentos de rutinas de la proposición CALL pasan de 30.
199	P	<b>POSSIBLE PERFORM RANGE ERROR</b> (Error posible en un segundo nivel de PERFORM)
		La definición del segundo nombre de procedimiento precede a la definición del primero. A menos que se transfiera el control desde el ámbito del primer nombre de procedimiento a la coma anterior al segundo nombre de procedimiento, los resultados son imprevisibles.
200	U	<b>INVALID PERFORM STATEMENT</b> (Sentencia PERFORM inválida)
		Un nombre de procedimiento referenciado en una proposición PERFORM está definido de manera inadecuada, o el ámbito de la PERFORM abarca dos segmentos. En cualquier caso la proposición se salta.
201	P	<b>MISSING OR INVALID PARAGRAPH-NAME</b> (Nombre de párrafo omitido o inválido)
		A un nombre-de-sección o a una proposición USE no le sigue un nombre-de-párrafo válido.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION**  
(referidos a los verbos de control de secuencia)

N.º	Tipo	Mensaje (traducción) y explicación del error
202	U	<b>INVALID OPERAND IN PERFORM OR GO TO DEPENDING</b> (Operando inválido en PERFORM o GO TO DEPENDING)
		El campo de datos referenciado en una proposición PERFORM o GO TO DEPENDING no es un campo elemental numérico. La proposición se salta.
203	U	<b>MISSING OR INVALID GO TO OPERAND</b> (Operando inválido u omitido a GO TO)
		Un verbo GO TO no va seguido de un nombre-de-procedimiento válido. La proposición se salta.
204	U	<b>INVALID PROCEDURE-NAME QUALIFIER</b> (Calificador de nombre de procedimiento inválido)
		En una referencia de nombre-de-procedimiento, a OF o IN no le sigue un calificador válido. La proposición se salta.
205	U	<b>MISSING OR INVALID PERFORM OPERAND</b> (Operando de PERFORM omitido o inválido)
		Se salta la proposición.
206	U	<b>INVALID ENTER OPERAND</b> (Operando de ENTER inválido)
		Se salta la proposición.
207	U	<b>MISSING OR INVALID ALTER OPERAND</b> (Operando de ALTER omitido o inválido)
		El nombre-de-procedimiento de una proposición ALTER falta o no es válido. La proposición ALTER se salta.
208	U	<b>MISSING OR INVALID GO TO DEPENDING OPERAND</b> (Operando de GO TO DEPENDING omitido o inválido)
		Se salta la proposición.
209	U	<b>INVALID EXIT</b> (EXIT inválido)
		Un verbo EXIT no va precedido por un nombre-de-párrafo. Se salta.
210	U	<b>PROCEDURE-NAME LIMIT EXCEEDED</b> (Límite excedido para nombres de procedimiento)
		El compilador ha excedido los límites internos para nombres-de-procedimiento. Se ignoran todos los nombre-de-procedimiento sucesivos.

<b>DIAGNOSTICOS DE LA PROCEDURE DIVISION</b> (referidos a los verbos de control de secuencia)		
N.º	Tipo	Mensaje (traducción) y explicación del error
211	U	<b>UNDEFINED PARAGRAPH-NAME QUALIFIER</b> (Calificador de nombre de procedimiento sin definir)
		El calificador de una referencia de nombre-de-procedimiento no está definido. La proposición se salta.
212	U	<b>INVALID UNQUALIFIED PARAGRAPH-NAME REFERENCE</b> (Sin definición para una referencia de nombre de párrafo)
		No se hace ninguna definición para una referencia de nombre-de-párrafo no calificado, o se hacen varias definiciones y esta referencia debe ser calificada. Si no hay definición, se salta la referencia; si hay varias definiciones, se utiliza la última.
213	U	<b>UNDEFINED QUALIFIED PARAGRAPH-NAME</b> (Nombre de párrafo calificado sin definir)
		Un nombre-de-párrafo calificado no está definido en la sección utilizada como calificador. La proposición se salta.
214	U	<b>DUPLICATE PROCEDURE-NAME DEFINITION</b> (Definición duplicada de nombre de procedimiento)
		Dentro de una sección hay nombres-de-párrafo o nombres-de-sección duplicados. Se utiliza la última aparición del nombre-de-procedimiento.
215	P	<b>INVALID GO TO STATEMENT</b> (Sentencia GO TO inválida)
		Una proposición GO TO no está en una proposición condicional y no es la última proposición de un párrafo. El resto del párrafo que sigue a la proposición GO TO no se puede ejecutar.
216	P	<b>MISSING CHANNEL NUMBER IN SPECIAL-NAMES CLAUSE</b> (Número de canal omitido en una cláusula SPECIAL-NAMES)
		No se ha fijado el número-de-canal en la cláusula SPECIAL-NA-MES. La proposición se procesa en la forma escrita.



**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a sentencias que contienen subíndices)**

N.º	Tipo	Mensaje (traducción) y explicación del error
221	C	<b>SUBSCRIPT LENGTH EXCEEDS 16 (DATA-NAME)</b> <i>(Longitud del subíndice excede 16, nombre dato)</i>
		La longitud se reduce a 16 por truncación a la izquierda.
222	C	<b>SUBSCRIPT LENGTH EXCEEDS 16 (LITERAL)</b> <i>(Longitud del subíndice excede 16, literal)</i>
		La longitud se reduce a 16 por truncación a la izquierda.
223	P	<b>SUBSCRIPT NOT NUMERIC, ELEMENTARY ITEM</b> <i>(Subíndice no numérico, campo elemental)</i>
		El subíndice se procesa como campo numérico elemental.
224	C	<b>INVALID LITERAL SUBSCRIPT VALUE</b> <i>(Valor del subíndice literal inválido)</i>
		El valor de un subíndice literal es igual o menor que cero o tiene más repeticiones que el campo de datos. El valor del subíndice se pone a 1.
225	U	<b>EXCESSIVE OR INVALID SUBSCRIPTS OR INDEX-NAMES</b> <i>(Excesivos o inválidos subíndices o nombres-índice)</i>
		Se detecta una de las siguientes condiciones, haciendo que la proposición salte: (1) Los operandos dentro de paréntesis no son válidos como subíndices o nombres-índice. (2) Hay más de tres operandos dentro de los paréntesis.
226	U	<b>INVALID SUBSCRIPT OR INDEX-NAME QUALIFICATION</b> <i>(Subíndice o calificación de nombre-índice inválido)</i>
		La palabra OF o IN, precedida por un subíndice o nombre-índice, no va seguida de un calificador válido. La proposición se salta.
227	C	<b>SUBSCRIPT NOT AN INTEGER</b> <i>(Subíndice no entero)</i>
		Hay un subíndice descrito con dígitos fraccionarios. Sólo se emplean los enteros.
228	C	<b>ITEM NOT SUBSCRIPTED OR INDEXED</b> <i>(Campo no subindexado o indexado)</i>
		Nombre-de-dato descrito en la DATA DIVISION con una cláusula OCCURS, pero en la referencia no se emplea la subindexación o indexación. Por cada subíndice o nombre-índice que falte se crea un literal con un valor 1.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(referidos a sentencias que contienen subíndices)**

N.º	Tipo	Mensaje (traducción) y explicación del error
229	C	<p><b>ITEM IMPROPERLY SUBSCRIPTED OR INDEXED</b> <i>(Campo indebidamente subindexado o indexado)</i></p> <p>Nombre-de-dato descrito en la DATA DIVISION sin cláusula OCCURS, pero en la referencia se emplea la subindexación o indexación. Se ignora la subindexación o indexación.</p>
230	C	<p><b>INCORRET NUMBER OR SUBSCRIPTED OR INDEX-NAMES</b> <i>(Número incorrecto de subíndices o nombres-índice)</i></p> <p>El número de niveles de OCCURS no concuerda con el número de subíndices o nombres-índice empleados en la referencia. El exceso de subíndices o nombres-índice se suprime por la derecha, o, si se necesitan más subíndices o nombres-índice, se insertan a la izquierda literales con valor 1. Si el número de subíndices o nombres-índice pasa de tres, se salta toda la proposición.</p>

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(diversos)**

N.º	Tipo	Mensaje (traducción) y explicación del error
233	U	<b>INVALID STOP LITERAL</b> <i>(Literal en verbo STOP inválido)</i>
		El literal que sigue al verbo STOP no es hexadecimal (es decir, 0 a 9, A a F). La proposición se salta.
234	C	<b>DUPLICATE NAME</b> <i>(Nombre duplicado)</i>
		Se hace una referencia a un nombre-de-dato, -de-condición o -de-registro que está duplicado dentro de la DATA DIVISION. La compilación continúa utilizando la primera definición.
235	U	<b>UNDEFINED NAME</b> <i>(Nombre indefinido)</i>
		Un operando utilizado en un procedimiento está indefinido. El operando se elimina del procedimiento. Toda la proposición se salta si ya no contiene el número mínimo de operandos necesarios.
236	U	<b>OPERAND NOT A DATA-NAME</b> <i>(Operando no a un nombre-dato)</i>
		Un operando que debía ser un nombre-de-dato no lo es. El nombre-de-dato queda indefinido a menos que se defina un nombre duplicado como nombre-de-dato en otra parte. Si está indefinido, el nombre-de-dato se omite. Toda la proposición se salta, si ya no contiene el número mínimo de operandos necesarios.
237	U	<b>OPERAND NOT A CONDITION-NAME</b> <i>(Operando no a un nombre-condición)</i>
		Un operando que debía ser un nombre-de-condición no lo es. El nombre-de-condición queda indefinido a menos que se defina en cualquier otra parte un nombre duplicado como nombre-de-condición. Si queda indefinido, la proposición se salta.
238	U	<b>OPERAND NOT A FILE-NAME</b> <i>(Operando no es un nombre-fichero)</i>
		Un operando que debía ser un nombre-de-fichero no lo es. El nombre-de-fichero queda indefinido a menos que esté definido un nombre duplicado en cualquier otra parte como nombre-de-fichero. Si el nombre-de-fichero queda indefinido, la proposición se salta; sin embargo, si la proposición es una OPEN que tenga todavía por lo menos un nombre-de-fichero, sólo se omite el nombre-de-fichero indefinido.
239	U	<b>OPERAND NOT A RECORD-NAME</b> <i>(Operando no es un nombre-registro)</i>
		Un operando que debía ser un nombre-de-registro no lo es. El nombre-de-registro queda indefinido a menos que en otra parte esté definido como nombre-de-registro un nombre duplicado. Si queda indefinido, la proposición se salta.

**DIAGNOSTICOS DE LA PROCEDURE DIVISION  
(diversos)**

N.º	Tipo	Mensaje (traducción) y explicación del error
240	U	<b>OPERAND INVALID FOR THIS PARTICULAR VERB STATEMENT</b> <i>(Operando inválido para esta sentencia en particular)</i>
		Una clase, uso, tamaño, etc. de un operando no es válida para esta determinada proposición de verbo. Se saltan el verbo y sus operandos.
241	U	<b>INVALID DATA-NAME QUALIFICATION</b> <i>(Calificación de nombre-dato inválida)</i>
		Las palabras OF o IN, precedidas por un nombre-de-dato, no van seguidas por un calificador válido. El operando se salta. Toda la proposición se salta si ya no queda el número mínimo de operandos requeridos.
242	U	<b>INVALID STOP STATEMENT</b> <i>(Sentencia STOP inválida)</i>
		El verbo STOP no va seguido de la palabra RUN o de un literal de uno a tres caracteres. La proposición se salta.
243	C	<b>PROCEDURE-NAME DEFINITION DOES NOT BEGIN IN MARGIN-A</b> <i>(Definición de nombre-procedimiento no comienza en margen A)</i>
		El nombre-de-procedimiento se procesa como si empezase en el margen A (columnas 8 a 11).
244	C	<b>INVALID PRIORITY NUMBER</b> <i>(Número de prioridad inválido)</i>
		Un número de prioridad de sección pasa de 99. Se pone con el anterior número de prioridad válido.
245	C	<b>PRIORITY NUMBER OUT OF SEQUENCE</b> <i>(Número de prioridad fuera de secuencia)</i>
		Una sección puesta a continuación del primer segmento independiente (uno con número de prioridad mayor que 49) tiene un número de prioridad no igual o mayor que el número de prioridad anterior. El nuevo número de prioridad será el número anterior.
246	U	<b>SEGMENT EXCEEDS MEMORY</b> <i>(Segmento excede el tamaño de memoria)</i>
		La longitud de un segmento es mayor que el tamaño de memoria del ordenador objeto especificado en la ENVIRONMENT DIVISION. El programa no se ejecutará. Se imprime el número de segmento en vez del número de línea.
247	U	<b>MORE THAN 25 SEGMENTS</b> <i>(Más de 25 segmentos)</i>
		En el programa fuente se especifican más de 25 segmentos. No se puede enlazar todo el programa porque se ha excedido el límite de enlazamiento.

<b>DIAGNOSTICOS DE LA PROCEDURE DIVISION (diversos)</b>		
N.º	Tipo	Mensaje (traducción) y explicación del error
248	U	<b>INVALID RECORD KEY CLAUSE</b> <i>(Cláusula RECORD KEY inválida)</i>
		Se ha encontrado una cláusula RECORD KEY no definida o no válida.
249	U	<b>INVALID OCCURS CLAUSE</b> <i>(Cláusula OCCURS inválida)</i>
		La cláusula OCCURS no está permitida en niveles 01, 77 u 88.
250	U	<b>INVALID USE OF VALUE AND OCCURS CLAUSE</b> <i>(Uso inválido de cláusulas VALUE y OCCURS)</i>
		Una anotación de datos no puede contener las dos cláusulas VALUE y OCCURS.
255	P	<b>NOT IMPLEMENTED-DOCUMENTATION ONLY</b> <i>(No implementado - sólo para documentación)</i>
		Reservada para futuras implementaciones.



